Antonia Mas · Antoni Mesquida
Rory V. O'Connor · Terry Rout
Alec Dorling (Eds.)

Communications in Computer and Information Science 770

Software Process Improvement and Capability Determination

17th International Conference, SPICE 2017
Palma de Mallorca, Spain, October 4–5, 2017
Proceedings

Springer

Communications in Computer and Information Science 770

Commenced Publication in 2007
Founding and Former Series Editors:
Alfredo Cuzzocrea, Xiaoyong Du, Orhun Kara, Ting Liu, Dominik Ślęzak, and Xiaokang Yang

More information about this series at http://www.springer.com/series/7899

Antonia Mas · Antoni Mesquida
Rory V. O'Connor · Terry Rout
Alec Dorling (Eds.)

Software Process Improvement and Capability Determination

17th International Conference, SPICE 2017
Palma de Mallorca, Spain, October 4–5, 2017
Proceedings

Springer

Editors
Antonia Mas
University of the Balearic Islands
Palma
Spain

Antoni Mesquida
University of the Balearic Islands
Palma, Baleares
Spain

Rory V. O'Connor
Dublin University
Dublin
Ireland

Terry Rout
Software Quality Institute
Griffith University
Brisbane, QLD
Australia

Alec Dorling
Impronova AB
Askim
Sweden

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-319-67382-0 ISBN 978-3-319-67383-7 (eBook)
DOI 10.1007/978-3-319-67383-7

Library of Congress Control Number: 2017953771

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

On behalf of the SPICE 2017 Conference Organizing Committee we are proud to present the proceedings of the 17th International Conference on Software Process Improvement and Capability Determination (SPICE 2017), held in Palma, Spain, during October 4–5, 2017.

The SPICE project was started in 1993 to support the development of an international standard for software process assessment. The work of the project eventually led to the finalization of ISO/IEC 15504 – Process Assessment, and its complete publication represented a climax for the work of the project. The standardization effort continues, with the publication of the first documents in the new ISO/IEC 330xx family of standards on process assessment.

As part of its charter to provide ongoing publicity and transition support for the emerging standard, the project organized a number of SPICE workshops and seminars, with invited speakers drawn from project participants. These have now evolved to a sustainable set of international conferences with broad participation from academia and industry with a common interest in model-based process improvement. This was the 17th in the series of conferences organized by the SPICE User Group to increase knowledge and understanding of the International Standard and of the technique of process assessment.

The conference program featured invited keynote talks, research papers, and industry experience reports on the most relevant topics related to software process assessment and improvement; a significant focus this year was on detailed studies of aspects of process implementation, assessment, and improvement, and the expansion in the range and variety of relevant process models. Members of the Program Committee selected the papers for presentation following a peer review process.

SPICE conferences have a long history of attracting attendees from industry and academia. This confirms that the conference covers topics that are up to date, important, and interesting. SPICE 2017 offered a unique forum for industry and academic professionals to discuss their needs and ideas in the area of process assessment and improvement and in related aspects of quality management.

On behalf of the SPICE 2017 Conference Organizing Committee, we would like to thank all participants. Firstly all the authors, whose quality work is the essence of the conference, and the members of the Program Committee, who helped us with their expertise and diligence in reviewing all of the submissions. As we all know, organizing a conference requires the effort of many individuals. We also wish to thank all the members of our Organizing Committee, whose work and commitment were invaluable.

October 2017

Antonia Mas
Antoni Mesquida
Rory V. O’Connor
Terry Rout
Alec Dorling

Organization

General Chair

Alec Dorling	Impronova AB, Sweden

Program Co-chairs

Terry Rout	Griffith University, Australia
Antonia Mas	University of the Balearic Islands, Spain

Local Organizing Co-chairs

Antoni Mesquida	University of the Balearic Islands, Spain
Milos Jovanovic	University of Novi Sad, Serbia

Industry Chair

Tomas Schweigert	SQS, Germany

Proceedings Chair

Rory V. O'Connor	Lero, Dublin City University, Ireland

Program Committee

Béatrix Barafort	Luxembourg Institute of Science and Technology, Luxembourg
Luigi Buglione	Engineering Ingegneria Informatica SpA, Italy
Aileen Cater-Steel	University of Southern Queensland, Australia
Melanie Cheong	Macquarie Telecom Group, Australia
Gerhard Chroust	University of Southern Queensland, Australia
Paul M. Clarke	Lero, Dublin City University, Ireland
François Coallier	Ecole de technologie Superieure, Canada
Antonio Coletta	Qual. IT. Consulting, Italy
Onur Demirors	Middle East Technical University, Turkey
Fabrizio Fabbrini	Italian National Research Council, Italy
Dennis Goldenson	Software Engineering Institute, USA
Christiane Gresse von Wangenheim	Federal University of Santa Catarina, Brazil
Victora Hailey	VHG Corporation, Canada
Linda Ibrahim	Enterprise SPICE, USA
Jørn Johansen	Whitebox, Denmark

Ravindra Joshi	Whitebox, Denmark
Ho-Won Jung	Korea University, South Korea
Giuseppe Lami	National Research Council, Italy
Marion Lepmets	SoftComply, Ireland
Catriona Mackie	BT, UK
Antonia Mas Pichaco	Universidad de les Illes Balears, Spain
Fergal McCaffery	Dundalk Institute of Technology, Ireland
Tom McBride	University of Technology Sydney, Australia
Antoni Mesquida	Universidad de les Illes Balears, Spain
Antanas Mitasiunas	Vilnius University, Lithuania
Takeshige Miyoshi	Miyoshi Art of Software Process Inc., Japan
Risto Nevalainen	FiSMA Association, Finland
Mark Paulk	The University of Texas at Dallas, USA
Saulius Ragaisos	Vilnius University, Lithuania
Alain Renault	Luxembourg Institute of Science and Technology, Luxembourg
Patricia Rodriguez-Dapena	SoftWcare SL, Spain
Clenio Salviano	CenPRA, Brazil
Jean-Martin Simon	CGI Business Consulting, France
Fritz Stallinger	Software Competence Center, Austria
Timo Varkoi	Spinet Oy, Finland
Bharathi Vijayakumar	Wipro Technologies, India
Larry Wen	Griffith University, Australia
Murat Yilmaz	Çankaya University, Turkey

Local Organizing Committee

Antonia Mas	University of the Balearic Islands, Spain
Antoni Mesquida	University of the Balearic Islands, Spain
Milos Jovanovic	University of Novi Sad, Serbia

Acknowledgments

The conference organizers wish to acknowledge the assistance and support of the SPICE User Group, the SPICE 2017 Program Committee, and the reviewers in contributing to a successful conference.

Contents

SPI in Various Settings

SPI and Gamification

SPI Case Studies

SPI in Agile Approaches

NDT-Agile: An Agile, CMMI-Compatible Framework for Web Engineering

Carlos J. Torrecilla-Salinas[1(✉)], Tatiana Guardia[1], Olga De Troyer[2], Manuel Mejías[1], and Jorge Sedeño[1]

[1] IWT2 Group, Universidad de Sevilla, Seville, Spain
{carlos.torrecilla, tatiana.guardia, jorge.sedeno}@iwt2.org, risoto@us.es
[2] Department of Computer Science, Vrije Universiteit Brussel (VUB), Brussels, Belgium
Olga.DeTroyer@vub.ac.be

Abstract. Agile and Web Engineering show important synergies, making Agile a common approach for Web development. Besides, several initiatives emerged to support CMMI-DEV within Agile, where CMMI-DEV aims to improve organizations' software development process. An approach integrating Agile, Web and CMMI-DEV might be of great value, since they might allow Web development teams to use Agile, as well as progress through CMMI-DEV maturity levels. For this purpose, we developed *NDT-Agile*, an NDT-based Agile framework to achieve the goals of CMMI-DEV in the context of Web Engineering. It was developed by mapping Agile practices to the goals of CMMI-DEV so as to identify existing gaps. Next, we searched for suitable Agile practices to cover the gaps and integrated them into a framework called *NDT-Agile*, which was validated using an expert-judgment technique: the Delphi method. This paper describes how we integrated Agile and CMMI-DEV into a Web Engineering framework. Besides, it also analyzes its initial evaluation, together with a first tool developed to support it.

Keywords: Agile · Scrum · CMMI · Web engineering · Expert judgment

1 Introduction

Agile methodologies, i.e. those that can be grouped under the principles and values described in the Agile manifesto [1], emerged as an alternative to classic software development approaches, which were frequently based on heavy up-front planning and on freezing requirements before the development started. Agile brings a completely different view on how to handle and approach requirements [2]. It is based on improved communication, close collaboration with business representatives and reduced delivery cycles, among other elements [1]. Several approaches can be found within the label Agile, like Scrum [3], eXtreme Programming (XP) [4] or Kanban [5], being Scrum and XP the most popular ones [6, 7].

Web Engineering has established itself as the field of Software Engineering in charge of developing Web Systems, those conceived, developed, deployed and used on

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 3–16, 2017.
DOI: 10.1007/978-3-319-67383-7_1

the Web [8]. Several methodological approaches have been proposed for Web Engineering, such as Navigational Development Techniques (NDT) [9]. In turn, NDT proposes a Model-Driven approach to Web Engineering. NDT is a methodology that focuses on the first phases of the Web development lifecycle and utilizes a bottom-up process. It uses a highly-detailed requirement gathering phase guided by objectives with three sub-phases: requirements capturing, definition and validation. It is important to note that NDT was not developed bearing the Agile approach in mind.

Both Agile and Web Engineering emerged simultaneously and independently. However, they show great synergies [10]. As such, several Web development teams have already applied an Agile approach.

CMMI (Capability Maturity Model – Integration) [11] is a well-known approach designed to improve organizations' processes. Out of the different maturity models proposed by CMMI, CMMI-DEV (Capability Maturity Model – Integration for Development) is the one focusing on software development. Currently, CMMI is used in more than 5,000 companies [12]. The progress through the different CMMI maturity levels is normally associated with increases in quality and customer satisfaction [13].

During several years, the approaches proposed by Agile and by CMMI were seen as opposite and even contradictory [14], but after that initial reluctance period, initiatives emerged from both sides trying to find common grounds [15]. Recently, we can find several proposals trying to combine these approaches, both for generic software development projects [16] as well as for Web specific projects [17].

Summarizing all the aforementioned elements, we can conclude that an approach based on Agile principles that allows progressing through the different CMMI-DEV maturity levels, simultaneously supporting Web specificities, will be of great value as, on the one hand, it will enable organizations to keep using an Agile approach for their Web development projects but, on the other hand, it will ensure repeatability and institutionalization by means of the process improvement approach carried out by CMMI.

Based on the foregoing arguments, we have developed *NDT-Agile*, a framework conceived to help organizations achieve the specific and generic goals of CMMI-DEV in the context of Web Engineering, while keeping agility and ability to respond to changes. This paper has the following objectives: (i) present how agility, CMM-DEV and Web Engineering have been integrated into a coherent framework, called *NDT-Agile*; (ii) describe the assessment of *NDT-Agile* by means of an expert-judgment process based on the Delphi method [18]; (iii) introduce a first version of a tool to support *NDT-Agile;* and (iv) draw relevant conclusions and present further lines of research.

To achieve the listed objectives, the paper is organized as follows: after this introduction, Sect. 2 asks the research question and describes the utilized research approach. Section 3 discusses related work. Then, Sect. 4 introduces *NDT-Agile* by describing its main elements. Afterwards, Sect. 5 presents the expert-judgment validation process, as well as the developed supporting tool and, finally, Sect. 6 states the main conclusions and further lines of research.

2 Research Question and Research Method

The main research question that we addressed in our research was: "*Can we develop an Agile approach compatible with CMMI-DEV and usable for organizations developing Web systems?*" To answer this question, we asked a few concrete research sub-questions:

- **RQ1:** *What are the existing gaps between the current most used Agile approaches and the specific and generic goals of CMMI-DEV for Web systems?*
- **RQ2:** *Are there any existing Agile techniques to cover those gaps, in case they exist?*
- **RQ3:** *How can we combine the characteristics of the most used Agile approaches with the Agile techniques identified in RQ2 in a single coherent framework suitable for Web systems development?*
- **RQ4:** *How can we validate this framework, in case it can be developed?*

Once the research questions were asked, the next step consisted in defining a suitable research approach, which finally comprised the following steps:

- **Perform a gap analysis:** To distinguish if the existing and most popular Agile approaches can cover the different specific and generic goals of CMMI-DEV maturity level. As previously stated, Scrum and XP are the most used Agile approaches [6] whose practices are mapped to CMMI-DEV. The results of our gap analysis will be presented in Sect. 3.
- **Identify suitable Agile techniques to cover the gap:** To identify goals not covered by Scrum or XP practices by means of the gap analysis. The next step consisted in searching (in existing Agile literature) for other suitable Agile practices to cover the gaps. Results of this exercise will be also presented in Sect. 3.
- **Combine the identified techniques in a single coherent framework:** To define a coherent framework, named *NDT-Agile*, where the identified Agile practices should be combined in a suitable way, avoiding duplicities, gaps or contradictions. *NDT-Agile* will be presented in detail in Sect. 4.
- **Validate the proposed framework:** To validate the proposed framework by means of an expert judgment process based on the Delphi method [18], before performing real-life experiments. The goal of this expert evaluation was to obtain an initial validation of the framework. This process will be presented in Sect. 5.
- **Develop an initial version of a tool to support the framework:** To develop a first version of a supporting tool to backup future deployments of the framework. Such tool will be introduced in Sect. 5.

3 Related Work

This section summarizes related work that was collected by means of a Systematic Literature Review (SLR) based on Kitchenham's approach [19]. This process is described in more detail in [17]. The main goal was to identify previous works tackling the relations among Agile, Web and CMMI. From the results of the SLR, different types of papers were identified such as: other existing SLRs in the context [16]; papers tackling different angles of the research problem; and theoretical studies or case studies coming from both Agile [20–22] and CMMI side [23], some of them including the Web perspective [24–26], but some others not [20, 22]. In this section, we focus on those works that performed a gap analysis or a mapping between Agile and CMMI, regardless of whether or not they considered the Web perspective.

In [21], a Scrum-based model named Model C-S is presented. It maps the specific practices of CMMI levels 2 and 3 to Scrum ones. This model includes 123 practices, but excludes some CMMI-DEV process areas linked to organizational issues. The work comprises a mapping describing which practices are fully or partially covered, or they are not covered at all, and some ad-hoc modifications to Scrum. Besides, the proposed model incorporates supporting elements to deploy and assess the model together with two case studies.

Further on, [22] assesses, from a theoretical point of view, whether the standard Scrum practices can cover the goals of a set of CMMI-DEV process areas from maturity levels 2, 3 and 4 (those linked to project management). The work presents an analysis of the coverage provided by Scrum to 22 of CMMI-DEV practices, establishing whether they are fully or partially covered, or they are not covered at all.

In [20], a theoretical study on whether Scrum standard practices can cover the goals of a set of CMMI-DEV level 2 process areas is presented. It particularly analyzes Project Planning (PP), Project Monitoring and Control (PMC) and Requirements Management (REQM). It also includes a case study based on an internal project assessment.

In [27], the coverage provided by XP practices to CMMI levels 2 and 3 process areas is studied from a theoretical point of view. From its conclusions, we took out that XP supports most level 2 practices and some level 3 ones. The paper also highlights some limitations to CMMI coverage depending on the project size.

Additionally, [28] maps specific goals of CMMI maturity level 3 process areas to three different Agile methods (Scrum, XP and Kanban). Then, it evaluates which of the practices proposed by the analyzed Agile methods can cover the different goals of CMMI and provide a percentage of coverage to each of the analyzed techniques. The main conclusion of this work is that there is compatibility between Agile approaches and CMMI level 3, as many of the goals of maturity level can be covered. Finally, the paper includes a case study to validate the proposal.

After this review process, we concluded that all the above-described works focus on generic development and do not consider Web specificities. Moreover, we confirmed that all of them are partial, not presenting a full gap analysis or a complete mapping to all CMMI-DEV maturity levels' process areas.

As mentioned in Sect. 2, the first step in our research was conducting a gap analysis to identify whether the existing and most popular Agile approaches can cover the different specific and generic goals of CMMI-DEV maturity level. This analysis specifically included Web specificities. In [26], we compared Scrum practices with the goals of CMMI-DEV maturity level 2. This work analyzed theoretically the gap between Scrum practices and those of CMMI-DEV level 2, concluding that, even though there is no full coverage between both, they are highly compatible. In this paper, we also included a proposal to extend Scrum with the aim to cover the identified gap. Later, in [25], we proposed a mapping between Agile practices (including Scrum and XP standard practices, but going beyond them) and goals of CMMI-DEV maturity level 3. The paper settles that Agile techniques and CMMI-DEV are still highly compatible. Finally, in [24] we performed a gap analysis between Scrum and XP, and the proposed goals of CMMI-DEV maturity levels 4 and 5. The main conclusion was that Scrum or XP standard practices do not cover CMMI practices. Based on that assumption, we identified a set of Agile practices that could be suitable to cover the gap. Table 1 summarizes the identified coverage (from [24–26]) to the different CMMI-DEV maturity levels. To obtain the percentage of coverage, the number of CMMI-DEV specific practices fully covered by Scrum/XP standard practices of a particular maturity level was calculated and then divided by the total number practices defined in the maturity level.

Table 1. Identified coverage of Scrum/XP per CMMI-DEV maturity level.

CMMI-DEV level 2		CMMI-DEV level 4	
Approach	Coverage	Approach	Coverage
Scrum	72.2%	Scrum	0.0%
XP	66.7%	XP	0.0%
Combined Scrum/XP	92.6%	Combined Scrum/XP	0.0%
CMMI-DEV level 3		CMMI-DEV level 5	
Approach	Coverage	Approach	Coverage
Scrum	34.8%	Scrum	0.0%
XP	54.7%	XP	0.0%
Combined Scrum/XP	60.5%	Combined Scrum/XP	0.0%

Table 2 describes the identified and proposed extensions spotted in [24–26] to complement Scrum and XP with the aim to cover all the goals of CMMI-DEV. It also indicates whether the proposal is either an existing Agile practice (and then points to a reference describing it), or it is just an ad-hoc modification:

4 *NDT-Agile*: An Agile CMMI-Compatible Framework for Web Engineering

In the previous section, we identified works related to RQ1 and RQ2, including our own gap analysis and mapping exercise [24–26]. We also identified a suitable list of Agile practices or ad-hoc modifications that simultaneously support Web specificities and all the specific and generic goals of CMMI-DEV. Nevertheless, a list of practices is not useful for organizations that focus on looking for a coherent framework to implement and customize. In this section, we present *NDT-Agile*, an Agile framework built upon the conclusions of the gap-analysis and on top of Scrum and XP practices, including all identified proposed extensions listed in Table 2. *NDT-Agile* also supports Web specificities by integrating NDT (i.e. a Web development methodology) and incorporating it into an Agile lifecycle. The description looks at the way in which CMMI-DEV is supported. *NDT-Agile* is composed of 3 main components (Fig. 1):

Table 2. Identified proposed extensions to Scrum/XP standard practices.

CMMI-DEV level 2 – Proposed extensions	
Extension	Type
Sprint 0	Agile practice [26]
Establish measurement objectives, how to measure them and how to store measures and collect data during the project	Ad-hoc modification
Establish how, when and where to store the project data and use the selected sources during the project	Ad-hoc modification
Establish how to communicate and manage the project data and follow the agreed approach during the project	Ad-hoc modification
Establish quality objectives, briefly documenting the agreements	Ad-hoc modification
Agile contracts techniques	Agile practice [29]
CMMI-DEV level 3 – Proposed extensions	
Extension	Type
Agile Project Management	Agile practice [30]
Scrum at Enterprise Level	Agile practice [31]
Lean Software Development	Agile practice [32]
Agile Risk Management	Agile practice [33]
CMMI-DEV level 4 – Proposed extensions	
Extension	Type
Performance and KPI baselines	Ad-hoc modification
Adapt the process to achieve desired quality and performance objectives	Ad-hoc modification
Select measures and techniques to be used for quantitative management	Ad-hoc modification
Use Agile Performance Indicators	Agile practice [34]
Agile EVM	Agile practice [35]
CMMI-DEV level 5 – Proposed extensions	
Extension	Type
Lean Software Development	Agile practice [32]

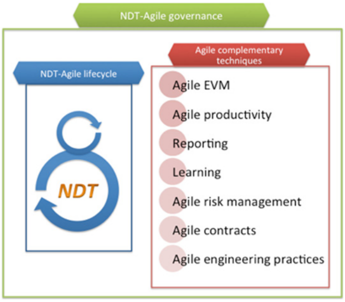

Fig. 1. *NDT-Agile* Components

- ***NDT-Agile* lifecycle**, an iterative and incremental lifecycle that describes the way projects are identified, planned, approved and developed, and which encapsulates NDT techniques. It focuses on covering the lifecycle related to the goals of CMMI-DEV levels 2 and 3.
- **Agile complementary techniques**, based on techniques identified in the gap analysis and complementing the framework beyond the scope of a project lifecycle. They cover the remaining not organizational-related goals of CMMI-DEV.
- **NDT-Agile governance,** which wraps the previous two elements and ensures a proper framework rollout, customization and improvement. It covers those goals of CMMI-DEV levels 4 and 5 that have an organizational dimension.

4.1 *NDT-Agile* Lifecycle

NDT-Agile lifecycle is used to manage projects (identify, plan and execute them) [35]. It comprises two main phases and Fig. 2 depicts the lifecycle:

- **Project launching**, which is the only non-iterative phase of the framework, where an initial plan is developed by means of Agile estimation techniques [36] combined with Agile Project Management inception techniques [30]. It is presented in the form of an Agile project charter for the organization's management to approve.

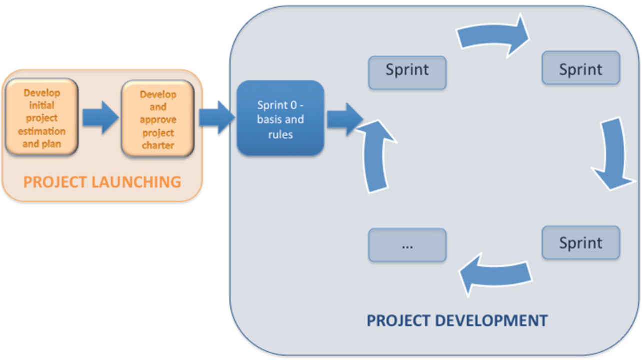

Fig. 2. *NDT-Agile* lifecycle [35]

- **Project development**, is an iterative phase that, based on the Scrum lifecycle, includes all identified modifications to achieve most project managements related to the goals of CMMI-DEV levels 2 and 3. After a ground-setting *Sprint 0*, a succession of Sprints is run with the aim to develop the project. The initial plan is adjusted Sprint-by-Sprint in order to ensure that business priorities are always identified and implemented. During this phase the requirements engineering, definition and validation proposed by NDT take place.

4.2 Agile Complementary Techniques

Agile complementary techniques were incorporated into our framework aiming to cover all the goals of CMMI-DEV that go beyond the scope of a project lifecycle and mainly come from the identified list of Agile practices resulting from the gap analysis. We identified and included a total of seven complementary techniques:

- **Agile EVM** [35]: It provides an Agile way to control project constraints like budget and schedule without including extra overhead. They are included to cover the remaining goals of Quantitative Project Management (QPM) process area not covered by the standard Scrum/XP practices.
- **Agile productivity metrics** [34]: They are proposed to cover the goals of OPP (Organizational Project Performance), helping to measure consistently teams' productivity and ensure continuous improvement.
- **Agile reporting:** It is established to cover the generic goals of CMMI-DEV, as described in CMMI-DEV standard, and propose an Agile approach so as to enhance communication with the stakeholder. It includes classic Agile elements such as burn-down or burn-up charts.
- **Agile Learning:** Coming from the Lean approaches [32], it is proposed to fully cover the goals of Organizational Training (OT), Causal Analysis and Resolution (CAR) and Organizational Project Management (OPM). It contains techniques to ensure both team and organization improvements, by means of elements like retrospectives or communities of practices.
- **Agile risk management** [33]: It is included to meet the specific goals of RSKM (Risk Management) and provide Agile projects with explicit risk management capabilities without extra overhead.
- **Agile contracting** [29]: It is proposed to cover the goals of Supplier Agreement Management (SAM) and as a way to ensure an Agile relation with providers. That guarantees that risk is well balanced, thus all parties gain with such a relationship.
- **Agile engineering practices** [4]: They mainly come from XP and are proposed to cover all CMMI-DEV engineering process areas, including Agile design, test and validation elements.

4.3 *NDT-Agile* Governance

NDT-Agile governance is the third component of the framework. It is proposed to cover the goals of level 4 and 5 process areas. It is based on Schwaber's proposal to scale Scrum to organizational levels [31] and prescribe the establishment of a governance body (named Enterprise Transition Team), which is set at organization level. It comprises the following main objectives:

- Tailor the framework according to the organization's specific needs.
- Define the different organizational assets (tool or lessons learnt, among others).
- Establish project baselines and define organizational KPIs.

The proposed governance body also ensures that Agile practices like Scrum of Scrums (in order to coordinate the different existing Agile teams) or the maintenance of an organization wide product backlog (to have a view of the progress at organizational level) are established.

5 Validation of the Approach and Supporting Tool

In this section, we explain how we carried out an initial validation of our proposal. For this purpose, an expert-judgment exercise based on the Delphi method was performed (Sect. 5.1). Furthermore, we also describe the first version of the supporting tool that enabled the framework deployment in practice (Sect. 5.2).

5.1 Expert-Judgment Process

As the implementation of new methods or frameworks within organizations always implies economical risks and presents organizational challenges, companies may be reluctant to incorporate them. If an initial validation conducted by a set of well-known experts is presented beforehand, some of these reluctances can be overcome and organizations might be more willing to experiment with the new working methods or frameworks. One of these expert-judgment techniques is the Delphi method [18], which consists in a panel of experts who, by means of structured and anonymous questionnaires and a series of rounds, reach consensus on a specific topic.

In order to validate *NDT-Agile* proposal using the Delphi method, a panel of 20 experts in one or more of the analyzed fields, coming from 8 different countries, was created and three consecutive rounds took place. The questionnaire used was composed of 21 statements, distributed among 4 different domains (i.e. dimensions) as follows:

- **Agile** dimension: 6 statements were used to assess the agility of the framework.
- **CMMI** dimension: 5 statements were used to evaluate the compliance with the different goals of CMMI maturity levels.
- **Web** dimension: 7 statements were provided to supervise the support given to Web specific characteristics.
- **Framework** dimension: 3 statements were used to test the internal coherence of completeness of *NDT-Agile*.

The questionnaire was made available to the experts in three rounds that were organized between February and June 2016. In each round, experts were asked to express their agreement with each of the statements by means of a Likert scale [37] ranging from "Complete disagreement" (value 1) to "Complete agreement" (value 5).

Two types of analysis were conducted in order to evaluate the results of the different rounds:

- **Descriptive analysis**, which assessed the level of agreement of the experts with the proposed statements and the internal grade of consensus reached by the panel by means of calculating the mean, median and standard deviation of the grades given to each statement, as well as analyzing the experts' textual comments.
- **Homogeneity and concordance analysis**, which used statistical tools like Chronbach's alpha [38], Kendall's W [39] and Simple Correspondence Analysis [40], calculated by means of R [41], to check the degree of consensus and stability of the panel's opinion on the analyzed subject through the different rounds.

In order to interpret the obtained results, we defined *strong agreement* of the panel experts on one of the statements if: the mean of the given grades was above 3.7 (in a scale ranging from 0 to 5, being 5 the maximum value), the median was 4 (representing "Agreement") and at least 60% of the raters' score was 4 or 5 ("Agree" or "Strongly agree"), with a minimum of 12 experts providing an opinion. We also defined *slight agreement* on one of the statements if: the mean of the given grades was between 3.5 and 3.7, with a median equal or higher than 3.5 and at least 45% of raters' score was 4 or 5, with a minimum of 12 experts providing an opinion. Figure 3 displays the obtained results after the third round distributed by the defined dimensions:

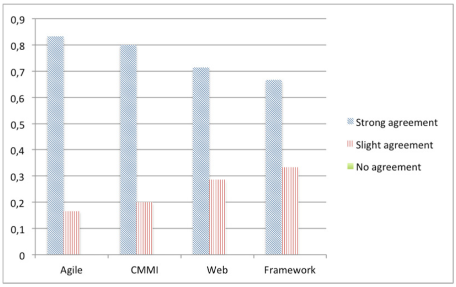

Fig. 3. Results by dimension

As Fig. 3 shows, a high level of agreement was achieved for all the four analyzed dimensions, ranging from more than 80%, in the case of Agile, to more than 65%, in the case of the Framework dimension. It must be pointed out that no disagreement with the overall *NDT-Agile* proposal was identified from the selected panel. Table 3 also presents the overall results of the Delphi method after the third round:

Table 3. Delphi method: overall results.

Level of agreement	Number of statements	%
Strong agreement	16	76.19%
Slight agreement	5	23.81%
Agile contracts techniques	0	0%

Out of the 21 proposed statements, 75% of them showed strong agreement whereas 25% showed slight agreement. The main questions marks stated by the panel were linked to feasibility of achieving the goals of CMMI-DEV level 5, support to Web systems' security and maintenance requirements within *NDT-Agile* and implementation of proposed governance model.

5.2 Supporting Tool

Finally, and as an essential element to support deployment, a first version of a supporting tool was developed. For that purpose, we conducted a comparison exercise among existing Agile project tools available in the market, in order to find out the most suitable one that could fulfil our needs, instead of developing a completely new tool from the scratch. After assessing Mantis, JIRA, Bugzilla and Redmine [42], we chose the last one due to its active community, its plugin mechanism and our previous knowledge of the tool. Table 4 presents the results of our analysis, including the assessment criteria used:

- **License/Cost:** Is it an Open Source tool or does it offer a "free of cost/community" version?
- **Plugin schema:** Does the tool offer a plugin/extension mechanism?
- **Community:** Does the tool have a well-established community?
- **Agile:** Are there any Agile extensions available to be used?
- **Integration:** Is it possible to integrate the tool with other tools in an ecosystem?

Table 4. Tool assessment: results.

Tool	License/Cost	Plugin	Community	Agile	Integration
Redmine	Open Source with Free/Community edition available	Yes	Yes	Yes	Yes
JIRA	Commercial tool	Yes	Yes	Yes	Yes
Bugzilla	Open Source with Free/Community edition available	No	Yes	Limited	No
Mantis	Open Source with Free/Community edition available	Yes	Yes	Yes	Yes

After identifying a suitable tool, we defined a series of epics and user stories that allowed us to support our proposal and, among them, select the ones to be included in the first version of the tool. Basically, we chose those related to *NDT-Agile* lifecycle support and Agile EVM calculations. Once the scope of the first version was clearly identified, we tried to achieve the desired functionality by two different paths:

- Configuring and customizing Redmine, which let us cover a significant amount of functionality without further development.
- Developing a custom-made plugin, in order to achieve the remaining functionality.

6 Conclusions and Future Work

This paper presented *NDT-Agile*, an Agile, CMMI-compatible framework for Web Engineering. Its inception process was based on a complete gap analysis between Scrum and XP and the different process areas of all CMMI-DEV maturity levels. The paper also provided an overview and justification of the initial validation of the framework, which was carried out by means of an expert-judgment process based on the Delphi technique. Finally, we briefly described the development of an initial version of a tool that could support the framework. In consequence, we were able to show that we could come up with *an integrated framework using an Agile approach, compatible with CMMI-DEV and usable for organizations developing Web systems*, providing an answer to our main research question. If we linked this work to the initially formulated research questions, we could state the following conclusions:

- *RQ1:* Several gaps were identified for all CMMI-DEV maturity levels. In the case of levels 2 and 3, we noticed that Scrum and XP are compatible with CMMI-DEV covering, either alone or combined, a significant amount of objectives. In the case of levels 4 and 5, we realized that there is no coverage at all, as those levels focus on organizational aspects, and Scrum and XP are more oriented towards operational ones.
- *RQ2*: A full list of complementary Agile techniques and modifications, suitable for Web systems, were identified to cover each of the gaps for all CMMI-DEV maturity levels.
- *RQ3*: We responded to this question by proposing *NDT-Agile*, a framework that, by means of an Agile lifecycle, ensures agility. It covers all remaining goals of CMMI-DEV, by including a set of complementary Agile techniques and a governance model, and supports Web specificities, by encapsulating NDT.
- *RQ4:* In order to perform an initial validation that afterwards would allow real-life deployments of the proposed framework, we conducted an expert judgment process based on the Delphi method. It offered promising results, as the identified panel agreed on the suitability of the approach. As a complementary element, we developed an initial version of a tool to encourage framework deployment.

As future lines of research we can highlight the improvement of the framework in those areas where experts expressed some concerns (such as security and maintenance practices, governance model or achievement of goals of CMMI-DEV level 5). Besides, the deployment of the framework in real-life projects and their assessment, via a formal SCAMPI process [43] or a self-assessment, remains to be done yet.

Acknowledgements. This research has been supported by the MeGUS project (TIN2013-46928-C3-3-R), the Pololas project (TIN2016-76956-C3-2-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish the Ministry of Economy and Competitiveness. We would also like to thank Dr. Diego Nieto Lugilde, Dr. Pedro Antonio García and Dr. Diego Torrecilla de Amo, from the University of Granada. Finally, we would like to show our gratitude to all experts participating in the Delphi method for their useful contributions.

References

1. Beck, K., et al.: Manifesto for Agile Software Development (2001). http://www.agilemanifesto.org. Accessed May 2017
2. Schön, E.M., Thomaschewski, J., Escalona, M.J.: Agile requirements engineering: a systematic literature review. Comput. Stand. Interfaces **49**, 79–91 (2017)
3. Sutherland, J., Schwaber, K.: The scrum guide: the definitive guide to scrum (2011). http://www.scrum.org/Scrum-Guides. Accessed May 2017
4. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston (2000)
5. Anderson, D.J.: Kanban - Successful Evolutionary Change for your Technology Business. Blue Hole Press, Sequim (2010)
6. Pikkarainen, M., et al.: The impact of agile practices on communication in software development. Empirical Softw. Eng. **13**, 303–337 (2008). Springer
7. VersionOne. 9th Annual State of Agile Survey (2015). http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf. Accessed May 2017
8. Deshpande, Y., Marugesan, S., Ginige, A., Hanse, S., Schawabe, D., Gaedke, M., White, B.: Web Engineering. J. Web Eng. **1**(1), 3–17 (2002)
9. Escalona, M.J., Aragón, G.: NDT: a model-driven approach for web requirements. IEEE Trans. Softw. Eng. **34**(3), 370–390 (2008)
10. Mendes, E., Mosley, N.: Web Cost Estimation: An Introduction. Web Engineering: Principles and Techniques, pp 182–202. IGI Global (2005)
11. CMMI Product Team: CMMI for Development, Version 1.3, November 2010. Carnegie Mellon University. Technical Report (2010). http://www.sei.cmu.edu/reports/10tr033.pdf. Accessed May 2017
12. CMMI Institute: Published Appraisal Results (2016). https://sas.cmmiinstitute.com/pars/. Accessed May 2017
13. Goldenson, D.R., et al.: Why make the switch? evidence about the benefits of CMMI. http://www.sei.cmu.edu/library/assets/evidence.pdf. Accessed May 2017
14. Staples, M., et al.: An exploratory study of why organizations do not adopt CMMI. J. Syst. Softw. **80**(6), 883–895 (2007)
15. Glazer, H., et al.: CMMI or agile: why not embrace both! November 2008. Carnegie Mellon (2008). http://www.sei.cmu.edu/reports/08tn003.pdf. Accessed May 2017
16. Silva, F.S., et al.: Using CMMI together with agile software development: a systematic review. Inf. Softw. Technol. **58**, 20–43 (2015)
17. Torrecilla Salinas, C.J., et al.: Agile, web engineering and capability maturity model integration: a systematic literature review. Inf. Softw. Technol. **71**(2016), 92–107 (2016)
18. Dalkey, N.C., Helmer, O.: An experimental application of the Delphi method to the use of experts. Manag. Sci. **9**, 458–467 (1963)
19. Kitchenham, B., et al.: Systematic literature reviews in software engineering – a systematic literature review. Inf. Softw. Technol. **51**(7–15), 2009 (2009)
20. Diaz, J., Garbajosa, J., Calvo-Manzano, Jose A.: Mapping CMMI level 2 to scrum practices: an experience report. In: O'Connor, Rory V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 93–104. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04133-4_8
21. Lukasiewicz, K., Miler, J.: Improving agility and discipline of software development with the scrum and CMMI. IET Softw. **6**(5), 416–422 (2012)
22. Marcal, A.S.C., et al.: Blending scrum practices and CMMI project management process areas. ISSE **4**, 17–29 (2008)

23. Jakobsen, C.R., Johnson, K.A.: Mature agile with a twist of CMMI. In: Proceedings of Agile Conference 2008, AGILE 2008, 04–08 August 2008, Canada. IEEE (2008)
24. Torrecilla Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M.: An Agile approach to CMMI-DEV levels 4 and 5 in Web development projects. In: Information Systems Development (ISD2016 Proceedings), Katowice, Poland (2014)
25. Torrecilla Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M.: Mapping agile practices to CMMI-DEV level 3 in web development environments. In: Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings), Varaždin, Croatia (2014)
26. Torrecilla Salinas, C.J., Escalona, M.J., Mejías, M.: A scrum-based approach to CMMI maturity level 2 in web development environments. In: Proceedings of International Conference on Information Integration and Web-based Applications and Services 2012, Bali, Indonesia, 3–5 December 2012. iiWAS, 12. ACM (2012)
27. Paulk, M.C.: Extreme programming from a CMM perspective. IEEE Softw. **18**(6), 19–26 (2001)
28. Bougroun, Z., et al.: The projection of the specific practices of the third level of CMMI model in agile methods: Scrum, XP and Kanban. In: Proceedings of 2014 3rd International Colloquium in Information Science and Technology (CIST), pp. 174–179 (2014)
29. Medinilla, A.: Contratos ágiles (2009). http://www.slideshare.net/proyectalis/090603-contratos-giles. Accessed May 2017
30. Highsmith, J.: Agile Project Management: Creating Innovative Products, Second Edition. Addison-Wesley, New York (2009)
31. Schwaber, K.: The Enterprise and Scrum. Microsoft Press, Redmond (2007)
32. Poppendieck, M., Poppendieck, T.: Lean Software Development. An Agile Toolkit. Addison-Wesley, Boston (2003)
33. Cohn, M.: Managing Risk on Agile Projects with the Risk Burndown Chart. http://www.mountaingoatsoftware.com/blog/managing-risk-on-agile-projects-with-the-risk-burndown-chart. Accessed May 2017
34. Downey, S., Sutherland, J.: Scrummetrics for hyperproductive teams: how they fly like fighter aircraft. In: Proceedings of the 45th Hawaii International Conference on System Science, Maui, Hawaii, USA, 4–7 January 2012 (2012)
35. Torrecilla Salinas, C.J., et al.: Estimating, planning and managing Agile Web development projects under a value-based perspective. Inf. Softw. Technol. **61**, 124–144 (2015)
36. Cohn, M.: Agile Estimating and Planning. Addison-Wesley, Englewood Cliffs (2005)
37. Likert, R.: A technique for the measurement of attitudes. Arch. Psychol. **140**, 5–55 (1932)
38. Cronbach, L.J.: Coefficient alpha and the internal structure of tests. Psychometrika **16**(3), 297–334 (1951)
39. Legendre, P.: Species associations: the Kendall coefficient of concordance revisited. J. Agric. Biol. Environ. Stat. **10**(2), 226–245 (2005)
40. Benzécri, J.P.: L'Analyse des Données. L'Analyse des Correspondances, vol. 2 (1973)
41. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2016). https://www.R-project.org/. Accessed May 2017
42. Redmine (2016). http://www.redmine.org. Accessed May 2017
43. SCAMPI Upgrade Team: Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A. Carnegie Mellon University (2011). http://www.sei.cmu.edu/reports/11hb001.pdf. Accessed May 2017

DevSecOps: A Multivocal Literature Review

Håvard Myrbakken and Ricardo Colomo-Palacios(✉)

Østfold University College, Remmen, 1757 Halden, Norway
{haavam,ricardo.colomo-palacios}@hiof.no

Abstract. Involving security in DevOps has been a challenge because traditional security methods have been unable to keep up with DevOps' agility and speed. DevSecOps is the movement that works on developing and integrating modernized security methods that can keep up with DevOps. This study is meant to give an overview of what DevSecOps is, what implementing DevSecOps means, the benefits gained from DevSecOps and the challenges an organization faces when doing so. To that end, we conducted a multivocal literature review, where we reviewed a selection of grey literature. We found that implementing security that can keep up with DevOps is a challenge, but it can gain great benefits if done correctly.

Keywords: DevSecOps · DevOps · Security · Multivocal literature review

1 Introduction

In recent years a large part of software development companies have changed focus from developing software as a product (SaaP), where companies developed the software and delivered a finished product to a customer that then installed and ran it locally, to develop software as a service (SaaS), where software is centrally hosted on a cloud infrastructure and accessed through for example a webbrowser [1], or other channels that delivers it directly to a customer's machine or device [2]. The use of it is then offered through licensing and subscriptions. With SaaS, the customers do not control the underlying cloud infrastructure or the application's functionality [1], as that is done by the provider. This gives the provider the opportunity to continuously improve and deliver their software without having to redistribute it to all their clients as they simply update the software on their own cloud infrastructure. This modern software engineering process of developing while continuously integrating and delivering software is complex. Continuous integration (CI) means to automatically integrate new code from several developers into the same version of the software and at the same time, check for errors [3]. Continuous Delivery (CD) means to deploy new software to production, with the differing factor from traditional software deployment being the frequency of deployment, which can happen multiple times every day [3]. "Continuous delivery enables businesses to reduce cycle time so as to get faster

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 17–29, 2017.
DOI: 10.1007/978-3-319-67383-7_2

feedback from users, reduce the risk and cost of deployments, get better visibility into the delivery process itself, and manage the risks of software delivery more effectively." [4]. These processes require a large number of tools and information systems [5]. These processes, tools and systems are often managed by independent operations teams [6]. Many challenges when implementing CI/CD resulted from lack of collaboration and communication between the operators and developers [2,3,6,7]. Attempts at overcoming these challenges have resulted in a concept, termed DevOps [2].

DevOps is described as the "conceptual and operational merging of development and operations' needs, teams, and technologies" [6]. This merging is meant to align the priorities of the development teams and operations teams so they work together towards a common goal of successful project execution [6] by cooperating on software development and deploying that software into production [2]. This can be done by involving operations in all development stages, by developers and operators collaborating to solve problems, make processes and products that can be automated, and agree on and develop metrics that everyone can make use of [8]. This reflects the four main principles of DevOps: culture, automation, measurement and sharing (CAMS) [2,4].

As DevOps has become more popular, many organizations are adopting the practices associated with it. However, a survey by the HPE Security Fortify team [9] from 2016 shows that while many believe that security should be a part of DevOps, security is not something many DevOps programs have included as part of their process. Gartner estimates in [10] that less than 20% of "enterprise security architects have engaged with their DevOps initiatives to actively and systematically incorporate information security into their DevOps initiatives". [10] points to management, developers, and operators viewing security as an inhibitor to the agility and speed required in DevOps practices, like CI and CD, as one reason for this.

The need for security in DevOps is met by DevSecOps. This concept is an attempt at creating and including modern security practices that can be incorporated in the fast and agile world of DevOps. It promotes an extension to DevOps' goal of promoting collaboration between developers and operators by involving security experts from the start as well [11].

Since DevSecOps is a new trend, it is important to obtain an overview of the practices and experiences accumulated on the subject. There is not a lot of research on DevSecOps, but a search of available literature shows: [12] is a study that through Internet artefacts and a survey looks at practitioners experiences with DevSecOps and the practices they perform, [11] is a systematic mapping (SM) study on what is being researched in the field and it showed research was being conducted on the aspects of "a definition, security best practices, compliance, process automation, tools for DevSecOps, software configuration, team collaboration, availability of activity data and information secrecy". A search for systematic reviews or mappings on continuous processes (CI/CD) used in DevOps, resulted in a several results. Examples are: [13] examines the impact agile release engineering (ARE) and the continuous processes involved had on software engineering. [14] maps literature related to CD and provides an analysis of the benefits and challenges related to CD. [15] uses a literature review to show differences in how CI is done for different cases. None of the literature we found gives a collected overview of DevSecOps and what it is.

To the best of authors' knowledge, there is not a systematic literature review on DevSecOps or a large body of scientific work related to DevSecOps. This absence of works devoted to the topic lead us to the need to work on the topic using a tool like multivocal literature review, intended to bridge the gap between professional and scientific literature. By this mean, authors examine the concept of DevSecOps, how it has evolved since it was first introduced, and the challenges and benefits DevSecOps brings to an organization.

The rest of this paper is structured as a systematic literature review. In Sect. 2 the methodology for the research is presented. In Sect. 3 we present the results from our study. In Sect. 4 we conclude on our paper, summarize the results and suggest future work.

2 Research Methodology

In this section, an overview of our research methodology is presented followed by an overview of the systematic approach used to gather relevant literature.

2.1 Multivocal Literature Review

After an initial search on literature to learn more on the topic of DevSecOps, we could not find a substantial body of academic research on the topic. We therefore decided to conduct a multivocal literature review (MLR). Multivocal literature is defined as all accessible literature on a topic [16]. This includes, but is not limited to: blogs, white papers, articles and academic literature. By using this variety of literature the results will give a more nuanced look at the topic, since it includes the voices and opinions of academics, practitioners, independent researchers, development firms and others with experience [16].

Previously published MLRs include but is not limited to: [17] is an MLR on automated software testing and the proposed guidelines from practitioners and researchers for when and what to automate. [18] is an MLR providing an overview of DevOps. [19] is an MLR on software test maturity assessments and test process improvement.

To the best of our knowledge, this is the first MLR on the topic although it is not the first for DevOps.

[20] points to the importance of MLRs in software engineering (SE) fields by stating that SE practitioners produces multivocal literature on a great scale, but that it is not published in academic forums. They mention however, that not including that literature in systematic reviews means researchers miss out on important current state-of-the-art practice in SE.

2.2 Research Questions

This MLR is conducted to obtain an understanding of what DevSecOps is, how it has evolved and the challenges and benefits of adopting such an approach. To specify the goal of this paper, four research questions were formulated:

Research Question 1: How does the literature define DevSecOps?
Research Question 2: What are the characteristics of DevSecOps?
Research Question 3: What are the main expected benefits and challenges of adopting DevSecOps?
Research Question 4: Since it was first mentioned, how has DevSecOps evolved?

2.3 Study Protocol

The study protocol describes the systematic way we found the literature used in our study. This section lists the databases used in the search, what search strategy was used to find related literature, the inclusion and exclusion criteria used to find the most relevant literature, and the process in which we catalogued the literature.

Databases. For this MLR we used Google's search engines to find relevant literature:

- Google Search (http://www.google.com/) to locate grey literature (white papers, blogs, articles etc.)
- Google Scholar (http://scholar.google.com/) to specifically locate available academic literature.

Google's search engines was chosen over more precise search engines (like Springer Link, ACM Digital Library, IEEE Explore etc.) because DevSecOps is a very new topic and very little academic research is available. We therefore knew beforehand that this literature review would rely mostly on the grey literature it would find, which Google's search engines would be able to locate.

Search Terms. DevSecOps is a new term based on adding the term "SECurity" to DevOps which stands for "DEVeloper" and "OPeration". There is not a consensus in the field on the ordering of the words, so the search terms must cover all possible permutations. The search string must also be made to find relevant literature according to the RQs. The search string used is therefore as follows:

```
("DevSecOps" OR "SecDevOps" OR "DevOpsSec") AND
("definition" OR "characteristics" OR "challenges" OR "benefits"
OR "evolution").
```

Study Selection. Once initial search results were retrieved, a procedure to exclude irrelevant papers were conducted using the following inclusion and exclusion criteria:

- Inclusion criteria:
  - Literature that explicitly discuss DevSecOps.
  - Literature that explicitly discuss DevOps and Security, particularly the challenges and benefits.
  - Literature discussing the present challenges to DevSecOps.
  - Literature discussing the benefits of DevSecOps.
  - Literature that discuss the definition of DevSecOps.
  - Literature published after 2014.
  - Include only the 5 first pages on Google Search.
- Exclusion criteria:
  - Literature that is inaccessible.
  - Results Google Search deems to similar to other results.
  - Vendors tool advertisements.

Search Procedure. The process is as follows: First we perform an advanced search in Google Search and Google Scholar. To let Google's search engine put primary focus on the different RQs, the term will be split into 5 parts, each focusing on all permutations of DevSecOps and one of the words related to the RQs. The 5 search strings can be seen in Fig. 1. For each search we then read the literature systematically applying the remaining inclusion and exclusion criteria, selecting only relevant literature for the primary study. The process is visualized in Fig. 1:

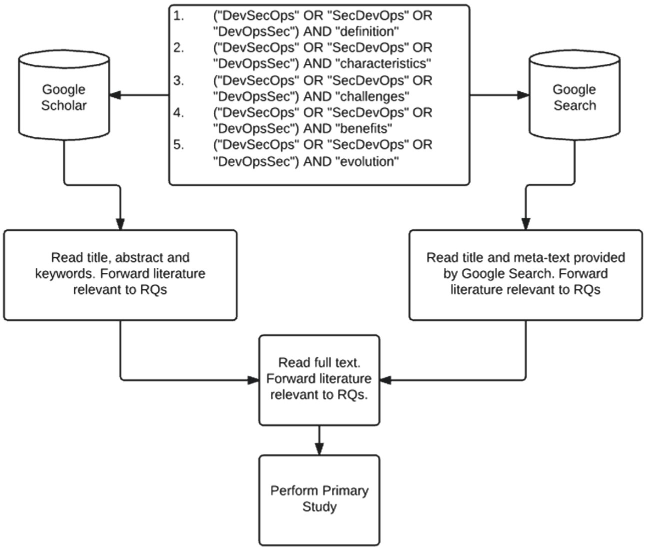

Fig. 1. An overview of the search process to find relevant literature for this study.

3 Results

In the following section we show the results from executing our search followed by our review of the literature in conjunction with our research questions.

3.1 Search Execution

The search was performed on during the first quarter of 2017. The initial search gave 284 results. After applying the inclusion and exclusion criteria 52 results were left. Table 1 summarizes the search:

Table 1. Summary of search results for primary study

Search engine	Initial results	Title, abstract, keywords and Meta text	Full text
Google scholar	34	4	2
Google search	250	62	50

3.2 RQ1 - How It Is Defined

In the literature we reviewed there seemed to be a consensus on what DevSecOps is seen as. DevSecOps is seen as a necessary expansion to DevOps, where the purpose is to integrate security controls and processes into the DevOps software development cycle [21] and that it is done by promoting the collaboration between security teams, development teams and operations teams [11].

3.3 RQ2 - DevSecOps Characteristics

When reviewing the literature, the features that stood out as characterizing DevSecOps were the principles seen as the basis and reasoning for DevSecOps and the practices used when implementing security into their software development processes.

Principles: The principles that characterize DevSecOps are based on DevOps and the CAMS principles [4,22], culture, automation, measurement, and sharing, but with the addition of adding security from the start:

<u>Culture:</u> A DevOps culture promotes collaboration between development teams and operation teams [4], where they all accept that they are responsible for delivering software to an end-user [2]. DevSecOps means to include collaboration with the security team as well as promote a culture where operations and development also work on integrating security in their work [11,23]. That means

involving the security team from the planning stages, and making sure everyone agrees that security is everyone's responsibilities [24,25]. To get everyone to think security, practitioners points to creating a common mindset with regards to success by developing a set a metrics that everybody agrees on and can use [25], promote customer focus by creating an alignment of business and security strategies to ensure just right and enough security that everyone in an organization can support and implement [26,27].

Automation: In DevOps the automation of build, deployment, and testing is important to achieve rapid development, deployment [4,22], and feedback from end-users [2]. DevSecOps promotes a focus on automating security as well, to be able to keep up with the speed and scale achieved by DevOps. The aim should be 100% automation of security controls, where the controls can be deployed and managed without manual interference [28]. It is important to implement automatic security in a way that does not hinder DevOps' agility in any way, which can cause friction [10,27,29].

Measurement: In DevOps measurements include monitoring business metrics such as revenue and key performance indicators, like the effect new releases have on the stability of a system, in order to know the current state and finding out how to improve it [2,4]. DevSecOps promotes the use and development of metrics that track threats and vulnerabilities throughout the software development process [10]. Automatic security controls throughout the software development process means metrics are available to track threats and vulnerabilities in real-time and that allows the organization to verify how good an application is on demand [22].

Sharing: In DevOps developers and operators share knowledge, development tools, and techniques to manage the process [2,4]. DevSecOps promotes the inclusion of the security team in the sharing promoted in a DevOps environment. By letting security teams know about the challenges faced by operators and developers, and vice versa, the security processes they develop will be improved [22].

Shift security to the left: In the traditional software development process, security is a step close to the end of the process. DevSecOps promotes a shift to the left for security, where it is to be included in every part of the software development process [23]. This means that security teams are involved from the very first planning step and is part of planning every iteration of the development cycle [29,30]. It also means security is there to help developers and operators on security considerations [24,28,31].

Practices: Several practices for DevSecOps were pointed to in the literature:

Threat modeling and risk assessments: Practicing secure DevOps means that organizations have to develop expertise and processes to best discover, protect against, and find solutions to threats and risks [32], preferably ahead of time [25]. Performing risk assessments from the first planning stage and continuously before every iteration is important as a way to prioritize risks, examine controls already in place and decide which are needed going forward [21,33]. Threat modeling is another method where you attack your system on paper early in the development cycle to identify how an attack can occur and where it is most likely to happen [34].

Continuous testing: Automatic security controls at every part of the software development process is important for security assurance and allows tests to continuously scan code for changes [32,34], continuously detect anomalies, and automatic rollback of code when needed [21,24].

Monitoring and logging: When automating security controls throughout the software developing process it is important for those involved to be able to generate evidence on demand that controls are working and that they are effective [35]. To that end, it is important to monitor every part of the inventory and to log every resource [21,25,33].

Security as code: This means to define security policies, for example integration testing, and network configuration and access, and write scripted templates or configuration files that can be implemented into the development process from the start of the project. These codified security policies can then be activated automatically according to schedules or be activated by user (simple push of a button), and be stored in a central repository for reuse on new projects [36].

Red-Team and security drills: To stay ahead of possible attackers, practitioners of DevSecOps create a Red-Team that runs security drill on the deployed software. They have the task of finding and exploiting vulnerabilities in the system [25,34]. This not only helps to find security flaws, but improves measurements, and helps the organization find solutions [26]. The point of the Red-Team is to have people that never claim something can't possibly happen.

3.4 RQ3 - Benefits

The following section provides an overview, according to the literature, of the benefits gained from DevSecOps and its practices:

Shifting Security to the Left: By involving security experts from the start of the development process it is easier to plan and execute integration of security controls throughout the development process without causing delays or creating issues by implementing security controls after systems are running [29].

Automating Security: This allows security controls to be fast, scalable and effective thus making it possible to keep a high pace for detecting errors, alerting about the errors, fixing the errors, finding countermeasures for future errors and forensics to identify why an error occurred [37]. This not only helps to lower risk and time spent on errors, but also makes it easier to understand risk and create policies and procedures [38]. The automation allows processes to be consistent and repeatable, with predictable outcomes for similar tests, it allows logging and documentation to be automatic [39] and letting security tests be run at the push of a button frees up developers time to write code instead of running tests [40]. This also reduces the risk for human error [39]. The ability to store security policy templates that is created during a development process in a central repository means that security teams don't need to manually configure every new environment when starting a new project which frees security experts from manual, repetitive and unproductive work [36].

Value: [38,41] points to how security missteps can be very expensive and that it is cheaper to implement security from the start than to wait for something to happen. [38] points to a survey that mentions how high-performance organizations spend 22% less time on unplanned work and rework. The ability to monitor and measure for security flaws early in the process ensures that bugs that prevent a delay in the deployment are caught and quantified [38]. This decreases the cost of making mistakes, finding them, and fixing them [36].

3.5 RQ3 - Challenges

The following section provides an overview of the challenges an organization faces from DevSecOps. The challenges are connected to the speed and agility needed not to slow down other DevOps practices, organizational changes, tools and practices:

Keeping up with DevOps: Using traditional, manual security methods heavily impairs the speed and agility of DevOps. This means security methods have to be more agile, and these agile security methods have to be understood by security teams and accepted by development teams [42] to make sure they contribute meaningfully to the DevOps movement without hampering their development speed and service delivery [43].

Organizational: Getting started with DevSecOps means the organization has to adopt change. Skills, culture, tools, processes, standards and practices must be considered as a possibility for implementing security [29]:

- There will be a need for skills in areas such as encryption and logging standards etc. [29].
- The organizational barriers between security teams and the rest of the organization must be broken down:
  - Developers and managers can be frustrated with the added time it takes to produce code, when adding security [21,33]. Developers and operators think of security as a hindrance to their goals, which is to deliver functionality fast, while security teams are focused on making sure the functionality is secure and robust [44].

  - The security teams not being properly trained on tools developers and operators use, hinders them from being able to integrate security in a transparent and understandable way for other users, which would limit collaboration between teams [31].
  - Organizations see security as a costly activity, and not something that generates revenue [32,35].
- There will be a need for new standards for security prevention, detection and response [29].

Tools and Practices: The dynamic environment when practicing DevOps means that security functionality has to be available in tools that work on the right platforms. There is a lack of available tools [21]. Any security functionality not automated in the available tools will create friction in the DevOps cycle. The users need to be properly trained when using advanced tools. [38] points to developers that had difficulties writing secure code because they couldn't use the tools efficient enough to keep up with DevOps' speed.

3.6 RQ4 - The Evolution of DevSecOps

The need for security to be integrated in DevOps was first mentioned in a blog by Neil MacDonald, a Gartner analyst, in a blogpost called "DevOps Needs to Become DevOpsSec" in 2012. DevSecOps has since become more and more acknowledged as a necessity. Table 2 shows the increase in number articles per year, which is evidence that awareness, recognition and use of DevSecOps is on the rise.

Table 2. Overview of number of results per year

Year	Number
Unknown	7
2014	2
2015	8
2016	27
2017 (first quarter only)	8

3.7 Limitations of Results

This research is based on multivocal literature, and most of the material has not been subject to the rigorous peer-review academic research usually is. The literature has instead consisted of blogs, white-papers, industry reports and academic research.

DevSecOps is a very new term, and the term has not even been agreed upon. It varies between SecDevOps, DevSecOps, DevOpsSec, Secure DevOps, and Rugged DevOps. In this research paper we have consistently used DevSecOps (with exception to where I am referring to other sources and their titles). The fact that it is as new as it is, means the results from this MLR can become outdated as best practices change.

4 Conclusion

This MLR presents the research we did on DevSecOps to find out how DevSecOps can be defined, what doing DevSecOps means for an organization in regard to what principles and practices they should adhere to, what challenges they would face attempting to adopt DevSecOps, the benefits if it's done successfully and how it has evolved from the need to implement security in DevOps to what could seem like a movement on its own.

We used Google Scholar and Google Search to locate literature and after applying our inclusion and exclusion criteria, 52 artefacts were found to be relevant to our search terms. Only 2 of those were academic research papers. The remaining 50 consisted of white papers, blogs and articles.

We found that DevSecOps is defined by many as the integration of security processes and practices into DevOps environments, that DevSecOps promotes a set of principles meant to shift the mindsets of all participants in the software development process so everyone participates and do what they can to ensure security in the project and a set of practices that can ensure security in the project based on the idea of planning and implementing security from the start and as code.

We identified a set of challenges and benefits to implementing DevSecOps. The challenges we identified should not be seen as deterrents to implementing DevSecOps, but a symptom of its youth. As DevSecOps matures, better methods, practices, tools etc. can probably overcome them. The benefits we identified indicates it is maturing, by for example resulting in less unplanned work and a decrease in manual labour.

As future work, it would be interesting to conduct surveys on organizations to possibly expand this study's coverage on DevSecOps. It is also of interest to investigate this study's suggested practices: observing practices effects on the surrounding environments (development, operations, business, customers) to find best practices. A possibility would then be to investigate and propose possible architectures or frameworks for implementing DevSecOps, [45] for example looks at continuous software engineering while using a microservices architecture which could be an alternative for "security as code" in DevSecOps.

References

1. Mell, P.M., Grance, T.: The NIST definition of cloud computing. Special Publications (NIST SP)-800-145, 7 P. NIST Definitions on Cloud Computing, September 2011
2. Fitzgerald, B., Stol, K.J.: Continuous software engineering: a roadmap and agenda. J. Syst. Softw. **123**, 176–189 (2017)

3. Svensson, R.B., Claps, G.G., Aurum, A.: On the journey to continuous deployment: technical and social challenges along the way. Inf. Softw. Technol. **57**, 21–31 (2015)
4. Humble, J., Joanne, M.: Why enterprises must adopt devops to enable continuous delivery. J. Inf. Technol. Manage. **24**, 7 (2011)
5. Hernantes, J., Ebert, C., Gallardo, G., Serrano, N.: Devops. IEEE Softw. **33**(3), 94–100 (2016)
6. Yankel, J., Cois, C.A., Connell, A.: Modern devops: optimizing software development through effective system interactions. In: 2014 IEEE International Professional Communication Conference (IPCC), pp. 1–7, October 2014
7. Callanan, M., Spillane, A.: Devops: making it easy to do the right thing. IEEE Softw. **33**(3), 53–59 (2016)
8. Spinellis, D.: Being a devops developer. IEEE Softw. **33**(3), 4–5 (2016)
9. Hewlett Packard Enterprise: Application security and devops. Technical report, Hewlett Packard Enterprise (2016)
10. MacDonald, N., Head, I.: DevSecOps: How to Seamlessly Integrate Security Into DevOps. Technical report, Gartner (2016)
11. Mohan, V., Othmane, L.B.: Secdevops: is it a marketing buzzword? - mapping research on security in devops. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 542–547, August 2016
12. Ashfaque, A., Rahman, U., Williams, L.: Software security in devops: synthesizing practitioners' perceptions and practices. In: Proceedings of the International Workshop on Continuous Software Evolution and Delivery, CSED 2016, pp. 70–76. ACM, New York (2016)
13. Oivo, M., Karvonen, T., Behutiye, W., Kuvaja, P.: Systematic literature review on the impacts of agile release engineering practices. Inf. Softw. Technol. **86**, 87–100 (2017)
14. Lwakatare, L.E., Teppola, S., Suomalainen, T., Eskeli, J., Karvonen, T., Kuvaja, P., Verner, J.M., Rodríguez, P., Haghighatkhah, A., Oivo, M.: Continuous deployment of software intensive products and services: a systematic mapping study. J. Syst. Softw. **123**, 263–291 (2017)
15. Ståhl, D., Bosch, J.: Modeling continuous integration practice differences in industry software development. J. Syst. Softw. **87**, 48–59 (2014)
16. Ogawa, R.T., Malen, B.: Towards rigor in reviews of multivocal literatures: applying the exploratory case study method. Rev. Educ. Res. **61**(3), 265–286 (1991)
17. Garousi, V., Mäntylä, M.V.: When and what to automate in software testing? a multi-vocal literature review. Inf. Softw. Technol. **76**, 92–117 (2016)
18. Junior, H.J., de França, B.B.N., Travassos, G.H.: Characterizing devops by hearing multiple voices. In: Proceedings of the 30th Brazilian Symposium on Software Engineering, SBES 2016, pp. 53–62. ACM, New York (2016)
19. Felderer, M., Garousi, V., Hacaloğlu, T.: Software test maturity assessment and test process improvement: a multivocal literature review. Inf. Softw. Technol. **85**, 16–42 (2017)
20. Felderer, M., Garousi, V., Mäntylä, M.V.: The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, EASE 2016, pp. 26:1–26:6. ACM, New York (2016)
21. Shackleford, D.: A devsecops playbook. SANS Institute InfoSec Reading Room. A DevSecOps Playbook, March 2016
22. Vonnegut, S.: 4 keys to integrating security into devops (2016), https://goo.gl/aZ0S3i

23. Lietz, S.: Shifting security to the left (2016), https://goo.gl/sbheKS
24. Bledsoe, G.: Getting to devsecops: 5 best practices for integrating security into your devops (2016), https://goo.gl/ZPzgxa
25. Lim, F.: Devsecops is the krav maga of security (2016), https://goo.gl/BH4MS2
26. Lietz, S.: Principles of devsecops (2015), https://goo.gl/N8zcXV
27. Greene, T.: What security teams need to know about devops (2016), https://goo.gl/c8VOn4
28. Anonymous User. Security breaks devops - here's how to fix it (2015). https://goo.gl/Yr1jk3
29. Shackleford, D.: The devsecops approach to securing your code and your cloud. SANS Institute InfoSec Reading Room A DevSecOps Playbook, February 2017
30. Caum, C.: Getting started with policy-driven development and devsecops (2016). https://goo.gl/AevVcX
31. Whitehat Security. Devops invites security to "join the party" (2016), https://goo.gl/spj0wK
32. Hornbeek, M.: Devops makes security assurance affordable (2015), https://goo.gl/g0iKfZ
33. Lindros, K.: How to craft an effective devsecops process with your team (2016), https://goo.gl/ppWtjx
34. Romeo, C.: The 3 most crucial security behaviors in devsecops (2016), https://goo.gl/FJKuYQ
35. Cureton, A.: Building security into devops: is devsecops the beginning of the future? (2017), https://goo.gl/Npv2Py
36. McKay, J.: How to use devsecops to smooth cloud deployment (2016), https://goo.gl/vqoh4L
37. Amazon Web Services. Introduction to devsecops on AWS (2016), https://goo.gl/wxl3YM
38. Francis, R.: 7 ways devops benefits cisos and their security programs (2015), https://goo.gl/RxieGr
39. Wallgreen, A.: Devsecops: 9 ways devops and automation bolster security, compliance (2015), https://goo.gl/RyA9QZ
40. Rotenberg, M.: 7 essential steps to devsecops success (2016), https://goo.gl/JAOQlF
41. Paul, F.: Secdevops: injecting security into devops processes (2015), https://goo.gl/Eul2fn
42. Rohr, M.: Agile security and secdevops touch points (2015), https://goo.gl/peuqpS
43. Goldschmidt, M., McKinnon, M.: Devsecops - agility with security. Technical report, Sense of Security (2016)
44. Elder, M.: Security considerations for devops adoption (2014), https://goo.gl/b0CStP
45. Clarke, P.M., O'Connor, R.V., Elger, P.: Continuous software engineering–a microservices architecture perspective. J. Softw. Evol. Proc. 2017, e1866 (2017)

Towards the Development of a Sequential Framework for Agile Adoption

Miloš Jovanović[1,2], Antoni-Lluís Mesquida[1](✉), Antònia Mas[1], and Bojan Lalić[2]

[1] Department of Mathematics and Computer Science, University of the Balearic Islands, Cra. de Valldemossa, km 7.5, Palma de Mallorca, Spain
{antoni.mesquida, antonia.mas}@uib.es

[2] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia
{milosj, blalic}@uns.ac.rs

Abstract. This research demonstrates the first steps towards the creation of a generic sequential Agile adoption framework. The presented Framework is the result of a detailed analysis of academic literature and industrial reports, and a multi-case study conducted in three large enterprises in Spain and Serbia. The proposed Agile adoption framework is composed of three main sequential phases for Agile method adoption process: Preparation, Transformation and Agile organisation. Preparation, the first phase of the framework, is developed to the highest level of detail and validated in three case companies. The main contribution of the paper is the proposed framework, from which the first phase is ready to be used by practitioners, and second and third phases are useful for academic society and they can be developed and validated further in the future. Integrated list of contingency factors, and list of situational factors, may be used by practitioners independently of using the generic Framework for Agile adoption presented in the paper.

Keywords: Agile adoption · Agile transformation process · Process improvement · Agile software development

1 Introduction

Agile methods, principles, values and practices are well known to the actors of software industry, but the problem starts when organisations start using and implementing them in practice [1]. Agile methods should be well tailored and integrated in the business process of the organisations [2, 3]. Even though there exist many guidelines and frameworks for Agile method adoption, organisations have problems with the selection of the most convenient Agile method and with the general initiation of the Agile transformation process. Each organisation as a whole, and each project being implemented in the organisation, have different circumstances and Agile transformation process is hard to standardize and offer a unique framework suitable for all the potential

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 30–42, 2017.
DOI: 10.1007/978-3-319-67383-7_3

cases. This problem was the main motivation for us to pursue this investigation and to try to contribute and explore this field more in detail.

The authors decided to conduct a research with the main objective of creating Agile adoption frameworks that could be used as guidelines by the organisations to facilitate the introduction of Agile methods. In order to reach our research objective, we designed several research steps to be implemented. First step was to explore available Agile adoption frameworks in the academic literature, as well as experience reports and technical guidelines used in the industry. After collecting and reviewing available frameworks, in a second research step, a multi-case study was conducted in three companies in two different countries (Spain and Serbia). This multi-case study embedded three main phases: conducting interviews, creation of an Agile adoption framework and validation of the framework.

The identified Agile adoption frameworks, which are introduced in the Sect. 2 of the paper, differ greatly in terms of their focus and the approach to the introduction of the Agile methods in the organisations. Therefore, we identified the need to further investigate the Agile adoption process in organisations and to make one step forward in creating a conceptual framework which would provide a more general (high-level) solution for companies pursuing Agile transformation. Our aim was to identify the typical lifecycle of an Agile transformation process and to be able to clearly define the main activities implemented in each of the lifecycle phases. Such an Agile adoption framework would help companies to identify their current phase of Agile transformation and to choose the most appropriate activities to be implemented to continue to the next phase.

For the authors, it was important to systematically consolidate current research in the initial version of the proposed Agile adoption framework, which is presented in Sect. 3. This initial version was verified and upgraded in the three companies in which the survey was conducted. For each case, the elements of the framework were adapted to the identified situation. Respondents had the opportunity to describe the process of introducing Agile methods in their company and to review and propose improvements to our proposed framework for the introduction of Agile methods in the enterprise. Two case companies have recently adopted Agile methods so in this research phase 1 of the framework is validated in the companies while phases 2 and 3 are described based on available data and should be explored further. The review and validation issues are detailed in Sect. 4, before Sect. 5 that concludes the article and opens discussion about next steps of this research.

2 Agile Adoption Frameworks and Guidelines for Agile Transformation Process

As a first step of our research, we intended to identify the existing Agile adoption frameworks in the academic literature, as well as other systematic approaches towards Agile transformation process in the industry. In continuation, a short description is provided for the different Agile adoption frameworks identified in the literature:

- The framework proposed by Conboy and Fitzgerald consists of two set of factors – four method characteristics and three developer practices, which together influence the effectiveness of method tailoring. They proposed a framework based on an extensive literature review and interviews conducted with 20 senior software development researchers [4].

- Knowledge management perspective is an essential part of the framework presented by Chan et al. In their research, they identified intensive communication of the developers and customers, and conceptual framework for accepting Agile methods in organisations was presented [5].
- The framework for adapting Agile development methodologies proposed by Cao et al. [6] was based on the results of multisite case study. They investigated how the structure of Agile methods, projects, and organisations affect the Agile method adoption process.
- A significant literature analysis of Agile adoption strategies was conducted by Rohunen et al. [7]. The obtained results are classified in three categories: Agile methods adoption strategies, Agile adoption stages and managing dependencies between different Agile method adoptions.
- The Agile adoption framework presented by Sidky et al. consists of two modules: an Agile measurement index and four-stage process [8]. It may be used to guide and assist the Agile adoption efforts of organisations.
- An empirically-developed framework for Agile transition and adoption [9] was presented by Gandomani and Nafchi. In their research, besides presenting the framework, they demonstrated factors of Agile adoption process, such as facilitators, challenges and issues, and prerequisites.
- The framework proposed by Barlow et al. provides guiding principle for big organisational systems with recommendations on how process change should be managed to implement Agile practices with success [10]. Three main factors influencing the choice of adequate methodology were identified in the research.
- Another approach is demonstrated in the framework for evaluating the suitability of Agile method fragments before initiating Agile transformation is presented in the research study [11]. Depending on the objectives of the transformation, employees should choose which Agile method practice to implement.
- A complete framework [12] to assist managers in evaluating the required degree of agility in the company, and in Agile method adoption initiative was presented by Qumer and Henderson-Sellers. In addition, analytical framework for measuring the degree of agility (4-DAT) was presented in another research [13].

Besides reviewing the Agile adoption frameworks identified in the literature, it was important to review guidelines for Agile transformation process applied in industry. In the continuation, several comprehensive concepts for Agile adoption, identified to be significant and commonly used in industry are presented:

- *Scaled Agile Framework 4.0 (SAF 4.0)* demonstrates organisational roles tools and techniques at three organisational levels (portfolio, program and team). It is based on nine fundamental principles [14]: take an economic view, apply system thinking, assume variability: preserve options, build incrementally with fast, integrated learning cycles, base milestone on objective evaluation of working systems, visualize and limit work-in-progress reduce batch sizes, and manage queue lengths, apply cadence, unlock intrinsic motivation of knowledge workers and decentralize decision making.

- *Holacracy* as a concept defines a flat organisational structure with rigor needed to run a business effectively. This concept bring structure and discipline to a peer-to-peer workplace [15]. According to the founders, Holacracy is implemented in more than 500 organisations.
- *Reinventing Organizations* book by Frederic Laloux, presents structure, practices and culture of teal organisation. Processes and structure of self-organised is demonstrated and evolution of organisational model from the past till now is summarized [16].
- *Sociokratija (eng. Sociocracy) 3.0* – demonstrates a framework for evolutive agility and elastic organisations [17]. The framework is based on seven principles: consent, empiricism, effectiveness, equivalence, transparency, accountability and continuous improvement.
- *How to Change the World* – is a framework to manage changes, also called management 3.0 [18]. The framework is based on four main principles: dance with the system, mind the system, stimulate the network and change the environment.
- *Nexus* is a framework for integrating larger number of scrum teams (3-9) so it supports scaling of Agile practices in the organisation [19].
- *Scaling Agile at Spotify* demonstrates organisational structure appropriate for scaling Agile practices in organisations [20]. Process of implementing Agile principles in Spotify Company working in the field of streaming music is demonstrated as an example of best practice.

The presented frameworks and structural approaches show various approaches towards Agile adoption process in the organisations. It can be stated that different aspects of Agile transformation process and organisational change management have been presented in research. Moreover, it can be concluded that the identified frameworks vary significantly in their structure and focus and that there is a need to further explore Agile transformation process in organisations. In the following section, we present our approach to the framework we propose for the introduction of Agile methods in organisations.

3 A New Framework for Agile Adoption

Our framework for Agile adoption is presented in Fig. 1. It is composed of three sequential phases of the Agile transformation process with two typical activities within each phase.

Phase 1 consists of two typical activities implemented before the initiation of the Agile adoption process in the organisation. The phase is named *Preparation* and it consists of selection of the transformation strategy and planning of transformation.

Phase 2 is named *Transformation* and, in this phase, the Agile adoption process is implemented at the team level. Two types of activities would be performed during this phase: Agile method implementation, and analysis, improvement and adaptation.

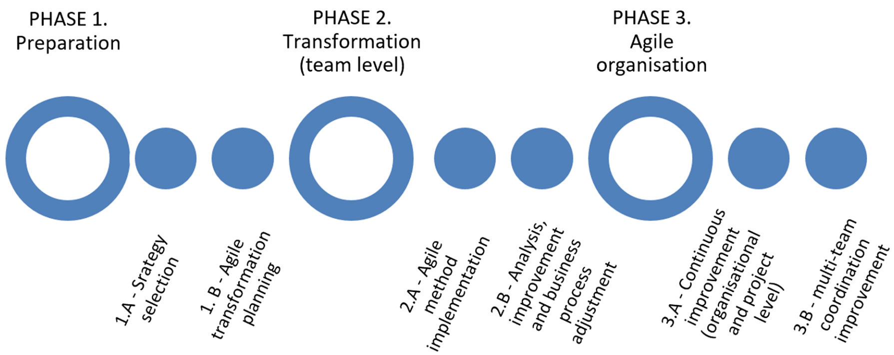

Fig. 1. The proposed Agile adoption framework.

Phase 3 is named *Agile organisation* and, at this point of transformation, Agile practices would be adopted at organisational level - above team levels. Usually, business process and multi-team coordination would be continually improved. Basically, in the second phase, focus of Agile adoption process would be on teams, while in the third phase, higher management structures and organisational processes would be in focus.

In the following subsections, the Agile adoption framework phases and typical activities are described.

3.1 Phase 1: Preparation Phase

The phase 1 of the proposed framework is related to the preparatory activities, adequate to be implemented before starting the introduction of the Agile method in the organisation. In this phase, there are two activities:

- Activity 1.A. Strategy selection.
- Activity 1.B. Agile transformation planning.

3.1.1 Activity 1.A. Strategy Selection

Two generic strategies for the introduction of Agile methods in the enterprise have been identified in the literature [21]:

- Contingency factors identification (1.A.1) and
- Agile methods engineering (1.A.2).

In the Fig. 2, a graphical presentation of the strategy selection for the introduction of Agile methods in the organisation is shown.

The strategy of identifying the contingency factors of the project (1.A.1) would mean that organization would evaluate contingency factors of specific project before starting the transformation. Based on the identified project context (factors presented in Table 1), the desired method for the transformation may be selected (plan-driven, hybrid or Agile). The selected method then becomes the objective of the Agile transformation. Factors that should be evaluated to determine the suitable method for

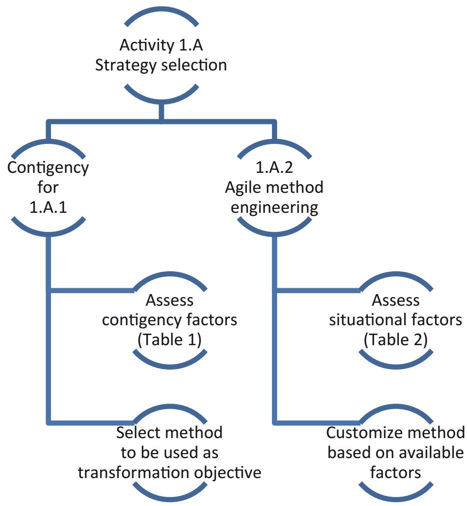

Fig. 2. Decision process for selecting the Agile method adoption strategy.

Table 1. Contingency factors used to determine the most adequate method

Factor	Agile method	Plan-driven method
1. Project team size	Small teams (<50)	Big teams (>50)
2. Project result criticality	Low criticality (financial resources)	Highly critical (impact on humans/life)
3. Project rhythm	Short delivery time frame	Easily reachable delivery date
4. Number of expected request changes per month/Project uncertainty	Big number of changes (>50)	Small number of changes (<50)
5. Team members' skill to adapt and follow a methodology	Majority of members with high expertise	Majority of members with low expertise
6. Culture	Accepting and fostering procedure flexibility	Accepting and fostering procedure stability
7. Team member rotation (stability)	Lower level of changes	Higher level of change
8. Project complexity/Project activity dependencies	High complexity/most of the activities in parallel	Low complexity/most of activities in sequence
9. Project type	Breakthrough and development projects	Derivative and platform projects

concrete project are presented in Table 1. The nine factors are based on four research studies [10, 22–24]. The proposed list of factors may be used for project contingency evaluation (project risk level). Depending on the evaluation of contingency factors, adequate method will be chosen as transformation objective.

Assuming that the company starts from plan-driven (traditional), transformation is implemented only when is intended to establish a hybrid or Agile method in the organisation, while in the case of plan-driven methods there is no need to pursue an Agile transformation.

For instance, if for specific project it can be identified that time frame of delivery is short (factor 3), team member rotation is low (factor 7) and that number of changes is high – more than 50 per month (factor 4), then “Agile method” should be used on that project. Ideally, the organisation is able to evaluate all nine the factors in Table 1, but if that is not possible, the more factors evaluated the better.

Hybrid methods represent a mix of Agile and plan-driven methods, implementing only some Agile techniques and practices in the existing business, depending on the desired outcome. If the outcome of the contingency evaluation is to transform towards Agile, then “Agile method” would be selected and implemented in the organisation.

Agile methods engineering (1.A.2) refers to the adjustment of the methods available to suit the specific needs of the organisation, keeping in mind the situational context of the project, organisation and environment. Based on the identified situational factors, Agile methods (one or more) are being adapted to the situation on the project and its environment.

As a result of our literature review of situational factors that can be used for Agile method engineering, the list of most influencing factors on the Agile adoption process is presented in Table 2. The following literature sources were used in Table 2: [2, 3, 5, 9, 10, 18, 22, 25–31, 33], [35, 36, 37, 38, 40]. First column shows the situational factor name, second column shows literature reference (source where factor was identified). Only factors having three or more literature references are presented in the table, sorted in descending order of number of references and then, alphabetically.

Table 2. Situational factors influencing the Agile adoption process

Situational factor	Literature references
1. Organisational/Corporate culture	[5, 22, 25–30]
2. Team size/scale	[2, 10, 32] [36, 38] [31] [40]
3. Management support	[2, 18] [37] [2, 3]
4. Training	[2, 9, 18] [35]
5. Previous experience	[18, 28] [37, 38]
6. Project budget/cost	[2] [37, 38, 40]
7. Team distribution/co-location	[2] [37, 38, 40]
8. Communication	[18] [37, 38]
9. Contract type	[2] [37, 40]
10. Customer collaboration/involvement	[2] [35, 38]
11. Domain knowledge/expertise	[37, 38, 40]
12. Organisation maturity level	[37, 38] [28]
13. Organisational instability/dynamism/turnover	[25, 32] [36]
14. Previous knowledge/expertise/skill	[36, 37, 38]
15. Project criticality	[2, 32] [38]
16. Project time/duration	[2] [38, 40]

On the one hand, the list of situational factors presented in Table 2 may be useful for the companies that selected the *Agile methods engineering strategy* (1.A.2) to be pursued in the Agile transformation process. In Phase 1 of the transformation it is necessary to identify which situational factors from the list are available and may be assessed. Then, based on the estimated situation, the Agile method is adapted to best fit the situational context of the project and its organisational environment.

On the other hand, if in the company it is decided to follow a *Contingency factor identification strategy* (1.A.1) then, the nine factors summarized in Table 1 may be useful to choose which management method to use on the project(s): plan-driven, hybrid or Agile.

3.1.2 Activity 1.B. Agile Transformation Planning

The first step (1.A) in preparing the transformation is the decision making on the general Agile adoption strategy. The second step, planning the Agile transformation (1.B), provides an operational plan of how to achieve the transformation. If we would consider the Agile adoption process as a process improvement initiative in the organisation, then we could say that the first group of activities to be conducted (1.A) are: defining the project objectives, the high-level work packages and the project constraints. In that case, the second stage (1.B) would be a detailed operational plan for reaching the project objectives, similar to a project plan. The organisation should plan the transformation before starting the *Transformation* itself (Phase 2 of the framework). The Agile transformation plan should be considered as any other project in the organisation and therefore, funds should be allocated, time frame should be defined and objectives should be clear and transparent for all project participants. Only then, the organisation is ready to move on to the next phase.

3.2 Phase 2: Transformation (Team Level)

The implementation of Agile methods in the company (at the level of project teams) starts with the execution of a pre-established plan for managing the organisational change. After having implemented this plan, Agile methods are implemented at the team level. In the *Transformation* phase, two typical activities are implemented:

- Activity 2.A. Agile method implementation.
- Activity 2.B. Analysis, improvement and business process adjustment.

In the *Agile method implementation* activity (2.A), it is necessary to adjust the organisational structure and business processes to the Agile method (chosen as the objective of the transformation). For instance, if the goal of the transformation was to introduce the Scrum method in the organisation, it would mean that change management plan would involve plans such as: transition of organisational roles, introduction and modification of existing processes for project documentation (artefacts), introduction of Agile ceremonies (meetings) and Agile tools and techniques (practice).

Activity 2.B represents *improvement and adaptation of business processes*. Since at this point Agile methods are implemented mainly at team level, techniques developed for teams would be a best fit to improve the transformation process. For instance, in our previous research [32], tools and techniques for process improvement in Agile retrospectives are analysed and grouped depending on the maturity of the project team, facilitating the selection of the most appropriate techniques to improve business processes.

3.3 Phase 3: Agile Organisation

The third phase of the Agile transformation framework is named *Agile organisation*. Second phase is terminated when Agile methods are expected to be successfully implemented at team level. Continuous improvement is mentioned implicitly in the second phase (activity 2.B *Analysis, improvement and business process adjustment*), but in the third phase it is explicitly stated in the first activity 3.A *Continuous improvement (organisational and project level)*. Therefore, in the third phase of the Agile transformation the focus is on process improvement on the organisational level and on the cooperation practices among teams. *Agile organisation* phase is identified in C3 and one of the changes that were implemented is that they created continuous improvement team (department). This department is established on the organisational level to provide trainings, coaching and gathering of best practices (identified externally and internally) with aim of improving multi-team coordination and improving processes at the organisational level. In this last phase, two concrete activities have been identified:

- Activity 3.A. Continuous improvement (organisational and project level).
- Activity 3.B. Multi-team coordination improvement.

4 Validation and Next Steps in the Framework Development

The multi-case study was conducted in three organisations during a two-year period, starting in March 2015 and finishing in December 2016. The study was conducted with an open ended questionnaire based on the questionnaire presented in the following research [33]. In total, 44 interviews were conducted, 21 interviewee from the first organisation, 9 from the second one and 14 from the third organisation. Interviews were conducted on the voluntary base, the employees were informed about the research and volunteers were invited to the interviewing process. Interviews were conducted and transcribed by the researchers. Minimally two researchers participated in the interviews in each company and an open ended questionnaire was used as a guideline for the case studies. Schedule for interviewing was made in line availability of employees, and in average interviews lasted 45 min per participant. First (C1) and third (C3) companies are located in Spain while second company (C2) is located in Serbia. All three organisations are large organisations and they provide IT services in tourism, automotive and e-commerce industries. At the time of interviews, the first and the second organisations were in the first year of Agile transformation process, and the third one implemented Agile methods four years ago.

Interviewees were given the opportunity to evaluate initial version of the model and the suitability to their concrete situation. They clearly identified in which stage their organisation currently was. Employees from C1 and C2 recognized that their company was in the second phase, while majority of participants from C3 stated to be in third phase of the Agile transformation.

In the Phase 1 of the Agile adoption process (*Preparation*) it is necessary to define the plan for introducing the Agile method or adapting the method currently used in the company. It is necessary to make a project plan and define priorities of the activities for implementing the Agile method in the organisation. Validation in the case companies showed that the C2 did not have clearly defined a plan and, in the opinion of the researchers, they did not implement the first phase properly. The implementation strategy was not clear and change management plan was not defined. They started directly with an education program of employees, which was followed with pilot projects of Agile method adoption. Five teams were chosen to participate in the pilot project and to be the first in the IT department to implement Agile method. Lack of planning in the beginning caused less effective *Transformation* in the second phase. In C1 and C3, external assistance (Agile coach) was employed and he assisted the entire transition. Agile coaches were acting as transition managers and, according to the interviewee responses, they were very useful in the first and second phase of Agile transformation.

All three companies have been using some strategy to adapt Agile method to their specific circumstances (1.A.2 *Agile method engineering*), but before seeing Table 2 with the situational factors influencing the Agile adoption process, interviewees stated that they did not really think how should they improve processes in a structured way. Therefore, Table 2 was useful for employees to think about further improvements and method adaption. They were also given opportunity to provide their opinion on the significance of the listed situational factors for Agile transformation process. Results are shown in Table 3. If more than 50% of employees evaluated some factor as significant for Agile transformation process, it is shown in the table as a company factor. It can be concluded that majority of situational factors identified in the literature are evaluated as quite relevant in the opinion of the interviewees.

Table 3. Situational factors influencing the Agile adoption process evaluated by interviewees

Situational factor	Company
1. Organisational/Corporate culture	C1, C2, C3
2. Team size/scale	C1, C2
3. Management support	C1, C2, C3
4. Training	C1, C2, C3
5. Previous experience	C1, C3
6. Project budget/cost	C1, C2
7. Team distribution/co-location	C1, C2, C3
8. Communication	C1, C2, C3
9. Contract type	C2, C3
10. Customer collaboration/involvement	C1, C2, C3
11. Domain knowledge/expertise	C1, C2, C3
12. Organisation maturity level	C1, C2, C3
13. Organisational instability/dynamism/turnover	C1, C2
14. Previous knowledge/expertise/skill	C1, C3
15. Project criticality	C1, C2
16. Project time/duration	C2, C3

As it was mentioned before, case companies at the moment of interviews were in Phase 2 and Phase 3 of the Agile transformation process. Therefore, interviewees gave most of the feedback, and have found most useful phase 1 of the framework. So, we can consider *Preparation* Phase 1 of the framework, as a well detailed and validated phase, ready to be used by other organisations.

During Phase 2 of the Agile transformation process "tensions" rise on the team level and processes are adapted to accommodate in line with organisational transformation. When similar tensions are starting to appear on higher organisational levels it is one of the signs that company is entering Phase 3. Challenges in cooperation of different teams start to arise, and processes should be tailored to best coordinate teams working on the same product. In all three case companies, the Agile transformation was implemented in the IT department. The IT department in C3 is virtually "integrated" with the financial department, as the company deals with e-commerce, and potential fraud and financial transactions are of great importance for the final software product (platform). After successful Agile transformation of the teams working in the IT sector, the gap and tensions between the financial sector and the IT sector have increased and there is a need for a new organisational change. The main difference of department "incompatibility" was in terms of planning cycles. On the one hand, Agile methods support a short-term, frequent and adaptive planning and adjustment of the plan and, on the other hand, the financial department has a different way of execution of work – long term planning and operational services customized for clients. Therefore, what can typically occur as a problem after the successful implementation of Agile methods in teams is how to tailor processes in other departments where Agile is not implemented, and how to reach the Agile organisation.

Phases 2 and 3 of the framework should be further explored and validated in the companies. This initiative is left for future work. More "mature" organisations, in terms of Agile adoption, should be involved in future, or the same validation should be repeated in the same participant organisations after a certain period of time. In this research, the organisational maturity in the beginning and after the transformation was not measured since research was conducted during, or after, the Agile transformation process. Even if we would be able to measure the situation before and after the Agile transformation, the question would be: how to measure the organizational maturity in the Agile context? In general, available organizational maturity models should be adapted and customized to fit Agile context.

5 Conclusion and Future Work

This paper demonstrates the first steps towards the creation of a generic Agile adoption framework. The proposed framework is the result of (1) a detailed analysis of academic literature and industrial reports and (2) a multi-case study conducted in three large enterprises. Our Agile adoption framework is composed of three main sequential phases for the organisational change in the process of Agile method adoption. Phase 1 (*Preparation*) of the framework can be considered as well developed and validated in the case companies. Since companies were in Phases 2 (*Transformation*) and 3

(*Agile organisation*), their employees were able to discuss and contribute to the validation and further improvement of Phase 1.

Main contribution lies in the developed framework for Agile adoption in organisations. Tables 1 and 2 show contingency and situational factors of the project and its environment that should be taken into account when initiating the Agile adoption process. The list of these factors are an integral part of the framework, but they can also be used separately as an isolated toolbox to evaluate contingency or situational context of a specific project.

Phases 2 and 3 are identified but not described in detail in this article. These two phases are planned to be further investigated in more "mature" organisations in terms of Agile method adoption. Activity description in these two phases will be refined by using the feedback received. Also, the assessment of the situational factors affecting the Agile transformation process would be significant to explore in future research initiatives. Moreover, we expect further validation of the complete framework from both industry and academia in order to develop a new detailed version of the complete framework.

Acknowledgements. This work has been supported by the Spanish Ministry of Science and Technology with ERDF funds under grants TIN2016-76956-C3-3-R and TIN2013-46928-C3-2-R and by the Erasmus Mundus Euroweb+ project.

References

1. Laanti, M., Salo, O., Abrahamsson, P.: Agile methods rapidly replacing traditional methods at Nokia: a survey of opinions on agile transformation. Inf. Softw. Technol. **53**(3), 276–290 (2011)
2. Kalus, G., Kuhrmann, M.: Criteria for software process tailoring: a systematic review. In: Proceedings of the 2013 International Conference on Software and System Process - ICSSP 2013, p. 171 (2013)
3. Pikkarainen, M., Salo, O., Kuusela, R., Abrahamsson, P.: Strengths and barriers behind the successful agile deployment-insights from the three software intensive companies in Finland. Empir. Softw. Eng. **17**(6), 675–702 (2012)
4. Conboy, K., Fitzgerald, B.: Method and developer characteristics for effective agile method tailoring: a study of XP expert opinion. ACM Trans. Softw. Eng. Methodol. **20**(1), 1–30 (2010)
5. Chan, F.K.Y., Thong, J.Y.L.: Acceptance of agile methodologies: a critical review and conceptual framework. Decis. Support Syst. **46**(4), 803–814 (2009)
6. Cao, L., Mohan, K., Xu, P., Ramesh, B.: A framework for adapting agile development methodologies. Eur. J. Inf. Syst. **18**(4), 332–343 (2009)
7. Rohunen, A., Rodriguez, P., Kuvaja, P., Krzanik, L., Markkula, J.: Approaches to agile adoption in large settings: a comparison of the results from a literature analysis and an industrial inventory. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 77–91. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13792-1_8
8. Sidky, A., Arthur, J., Bohner, S.: A disciplined approach to adopting agile practices: the agile adoption framework. Innov. Syst. Softw. Eng. **3**(3), 203–216 (2007)
9. Javdani Gandomani, T., Ziaei Nafchi, M.: An empirically-developed framework for agile transition and adoption: a grounded theory approach. J. Syst. Softw. **107**, 204–219 (2015)

10. Barlow, J.B., et al.: Overview and guidance on agile development in large organizations, vol. 29, pp. 25–44, July 2011
11. Chiniforooshan Esfahani, H., Yu, E., Cabot, J.: Situational evaluation of method fragments: an evidence-based goal-oriented approach. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 424–438. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13094-6_33
12. Qumer, A., Henderson-Sellers, B.: A framework to support the evaluation, adoption and improvement of agile methods in practice. J. Syst. Softw. **81**(11), 1899–1919 (2008)
13. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Inf. Softw. Technol. **50**(4), 280–295 (2008)
14. Leffingwell, D., Yakyma, A., Knaster, R., Jemilo, D., Oren, I.: SAFe Reference Guide 4.0 - Scaled Agile Framework for Lean Software and System Engineering. Scaled Agile, Inc., Indiana, USA (2016)
15. HolacracyOne LLC, Holacracy - Discover a Better Way of Working, pp. 1–10 (2015)
16. Laloux, F.: Reinventing organizations: a guide to creating organizations inspired by the next stage of human consciousness, p. 379 (2014)
17. Priest, J., Bockelbrink, B.: Sociocracy 3.0, p. 147. Sociocracy30.org (2017)
18. Appelo, J.: How to change the World - Change Management 3.0 (2012)
19. Schwaber, K.: Nexus guide. In: 2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games), p. 11. Scrum.org (2015)
20. Henrik, K., Anders, I.: Scaling Agile @ Spotify - with Tribes, Squads, Chapters & Guilds, p. 14 (2012)
21. Fitzgerald, B., Hartnett, G., Conboy, K.: Customising agile methods to software practices at Intel Shannon. Eur. J. Inf. Syst. **15**(2), 197–210 (2006)
22. Boehm, B., Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley, Reading (2003)
23. Shenhar, A.J., Dvir, D., Lechler, T., Poli, M.: One size does not fit all - true for projects, true for frameworks. In: PMI Res. Conference, September 2016, pp. 99–106 (2002)
24. Wheelwright, S.C., Clark, K.B.: Creating project plans to focus product development. Harv. Bus. Rev. **2**, 1–15 (1992)
25. Sheffield, J., Lemétayer, J.: Factors associated with the software development agility of successful projects. Int. J. Proj. Manag. **31**(3), 459–472 (2013)
26. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: a systematic review. Inf. Softw. Technol. **50**(9–10), 833–859 (2008)
27. Misra, S.C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. J. Syst. Softw. **82**(11), 1869–1890 (2009)
28. Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: towards a comprehensive reference framework. Inf. Softw. Technol. **54**(5), 433–447 (2012)
29. Campanelli, A.S., Parreiras, F.S.: Agile methods tailoring - a systematic literature review. J. Syst. Softw. **110**, 85–100 (2015)
30. Yang, C., Liang, P., Avgeriou, P.: A systematic mapping study on the combination of software architecture and agile development. J. Syst. Softw. **111**, 157–184 (2016)
31. Brhel, M., Meth, H., Maedche, A., Werder, K.: Exploring principles of user-centered agile software development: a literature review. Inf. Softw. Technol. **61**, 163–181 (2015)
32. Jovanovic, M., Mesquida, A.-L., Mas, A.: Process improvement with retrospectives gaming in agile software development. Commun. Comput. Inf. Sci. **543**, 287–294 (2015)
33. Paasivaara, M., Lassenius, C.: Communities of practice in a large distributed agile software development organization - case Ericsson. Inf. Softw. Technol. **56**(12), 1556–1577 (2014)

SPI in Small Settings

Comparing SPI Survival Studies in Small Settings

Xabier Larrucea(✉) and Izaskun Santamaria

Tecnalia, Parque Tecnológico de Bizkaia 700, 48160 Derio, Spain
{Xabier.Larrucea,Izaskun.Santamaria}@tecnalia.com

Abstract. Small organisations have been applying several quality approaches such as CMMI-DEV or ISO/IEC 15504-5 with quite diverse results. In order to build an experience factory we are gathering our experiences in a database containing the assessment results of more than 90 initiatives. This paper provides an empirical comparison of survival analysis for different improvement initiatives in the context of very small entities. We compare the Cox Proportional Hazard Regression models of our 90 initiatives, and we discriminated them by the reference model used: ISO/IEC29110, CMMI-DEV or ITMark.

Keywords: ISO/IEC2 9110 · Survival analysis · Product life cycle

1 Introduction

Small organisations have been applying several quality approaches such as CMMI-DEV [1] or ISO/IEC 15504-5 (SPICE) [2] with quite diverse results. Some research works are highlighting the fact that all these traditional reference models are not appropriated for these settings [3]. Several industrial [4, 5] and research [6] works have been carried out in the realm of very small entities (VSE) as defined by the ISO/IEC 29110 [7]. Other experiences have been reported in this sense such as [8] which identifies financial, skills, culture and reference models as the most common barriers for VSEs. In fact VSEs are always hesitating to embark or not onto these improvement initiatives related to one specific reference model because they cannot foresee the expected results. In fact, reference models have several aspects in common because most of them derived from the same roots. According to [9] there is a wide variety of reasons related to why projects fail, such as unclear objectives, unrealistic or unarticulated project goals or inaccurate estimates of needed resources. Therefore reference models used to deal with these elements among others. In addition, one of the most relevant aspects for these small organisations is the assessment and/or the expected results when they are applying a reference model. There are several discussions around reference models such as SPICE [10], or for improving processes and products [11] or even approaches for dealing with multiple reference models at the same time [12]. However the financial aspect is not usually measured or reported. As identified by [8] the financial aspect and the associated reference models are some of the barriers for VSEs as stated previously. This financial aspect includes among other factors the time invested and required for implementing one of these initiatives in a

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 45–54, 2017.
DOI: 10.1007/978-3-319-67383-7_4

VSE context. In its turn, this invested time depends on several factors such as organisations' size, resources involved for launching and carrying out these initiatives, their duration and several other factors. In this sense, an estimated duration for implementing these initiatives provides an overview of the time required by VSEs. This "survival" time is one of the elements that we have measured from our current experience factory [8].

This paper aims to provide an empirical comparison of survival analysis of improvement initiatives in small contexts such as the ISO/IEC 29110. In addition we provide the survival analysis of 90 initiatives referenced by [8, 13]. This statistical analysis helps VSEs to identify whether the improvement initiative is going to fail or not.

This paper is organised as follows. First a brief background introduction to ISO/IEC 29110 and survival methods are provided. Second the research method, data collection and data analysis method are described. Finally the main results are discussed in order to conclude this paper.

2 Background

2.1 ISO/IEC29110

The ISO/IEC29110 [7] is defined for helping very small entities (VSE) to improve their quality through the use of profiles [14]. This standard allows VSEs to adapt smoothly the activities defined by these profiles into their organisations' needs. Some research works are aligned with these principles such as [15] which defines a framework called Rapid-Q predefining a set of processes that can be customized to the organization's needs. Some authors such as [16] have analysed the software process improvement (SPI) efforts devoted by VSEs, and our paper contributes directly in this sense. In fact, VSEs require a clear and defined route [17] for launching and investing resources, and they need to estimate the required time to achieve a set of goals. Several contributions have been reported at different levels such as the assessments carried out [18], the project management activities [19], or the activities related to software engineering [15]. It is also relevant to mention that this standard has been used in the educational environment [20, 21]. As identified by [22] there are six common problems observed in this kind of environments: poor project planning, poor measurements, poor cost estimating, poor change control, poor milestone tracking, and poor quality control. All these aspects can be managed under the ISO/IEC29110.

2.2 Survival Methods

In this paper we define "survival time" as the time required until an organization has achieved a set of activities prescribed by a reference model. In our context we are considering to meet requirements defined by ISO/IEC 29110. This achievement can be measured by a traditional assessment or by a more light weight approaches. Survival methods are defined in the realm of statistical methods which have been applied to several domains such as health sector [23], or in economics [24]. The survival data refers to the observations related to the time required to a certain event [25].

This approach is similar to [26] where the survival time is a positive event and it is the duration time until an event has occurred. Traditionally the survival methods cover parametric (e.g. lognormal, Weibull, etc), non-parametric (or semi-parametric) approaches (e.g. Kaplan-Meier), semi parametric (e.g. Cox Proportional Hazard Regression model) among others. In our context we use the semi-parametric approach called Cox Proportional Hazard Regression (CPHR) model because the distribution is unknown, and the time to the event is not fully observed. In fact CPHR is a blend model mixing time dependent variables and categorical data. Process improvement assessments are usually carried out at a certain intervals. In addition some initiatives are abandoned or failed during this interval of observation. Therefore we have censored the data falling outside the limits of our study. The events occurred before the starting times are called left truncated data. And the events occurred after the ending times are called right censored data.

3 Survival Analysis for Small Settings

There is a wide set of survival methods for analyzing "time to event" approaches. This section provides an overview of the non-parametric models and a semi-parametric model such as the Cox Proportional Hazards Regression (CPHR) model [27]. As stated before we are going to use CPHR because software process improvement assessments rely on time dependent variables and categorical data. The first step is to introduce the non-parametric models. Second we need to interpret and adapt the Cox Proportional Hazard Regression model to our study. Third we analyse the scenario, and we need to specify which software process improvement initiatives are taken into account or not.

3.1 Non Parametric Models

Kaplan-Meier [28] and Nelson-Aalen estimators are some of the most well-known non-parametric models. Kaplan-Meier defined the following model:

$$\hat{S}(t) = \frac{1}{n}\sum_{i=1}^{n} I\{t_i > t\} \tag{1}$$

Where I is an indicator function evaluating whether $t_i > t$ is True or not.

Another survival function approach is the Nelson-Aalen which is used when we consider estimating the cumulative hazard.

$$\hat{\Lambda}(t_{(i)}) = \sum_{j=1}^{i} \frac{d_j}{n_j} \tag{2}$$

This function accumulates (sum) the hazard from time = 1 to time = i.

Both functions are used in our study for comparing the duration of software process improvement initiatives in VSEs.

3.2 Semi-parametric Model: Cox Proportional Hazard Regression Model

We are going to use the semi–parametric Cox proportional hazards regression model [27] which is an extension of the Kaplan-Meier estimator because we can use numerical variables, and because we do not know the distribution behind the software process improvement initiatives:

$$h_i(t) = h_0(t) * \exp(\beta * \mathrm{X(t)}) \tag{3}$$

where $h_i(t)$ is a hazard rate for a subject "i", $h_0(t)$ depends on time (not on the covariates) with unspecified baseline hazard function that describes the instantaneous risk of experiencing an event at some time, t, when the values of all covariates are zero. $\exp(\beta * \mathrm{X(t)})$ depends on the covariates (not the time). X(t) is a vector of possibly time-varying covariates that are collected at each event occurrence that may or may not have predictive power over the time to the event. This vector is composed by several parameters which are common in several reference models such as the ISO/IEC 29110 basic profile elements. β is a vector of regression coefficients (i.e., one coefficient for each covariate)

In our cases there is an interest to compare two different initiatives. The main difference between 2 subjects under study (two software process improvement initiatives) only depends on their covariate values as described in formulae 4.

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t) * \exp\left(\beta * X_i(\mathrm{t})\right)}{h_0(t) * \exp\left(\beta * X_j(\mathrm{t})\right)} = \exp\left(\beta * \left(X_i(\mathrm{t}) - X_j(\mathrm{t})\right)\right) \tag{4}$$

Our aim is not to provide a depth explanation of the mathematics behind the Cox's model. In this paper we explain the Cox's model formulae because we want to compare the behavior of different process improvement initiatives. For representing the results use the R studio [29] and the Cox's model implementation in the R [30] package survival [31].

4 Empirical Study of Survival Analysis in Small Settings

4.1 Research Method

Recent research works such as [32] where authors outline a research agenda, or [33] where authors provide an approach for predicting delays of issues with due dates, are suggesting that there is an evident need for setting a grounded theory [34] in this sense. As stated before we have analysed 90 improvements initiatives stemming from our experience factory [35] which has been published in Tecnalia's website (https://tinyurl.com/larnc8q). In fact the aforementioned webpage contains further experiences but they are not taken into account because they are not small companies and/or we do not have enough information regarding the assessments and the time used for each initiative. Therefore we analyzed a wide set of process improvement projects related to VSEs. There are some companies which are small- medium entities but they are not

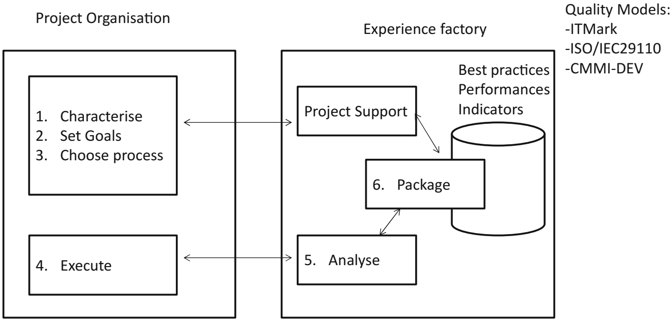

Fig. 1. Empirical research method [35] of Tecnalia VSEs' experiences.

VSEs, and we focused on those organisations achieving a set of reference models such as CMMI-DEV or ITMark. During this research method (Fig. 1) we packaged these experiences in a database containing which reference model was used, the time required/invested, and best practices.

4.2 Data Collection

We collected the data from this experience factory, and we identified different types of events (Fig. 1):

(1) Starting event – this is the first observation that a software process improvement initiatives has started.
(2) Succeeded/Failed event- this event is positive or negative depending on whether the improvement initiative succeeded or not.
(3) Censoring event – an event that is falling outside the interval of study. This event can be left-truncated data or right censored data. Left truncated data is not considered on this study because we do not know whether the organisations started or not an improvement initiative before the time t1 (Fig. 2). T1 is basically the first

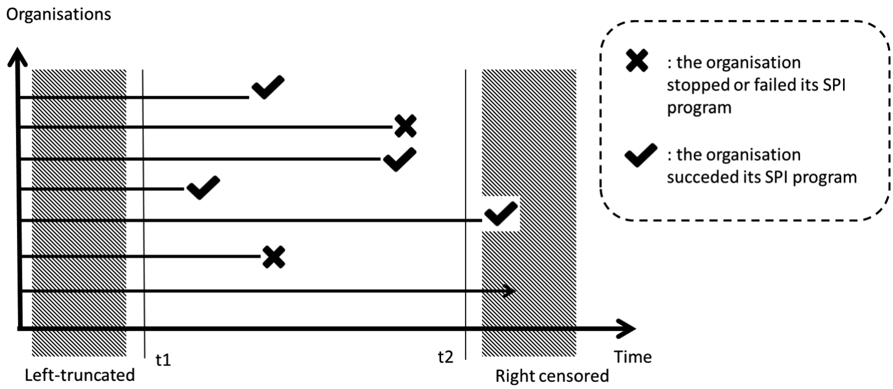

Fig. 2. Types of events: failure/success possibilities including right censored data

Table 1. An excerpt of the experience database

Duration	Event	Censor	ITMark	CMMI	VSE	Pm1	Pm2	Pm3	Pm4	Pm5	Pm6	Pm7	si1	si2	si3	si4	si5	si6	si7
13	1	0	1	1	1	40.32	45.13	54.52	62.67	49.82	41.81	24.05	55.47	67.69	14.96	17.38	37.24	50.89	40.24
19	1	0	1	0	1	56.83	45.68	64.38	50.96	48.42	46.6	25.28	59.39	65.34	13.83	7.39	43.91	48.01	30.99
17	1	0	1	1	1	49.92	33	61.9	58.77	48.91	50.24	32.7	56.32	62.82	16.74	10.59	32.3	55.74	34.04
21	1	0	1	1	1	57.42	52.72	57.66	45.76	50.33	42.29	23.49	52.56	65.72	9.76	12.3	33.67	42.17	30.02
17	1	0	1	0	1	57.7	54.94	46.42	56.6	49.87	41.18	29.22	53.24	53.98	11.15	13.66	41.47	53.28	38.78
19	1	0	1	0	1	51.59	34.9	62.9	59.7	46.71	53.81	29.57	65.91	48.43	6.98	17.06	40.05	52.83	33.81
17	1	0	1	1	1	47.28	45.77	60	52.81	48.9	51.21	28.47	57.86	61.47	7.76	5.75	42.14	42.39	38.64
22	1	0	1	0	1	64.28	47.88	46.7	58.13	49.34	48.91	29.7	56.81	57.6	9.94	9.05	40.04	51.63	37.2
19	1	0	1	1	1	57.98	53.97	66.88	44.41	48.22	50.22	27.61	60.87	72.54	17.52	13.29	37.17	43.95	32.9
22	1	0	1	1	1	52.45	72.74	56.29	47.86	51.37	56.05	31.59	66.31	33.95	14.14	13.09	35.54	49.21	34.62
15	1	0	1	0	1	64.71	40.21	80.64	47.69	46.42	45.32	32.35	59.84	49.54	9.4	14.24	38.52	52.03	35.84
19	1	0	1	1	1	46.2	45.09	26.19	55.85	47.12	47.46	23.95	53.77	54.33	14.55	10.91	41.16	60.36	32.8
24	1	0	1	1	1	65.44	63.14	33.77	59.75	50.43	56.33	26.23	63.56	60.75	8.23	13.72	47	54.48	40.59
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

observation of our study. Right censored data is the data falling behind t2, and it is not relevant whether this organisation obtained or failed on their SPI initiatives.

All these experiences are gathered in a set of excel sheets, and Table 1 represents an excerpt of the data we are managing. For example, we include the duration required to the event. In fact, the event is Boolean (1 or 0) for representing whether this organization achieved its goals. In addition we are gathering other aspects such as the reference model used (CMMI-DEV or ITMark). Pm1 to pm7 and si1 to si7 are activities defined by the ISO/IEC29110, but they are used in all reference models.

4.3 Survival Results

We use the R studio to represent the results of the CPHR model. Figure 3 provides the survival probability for VSEs included in our experience factory. This figure is the final results of applying CPHR to all VSEs. Those initiatives which do not require more than 13 months are going to comply with reference models requirements. It is worth mentioning that in this case we do not discriminate by the reference model used. This set of initiatives is related because we applied ISO/IEC29110 for other studies.

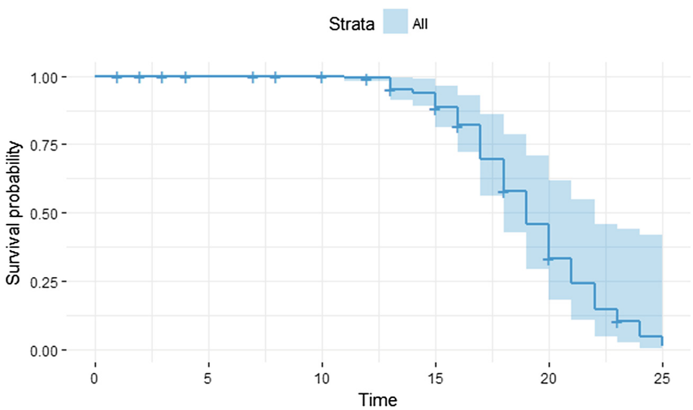

Fig. 3. CPHR for VSEs

Figure 4 compares graphically two subsets. The green line is the CPHR model for VSEs applying CMMI-DEV. The red line is the CPHR model for VSEs applying ITMark [36]. The overall behavior is similar between them to some extent. In fact both subsets their survival probability (Y axis) decrease over the months (X axis). However ITMark is a lightweight reference model and VSEs realize earlier that their initiatives are going to failed or succeed. With CMMI-DEV VSEs require more time to achieve the goals defined by the reference model. In fact there are 4 months of differences between these two results.

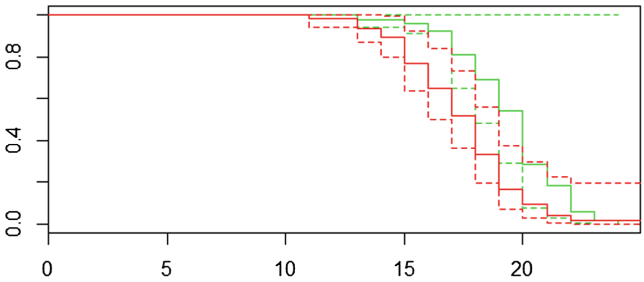

Fig. 4. Comparing CPHR results for ITMark (redline) and CMMI-DEV (green line) (Color figure online)

5 Conclusions and Future Works

This paper is a step forward to build a grounded theory on the survival analysis of software process improvement initiatives. We are building an experience factory containing the assessments of more than 90 initiatives. The gathering process is not cost effective and it requires too much effort identifying the duration and the evidences behind each assessment. Once we have collected all this information we processed them and we used to semi-parametric model called CPHR model for the survival analysis study. This model has been applied to measure negative events such as death. But our approach is focused on positive events such as the achievement of reference models requirements. Our experience factory database contains all these events for each assessment. Most of these assessments are performed in the VSEs context, and we are not just focused on one specific reference model. In fact, we are using different reference models depending on the customer or on the VSEs requirements. Therefore we record and track these elements in our database.

Our survival analysis comparison has some interesting outcomes:

- VSE's initiatives taking more than 13 months are decreasing their survival probability. This means that there are more likely to fail.
- The use of lightweight approaches such as ITMark requires less time to check whether they are going to fail or not on applying an improvement initiative.
- When an initiative is taken too much time their survival probability decreases and their variability increases.

As a future work we will be focused on analysing the roots of this situation. From a subjective perspective we can figure out some reasons and causes of such situation, but this is not real motivation of our study. In fact we are building an experience factory, and we need to manage real data for predicting and planning improvement initiatives.

Acknowledgments. This paper has been partially funded by the AQUAS project with number 737475.

References

1. Software Engineering Institute: CMMI® for Development, Version 1.3 (2010). https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf
2. International Standard Organisation: ISO/IEC 15504-5:2012 - Information technology – Process assessment – Part 5: An exemplar software life cycle process assessment model (2012). https://www.iso.org/standard/60555.html
3. O'Connor, R.V., Coleman, G.: Ignoring "Best Practice": why Irish software SMEs are rejecting CMMI and ISO 9000. Australas. J. Inf. Syst. **16**(1), 7–30 (2009)
4. Laporte, C.Y., O'Connor, R.V.: A systems process lifecycle standard for very small entities: development and pilot trials. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 13–24. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43896-1_2
5. Polgar, P.B., Kazinci, F.: Report on an assessment experience based on ISO/IEC 29110. J. Softw.-Evol. Process. **26**, 313–320 (2014)
6. Tripathi, N., Annanperä, E., Oivo, M., Liukkunen, K.: Exploring processes in small software companies: a systematic review. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 150–165. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_12
7. ISO/IEC: ISO/IEC TR 29110-1. Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — 2011 (2011)
8. Larrucea, X., O'Connor, R.V., Colomo-Palacios, R., Laporte, C.Y.: Software process improvement in very small organizations. IEEE Softw. **33**, 85–89 (2016)
9. Charette, R.N.: Why software fails (software failure). IEEE Spectr. **42**, 42–49 (2005)
10. Boronowsky, M., Mitasiunas, A., Ragaisis, J., Woronowicz, T.: An approach to development of an application dependent SPICE conformant process capability model. In: Woronowicz, T., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 61–72. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38833-0_6
11. Ebert, C., Dumke, R.: Improving processes and products. In: Ebert, C., Dumke, R. (eds.) Software Measurement, pp. 329–434. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71649-5_11
12. Peldzius, S., Ragaisis, S.: Usage of multiple process assessment models. In: Woronowicz, T., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 223–234. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38833-0_20
13. Larrucea, X., Santamaria, I.: Towards a survival analysis of very small organisations. In: Presented at the EuroSPI 2017, Ostrava (2017). (to be published)
14. ISO/IEC: ISO/IEC 29110-4-1. Software engineering — Lifecycle profiles for Very Small Entities (VSEs) (2011)
15. Ribaud, V., Saliou, P., O'Connor, R.V., Laporte, C.Y.: Software engineering support activities for very small entities. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 165–176. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15666-3_15
16. Pino, F.J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Softw. Qual. J. **16**, 237–261 (2008)
17. Sánchez-Gordón, M.-L., Colomo-Palacios, R., de Amescua Seco, A., O'Connor, R.V.: The route to software process improvement in small- and medium-sized enterprises. In: Kuhrmann, M., Münch, J., Richardson, I., Rausch, A., Zhang, H. (eds.) Managing Software Process Evolution, pp. 109–136. Springer, Cham (2016)

18. Varkoi, T.: Process assessment in very small entities. In: 2010 Seventh International Conference on the Quality of Information and Communications Technology Process, pp. 436–440 (2010)
19. Sánchez-Gordón, M.-L., O'Connor, R.V., Colomo-Palacios, R., Sanchez-Gordon, S.: A learning tool for the ISO/IEC 29110 standard: understanding the project management of basic profile. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) Software Process Improvement and Capability Determination, pp. 270–283. Springer, Cham (2016)
20. Sanchez-Gordón, M.-L., O'Connor, R.V., Colomo-Palacios, R., Herranz, E.: Bridging the gap between SPI and SMEs in educational settings: a learning tool supporting ISO/IEC 29110. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) Systems, Software and Services Process Improvement, pp. 3–14. Springer, Cham (2016)
21. Moreno-Campos, E., Sanchez-Gordón, M.-L., Colomo-Palacios, R., de Amescua Seco, A.: Towards measuring the impact of the ISO/IEC 29110 standard: a systematic review. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 1–12. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43896-1_1
22. O'Connor, R.V., Laporte, C.Y.: Software project management in very small entities with ISO/IEC 29110. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 330–341. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31199-4_29
23. Althouse, A.D.: Basic principles of survival analysis with composite endpoints: why you must use the "first" event, not the "worst" event. Int. J. Cardiol. **201**, 319–320 (2015)
24. Fenech, J.P., Yap, Y.K., Shafik, S.: Modelling the recovery outcomes for defaulted loans: a survival analysis approach. Econ. Lett. **145**, 79–82 (2016)
25. Datta, S., del Carmen Pardo, M., Scheike, T., Yuen, K.C.: Special issue on advances in survival analysis. Comput. Stat. Data Anal. **93**, 255–256 (2016)
26. Sentas, P., Angelis, L., Stamelos, I.: A statistical framework for analyzing the duration of software projects. Empir. Softw. Eng. **13**, 147–184 (2008)
27. Cox, D.R.: Regression models and life-tables (1972). http://www.jstor.org/stable/2985181
28. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. J. Am. Stat. Assoc. **53**, 457–481 (1958)
29. R Studio. https://www.rstudio.com/
30. The R project of statistical computing. http://www.r-project.org
31. Therneau, T.M.: Survival Analysis. https://cran.r-project.org/web/packages/survival/survival.pdf
32. Dingsoyr, T., Moe, N.B., Fægri, T.E., Seim, E.A.: Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. Empir. Softw. Eng. **22**, 1–31 (2017)
33. Choetkiertikul, M., Dam, H.K., Tran, T., Ghose, A.: Predicting the delay of issues with due dates in software projects. Empir. Softw. Eng. **22**, 1223–1263 (2017)
34. Carver, J.: The use of grounded theory in empirical software engineering. In: Basili, V.R., Rombach, D., Schneider, K., Kitchenham, B., Pfahl, D., Selby, R.W. (eds.) Empirical Software Engineering Issues. Critical Assessment and Future Directions. LNCS, vol. 4336, p. 42. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71301-2_15
35. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012)
36. Larrucea, X., Santamaría, I., Colomo-Palacios, R.: Assessing ISO/IEC29110 by means of ITMark: results from an experience factory. J. Softw. Evol. Process. **28**, 969–980 (2016)

Assessment Model for HCI Practice Maturity in Small and Medium Sized Software Development Companies

Abiodun Ogunyemi[1](✉), David Lamas[1], Jan Stage[2], and Marta Lárusdóttir[3]

[1] Tallinn University, Tallinn, Estonia
{abnogn, david.lamas}@tlu.ee
[2] Aalborg University, Aalborg, Denmark
jans@cs.aau.dk
[3] Reykjavik University, Reykjavik, Iceland
marta@ru.is

Abstract. Usability and user experience aspects need to be better integrated during software process improvements in the software industry. This study develops and evaluates a comprehensive continuous representation model for conducting peer and self-assessment of HCI practice in software development companies. In this paper, we report on the evolution of an HCI practice assessment model that can be used in software development projects and processes. The specific focus of the model is on the human-centered design practice in four categories: Human-centredness; Process and Infrastructure; Design and Outcomes; and Usability Impact. In order to know whether the model would be beneficial to companies, we conducted two case studies in Estonian software companies. The outcomes indicate that the model can help to increase HCI awareness, self-learning and sharing of a common vision among software practitioners and propel companies towards goal setting for continuous improvement of their HCI practice.

Keywords: Self-assessment · Maturity tool · HCI · Software development · Human-centered design · Usability · Software process improvement · SPICE

1 Introduction

Software process improvement and capability determination are major considerations for software development. Software companies have to focus on these important considerations if they aspire to produce quality products [1]. However, the production of quality products is incomplete without focusing on usability of the products [2]. Even though adequate attention is being paid to products quality, considerably lesser attention has been devoted towards understanding usability aspects in software process improvement [3]. A recent study by García-Mireles et al. [3] revealed that out of 74 papers which they reviewed, 69% were on security aspect of software process improvement. García-Mireles et al. [3] remarked that security, usability, and reliability aspects of quality assessments required certain practices to be deployed in software

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 55–69, 2017.
DOI: 10.1007/978-3-319-67383-7_5

processes and these practices as the authors suggest can be achieved through other disciplines.

Human-Computer Interaction (HCI) is a science of design whose key concepts, that is, usability and user-centered design, tie with ISO standards for development of interactive systems [4]. As a result of this tie, the HCI field has produced set of techniques, and methods, with resounding results that development companies could use [4]. Usability assessments help companies to determine the effectiveness, and efficiency of, as well as customer's satisfaction with their products [5]. The literature on usability evaluation show that despite much efforts put into coding and quality assurance testing, there are still major usability problems in the resulting software because usability aspects were neglected during development [2]. It is said that a product might be deemed a failure if the user is not satisfied because he/she cannot use such a product effectively and efficiently [6]. For example, the current literature indicates that user requirements, user involvement, and user testing are not fully prioritized in development projects (see e.g. [4]). Similarly, many companies either still ignore usability and user experience (UX) design or struggle with it [7]. Therefore, usability assessment in software process improvement is important.

A previous study shows that a lack of communication significantly impacts on the success of software teams [4]. Lack of communication was the most pervasive problem facing a large team and linked to a lack of shared visions of software development [8]. In the same study, Kuusinen, [8] reported that the developers and the UX team rarely discussed common visions, which made the design work by the UX team sometimes unsuitable or not implementable. Communication issues do not only impact on decision-making in large teams or companies [9], nor do they relate only to relationships between developers and designers. Indeed, there is considerable evidence that design ideas and key HCI concepts (e.g. usability) [10], should be communicated amongst other stakeholders, such as developers, users and project managers, (e.g. [11]).

In this paper, we propose a new model based on the concepts of peer and self-assessments, with the main objective being to help companies increase their awareness and knowledge of HCI, and improve team communication by sharing visions and common goals. The knowledge of HCI would help development teams to have the capability to include human dimensions to their development processes. Precisely, HCI considerations in development projects help companies to achieve production of usable and user-friendly products. Companies could remove unnecessary costs associated with training and bug fixing by including HCI dimensions, that is, usability and user experience dimensions in their development processes.

This paper is an outcome of an ethnographic research conducted in two countries; one developed and one developing. We have used the theory of Diffusion of Innovations (DoI) [12] to conduct exploratory investigations in the software development companies. The DoI describes a process through which an innovation spread overtime and explains what to do in order to promote an innovation in a social system. The outcomes of the ethnographic research show that many small and medium sized companies; especially those that develop business support software often experience products failure because they fail to address HCI related practices in their development processes. Our results suggest that there is a need for HCI practice assessment in development companies and not only software process improvement and capability assessments.

Many small and medium software development companies cannot afford third-party assessment tools such as the CMMI and they need to be supported by low-cost, easy to use, and affordable models. Our objective for the HCI practice model is to develop an assessment model for software process improvement and capability evaluation that is scalable, affordable and flexible for small and medium enterprises (SMEs) to support their HCI practice maturation.

The paper is organized as follows: in the next section, we summarize the background of the paper, followed by a presentation of the model. After that, we present the two case studies and results from the software companies where the model has been introduced. Next, we discuss our results and we finally conclude by putting forth limitations of our study and shed light on directions for further research.

2 Background and Related Work

One approach to help software companies to mature their HCI practice is by conducting HCI practice assessments. The usability maturity model (UMM-HCS) [13] was developed based on an European usability support project. Human Factor and HCI consultants, Process Improvement, Business Process engineers and processes assessors and developers of maturity models could use the tool. The UMM-HCS is built based on the principles described in the ISO/IEC TR 15504-2 [14] and 15504-5 [15], that is, reference model for processes and process capability, and assessment model and indicator guidance respectively. However, these underlying models in the UMM-HCS have since been updated. The goal of the UMM-HCS is to help organizations improve their attitude for human-centered system development and project support [13]. The UMM-HCS has 5 major maturity levels and organizations can see how they progressed through these levels when assessed by a consultant. The UMM-HCS scale "measures progress towards the particular goal of human-centredness, rather than the usual overall quality goal used by software process improvement models" [13]. The UMM-HCS has been used to investigate the state of HCD maturity in IT/software development companies (see e.g. [4, 16, 17]. The work of Earthy [13] can be referred to for details regarding the UMM-HCS model. Another maturity model, KESSU, was developed to address the specific challenge of interaction design in IT companies, and for planning usability activities in new projects [18]. The model was built in line with the need for development projects to consider user-centredness in systems development. The underlying principles in KESSU are based on the ISO 13407 [19] and ISO/TR 18529 [20]. The tool is a process model that aims to bridge the gap between usability design and product design; guided by the notion, that: "the usability of a product is not built into the product by usability engineers but by those who design the product - software and other designers" [18].

The Nielsen's Corporate UX Maturity Model[1] describes how an organization progresses from being initially hostile to usability to heavily relying on user research.

[1] See www.nngroup.com/articles/usability-maturity-stages-1-4 and www.nngroup.com/articles/usability-maturity-stages-5-8.

The model has 8 basic stages, which describe how an organization evolves with UX. The major concern is that the model is descriptive. The first 4 stages of the Corporate UX Maturity Model appear to describe the 'what' and 'why' regarding an organization UX maturity, while the last 4 stages tell an organization how to deal with its state of maturity. The model, however, could be challenging to organizations that lack a basic understanding of UX [21]. Recently, the STRATUS model was developed as a response to the need for strategic usability assessment in companies [5] and particular barriers to usability practice (e.g. methodology, knowledge, and organization attitudes). STRATUS is just a recent tool and its goal is distinct from our purposes.

Kar et al. [22] recently produced a related work (SMART SPICE). SMART SPICE is based on the concept of self-assessment and its goal is to popularize the SPICE model in companies. The SMART SPICE is based on the ISO/IEC 15504-5 process assessment model [23] and is a questionnaire tool and has three process dimensions, which are measurable by five attributes each. These constructs are organization, management, and engineering. The model is designed for SMEs to self-assess their process in the three dimensions specified. There are five questions in each of the five attributes. Thus, a total score of 125 is obtainable. The evaluation is such that the total score obtained is converted to 100%. The score from 0 to 50% is "poor", 51% to 65% is "fair", 66% to 80% is "average & manageable", 81% to 90% is "established", and above 90% is "well established". Companies can, therefore, see "where they stand" and figure out how to improve.

Although some of the existing models can benefit companies and researchers, there is still a need for a more comprehensive assessment model that does not only help measure process, but as well performance in projects, the expertise to carry out HCI activities, and what usability impacts are being achieved on products. Thus, we propose a holistic, unified model where the long-term goal is to be accessible to practitioners, enabling different team members to assess their HCI practice continuously and reflect. By discussing visions and goals, it is expected that practitioners improve communication and develop their HCI knowledge.

3 Development of the HCI Practice Assessment Model

De Bruin et al. [24] suggest that developing a maturity model involves six phases. The first phase defines the scope of the model, specifies its focus, whether that is a domain or a more general application, and the stakeholders. Stakeholders could be academia and practitioners for domain specific focus or government and combination for general focus. The second phase defines the design architecture of the model on which its development and application will be based. In carrying out the design tasks, the criteria to follow are to define the audience, which could be internal (e.g. executives and management) or external (e.g. auditors and partners). The other criteria are the method of application, including self-assessment, third-party assisted or by certified practitioners. The driver of application could be based on internal requirements, external requirements or both. The criterion for the application could be one entity/one region, multiple entities/multiple regions. The design of a model should consider the needs of intended users and how such needs will be met. The needs are based on *why* the user seeks to use the model, *how* the model can be applied to different organizational structures, *who* should be involved in applying the model and *what* can be derived from applying the model. Furthermore, the design should define the approach to representing maturity "as a number of cumulative stages where higher stages build on the requirement of lower states" [24]. The representation of maturity stages could be based on a top-down or bottom-up approach. How maturity stage should be reported to the user is another point of consideration when designing a maturity model. The third phase populates the content, including identifying what needs to be measured and how they can be measured. The fourth phase tests the model for relevance and rigor, and the fifth and sixth phases deploy and maintain the model. Figure 1 is the process defined by De-Bruin et al. [24].

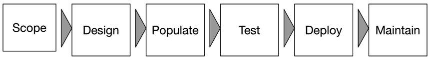

Fig. 1. Model development phases (De Bruin et al. [24])

We followed De Bruin et al.'s [24] procedure to define the scope, design, populate and test the HCI Practice Assessment Model. The scope of the model is domain specific and our focus is to support companies' practitioners and encourage the use of the model by HCI and software engineering researchers to investigate the state of HCI practice in software development companies. Figure 2 shows how we plan to evolve our current research.

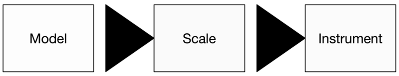

Fig. 2. Our planned activities

We developed the HCI Practice Assessment Model based on two existing models and an overview of the current literature on HCI practice. The two underlying models are the UMM-HCS [13] and KESSU Usability Design Process Model [18]. A new model could be built on the existing ones [25]. We chose UMM-HCS because of its robustness and usefulness as a descriptive checklist for companies to reflect on where they are in their processes and what they need to do to mature and why those actions would be important for them [16]. Similarly, KESSU has a unique focus on the performance of usability, not just the process [26]. A major consideration of KESSU is its usefulness for determining usability requirements in projects [27]. Finally, we developed the HCI Practice Assessment Model based on the existing literature on HCI practice and outcomes of our field investigations in software companies in two countries. The rationale for selecting the UMM-HCS and KESSU is that the two models match the needs for HCI practice assessment. The rationale for the combination of the two models is that one is focused more on development process (UMM-HCS) and the other on the performance of projects (KESSU). Thus, the reason for designing a new model is to build a complementary model, which can eventually be used in-house as an instrument (tool).

The HCI assessment model would be beneficial to software development project practitioners in three aspects: first, the use of the model would encourage reflective practice and not reactive practice. Second, practitioners can become more familiar with HCI concepts, techniques and methods in order to develop their HCI knowledge by searching online for relevant materials. Finally, companies can see aspects they need to pay more attention to, regarding their HCI practice.

3.1 Dimensions of the HCI Practice Assessment Model

Our proposed HCI practice assessment model is presented in Fig. 3. We have carefully studied the attributes in the UMM-HCS and KESSU and the literature as well as our field experience in two countries [28, 29]. We came up with 75 statement items for continuously assessing a company's HCI practice process. The approach used to structure the HCI practice assessment model is the Likert style questionnaire described by Fraser, Moultrie, and Gregory [30] used in De Paula, Fogliatto, and Cristofari [31]. The dimensions of the model are presented in form of statements, which the assessor can respond to by indicating the extent to which they are complying with each statement in line with their processes and project activities.

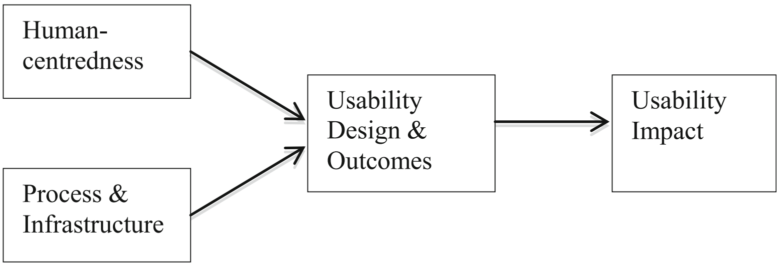

Fig. 3. Our proposed HCI practice assessment model

Usually, it is encouraged that key members of a development/project team, which includes the designer, developer, product owner, project manager, and system analyst do a peer and self-assessment of their process. The model should not be perceived as a 'gamifying' tool. The idea of peer and self-assessments is such that this kind of assessment provides opportunity for reflection. The need for reflection could arise when team members see that their assessment results differ.

Table 1 is the overview of the dimensions of the proposed HCI practice assessment model. As can be seen in Table 1, there are five attributes for human-centredness. Human-centredness is measured by a company's vision for, and awareness of human-centered development as well as ability to demonstrate competence and leadership in development projects. Basically, the dimension of human-centeredness is adapted from the principles described in the ISO 9241-210 framework [32].

Table 1. The dimensions of HCI practice assessment model.

Human-centredness	Process and infrastructure	Usability design and outcomes	Usability impact
Vision	Requirement process	Usability performance	Product effectiveness
Awareness	Design process		Product efficiency
Expertise	Evaluation process		Customer satisfaction
Team			
Leadership			

Our adaptations of the principles from ISO 9241-210 are specified as follows: (1) Design has to be based on the actual understanding of the user, task, and environment in which the intended system will be used; (2) The intended users of the system have to be actively involved in the requirement and analysis, design, and evaluation stages of the development; (3) The design of the intended system should be driven by user-centered evaluation; (4) The design of the intended system should be iterative; (5) The design of the intended system should address the appropriate experience needs of the intended user; and (6) The design should be approached with multidisciplinary skills and perspectives.

The assessor has to indicate the extent to which his/her company is following the 26 statements in the model of human-centredness. Similarly, the dimension of process and infrastructure has 20 statements. The dimension of process and infrastructure is determined by attitudes towards requirement, design, and evaluation processes in development projects. The dimension of usability design and outcomes is determined by attitudes towards usability performance in development projects. This dimension has 18 statements that clearly show what can be done to achieve effectiveness, efficiency, and satisfaction of the product being built. Finally, the dimension of usability impact is determined by the effectiveness, efficiency, and satisfaction of the product already built and deployed to the customer. There are 11 statements to answer; 3 statements for assessing the effectiveness and 3 for assessing efficiency of a product, and 5 statements are for assessing customer satisfaction. Ideally, it is expected that a company that pays attention to human-centredness, process and infrastructure, and usability design and outcomes dimensions should achieve great usability impact. The rubrics of the model and the full questionnaire can be accessed at https://goo.gl/NnA4o3.

3.2 Scoring

Our proposed model is a continuous representation model and not yet divided into maturity levels. The model is rated 0–4 for all of the statements. The statements are from “Not at all” (0), “Little” (1), “Moderately” (2), “A lot” (3), and “Completely” (4). Statements in usability design and outcomes are from a typical project. When answering, therefore, users have to first think of a project the company has executed, usually a recent project. We have not finalized the interpretation of the scores from this model. However, based on standard practices, the interim interpretations for each dimension are as follow: 0–49% (Worst Imaginable), 50–69% (Fair Enough), 70–84% (Good), 85–90% (Excellent), and 91–100% (Best Imaginable). Our interpretations are similar to those of Kar et al. [22].

4 Validation

The questionnaire underlying our proposed model was initially scrutinized by three senior HCI researchers, refined and later reviewed by two representatives from two Danish software/IT companies. In the following section we describe two case studies, which were recently carried out in two Estonian software development companies to validate our proposed model.

4.1 Case Studies

We administered the questionnaire to 6 senior professionals; 3 at each company, and 4 follow-up interviews were conducted afterwards with all the 3 professionals from one of the companies, and one professional from the second company. The profiles of the participants and their companies are presented in Table 2. Due to a condition of anonymity, the names used to describe the two companies are fictitious.

Table 2. Participants and their companies' profiles

Profile	Company	
	Alpha	*Beta*
Participant's role	Product Owner, User Experience Specialist, Project Manager	Usability Designer, Senior Developer, Project Manager
Company's size	Small (< 50 staff)	Medium (51-200 staff)
Years since operation	More than 10 years	More than 10 years
Project team size	5-10 people	5-10 people
Development culture	Agile (SCRUM)	Own method
Software development kind	Business support software	Business support software
Roles of HCI experts in projects	Usability Designer, User Experience Specialist, Graphics Designer	Usability Designer

4.1.1 Method

The selection of the professionals was purposive. The companies were found through snowball sampling [33].

Materials

Participants were sent the HCI practice assessment model questionnaire via emails and requested to fill it in and return to us. Follow-up interviews were afterwards conducted with 4 out of the 6 participants in their companies face to face or at other locations, in one occasion over Skype. The rest of the interviews were face-to-face. The interviews focused on ease of use, usefulness, reflectiveness of the model, and a few general questions.

4.2 Results

Interestingly and as envisaged, there were variations in how the participants assessed their processes. The full results are in Table 3. There are several variations in the assessments provided by the 6 representatives. The results from Alpha representatives indicate their HCI practice maturity is fair enough. However, although design issues are normally discussed in project meetings and in teams, it was understood that the level of knowledge of the participants about HCI was responsible for the variations in their assessments. Similar results were obtained for representatives from Beta. Overall, the HCI practice in Beta can also be said to be fair enough. However, the usability designer from the company did not answer the aspect of assessing usability impacts. The results of the maturity assessments from participants from Beta are very similar, which is approximately at 60% overall. We cannot fully ascertain why this is so. As earlier indicated, the questionnaire was sent to the practitioners via email. It may be possible they discussed some of the contents. The most interesting thing, however, is the differences in the results from each dimensions and the fact that practitioners can discuss their process together based on their peer and self-assessments. We, therefore, scheduled follow up interviews to gain more insights regarding the tool and to discuss the results with the participants.

Table 3. Participants' assessments of their companies' HCI practice

Company	Participants	HCI practice maturity dimensions			
		Human-centredness	Process & infrastructure	Usability design & outcomes	Usability impact
Alpha	Product owner	50%	˜54%	˜73%	˜59%
	User experience specialist	˜13%	˜30%	˜51%	˜27%
	Project manager	˜64%	˜60%	˜26%	˜56%
Beta	Senior developer	˜66%	˜60%	˜64%	˜64%
	Project manager	˜38%	˜61%	˜52%	˜45%
	Usability designer	˜78%	˜51%	˜48%	Not done

The follow-up interviews were done with four of the participants. The usability designer and the project manager from Beta were not available for the follow-up interviews. The interviews were focused on three major issues, which are discussed in the following subsections.

Ease of use

All the interviewees found the tool quite easy to use. However, the interviewees commented that some of the questions might be difficult for team members such as developers and project managers to answer. For example, the project manager from Alpha and the senior developer from Beta explained that they struggled when trying to answer a few questions. Precisely, both of them felt that only a designer could answer many of the questions. They both suggested to include an option for selecting 'Not Applicable or Don't Know' when the model is finally computerized into an online self-assessment tool.

Usefulness

All the interviewees perceived the tool to be very useful, not only to small and medium sized companies but large companies as well. For example, the user experience specialist and the project manager from Alpha, and the senior developer from Beta perceived the tool to enhance communication within teams. The product owner from Alpha reveals that the tool could act as a checklist reminder to ensure development teams are not leaving things out. All the participants perceived the tool to enhance sharing of vision and common values, and facilitate common goals and understanding. Furthermore, all the interviewees supported the use of visual analytics to interpret results in the online version. However, the user experience specialist from Alpha and the senior developer from Beta suggested adding a functionality that would allow companies to see how they compare with other companies in the industry. This suggestion reveals an area where the model could be enhanced in future, when it becomes an online tool.

Reflectiveness

The majority of the interviewees perceived the tool as one that could lead practitioners to reflect on the discrepancies in how people see or perceive things. The product owner from Alpha and the senior developer from Beta propose that the online version of the model should include hints, especially to facilitate understanding of key terms. For example, the project manager from Alpha wanted to understand what 'quality in use' means. The product owner from Alpha wanted to see a list of values, concepts, techniques, and methods, which one could use as guides to answering, related questions.

All the interviewees want the model to be accessible online. The product owner from Alpha and the senior developer from Beta would like to know how to improve themselves and suggest including a 'help' functionality in the online version in form of a library of online tools, and relevant literature as prescriptive measures to facilitating their HCI knowledge.

Two interesting issues came up during the interviews. One of the interviewees – the product owner in Alpha wanted to understand what HCI concepts are. An example was given of the concept of affordance. Prior to the time of the interview, the product owner had never heard of this concept. The lead researcher, therefore, explained the concept of affordance to him by using the example of the design of the door where the interview was held. It is interesting that the product owner immediately searched online and began to read further about the concept of affordance.

The second interesting thing is that while filling in the questionnaire, the project manager from Alpha was curious to know what ‘quality in use’ means and she therefore, searched online and as well raised the topic during the follow up interview. The insight we draw from the two issues is the possibility of the questionnaire to lead users to self-learning, thereby increasing their knowledge of HCI practice. Similarly, the senior developer from Beta shared an experience from a recent project in his company where the concept of cultural probe was used. The senior developer feels that practitioners would benefit a lot from having not just practical knowledge but theoretical knowledge underlying what they do as practitioners. The senior developer hinted that practitioners often use HCI concepts and techniques implicitly. This finding is similar to the study by Ardito et al. [7], which also revealed that practitioners are not able to explain what they do because in many cases, they possess implicit knowledge.

5 Discussion

Peer and self-assessment of software processes tend to be very promising. This is because these kinds of assessments give opportunities for reflection. De Bruin et al. [24] indicate three methods for the application of maturity models, which are: self-assessment, third-party assistance, and by a certified practitioner. Our literature search reveals that third party assisted assessment is commonly used. For example, Wendler [25] and Poeppelbuss et al. [34] noted that the CMMI is mostly used by researchers and practitioners. HCI and Software Engineering scholars have spent considerable amount of efforts in trying to bridge the gap between HCI and software engineering methods. However, it is imperative to deal with the causes of the gap rather than the effects. What is missing majorly in companies is a lack of awareness of HCI and how communication can be strengthened in development/project teams. Knowledge of HCI is considered a useful tool for development companies for value creation [10]. In response to these needs, our efforts have been to design a self-assessment model, which allows team members to do peer and self-assessments of their development/project processes in line with the HCI dimensions. Our model was built from two relevant models namely, the UMM-HCS and KESSU. According to De Bruin et al. [24] used in Poeppelbuss et al. [34] there are three basic purposes, which a maturity model seeks to serve. The purposes of maturity models are descriptive, comparative, and prescriptive. Different scholars have used the UMM-HCS for the broad purposes earlier stated. For example, Hussein et al. [17] used the UMM-HCS to describe the state of usability awareness in 72 Malaysian IT organizations. Similarly, Liu [16] used the UMM-HCS to described the state of usability engineering by conducting semi-structured interviews in five Chinese IT enterprises. Other scholars have applied the UMM-HCS in combination with other models to explore whether they are adequate for promoting usability in in-house projects. For example, Paananen [35] combined the UMM-HCS and CMMI to investigate how an in-house development process geared towards the development of an Enterprise Resource Planning (ERP) software could be improved. Paananen [35] concluded that applying HCI activities and usability methods could increase the integration of the UMM-HCS and CMMI models. The conclusion of the study is logical to our approach for designing the

HCI practice assessment model. In our case we combined the UMM-HCS and KESSU models to enable practitioners achieve a comprehensive focus on HCI and usability especially in development projects and processes.

The current basic purpose of our model is reflective, that is to let software development companies know "where they stand" and reflect on what they need to know or do in order to mature HCI practice. This could be achieved through improving team communication and increasing knowledge of HCI. We feel that one indication that the model is promising is the outcome of the interviews with four out of the six representatives covered in this study. So far, our experience is positive; thus, we will conduct more cross-sectional studies in companies in order to examine in greater detail how the model aids practitioners' reflective practice and fosters team communication. Our model can be used hand-in-hand as complementary tool to existing software process improvement self-assessment tool such as the SMART SPICE.

6 Conclusion

This paper presents the description of a peer and self-assessment HCI practice assessment model whose goal is to support HCI practice maturation in development projects and processes. Our proposed model is, however, not yet divided into maturity levels. Our previous findings from field investigations in companies, and the literature show that small and medium software companies in many cases cannot afford third party and very expensive tools such as the CMMI. Small and medium companies dominate many software industries globally and they need to be supported. Small and medium companies are involved with development projects to support other businesses. The level of HCI awareness is low in this set of companies and there is danger of expending unnecessary costs on training and bug fixing, especially when companies neglect HCI dimensions. In order to address this problem, we developed a peer and self-assessment model, to facilitate awareness of HCI dimensions in small and medium software development companies, and provide opportunities for self-learning, and enhancement of team communication through reflective practice.

The outcomes of the case studies in the two companies where the model has been introduced indicate that the use of the model could help development/project team members to improve their communication, share vision and common values, and pursue common goals and understanding in teams. The HCI practice assessment model could foster self-awareness, self-consciousness, self-learning, and reflective thinking among practitioners and propel companies towards goal setting, and process re-engineering for the maturation of HCI practice. Participants indicate areas where the model should be improved for future deployment. Furthermore, there is an insight gained that communication could improve when team members conduct a peer and self-assessment of their own processes. This paper contributes to the literature on the facilitation and maturation of HCI practice in IT/software development companies' especially agile processes, which have some commonalities with the human-centred design principles. The paper also complements existing work on software process improvement by highlighting the importance of usability design and evaluation.

6.1 Limitations and Further Work

We have only introduced our proposed model to two software companies and this does not ensure empirical rigor and robustness of the model. However, there are insights that the model is interesting to the employees of the companies and could at least serve as checklist to remind them not to leave important things out. There is an indication that the use of the model as a peer and self-assessment tool could lead practitioners to improve on team communication and as well increases their awareness of HCI and become more familiar with HCI dimensions for achieving usable and user-friendly software. Our next task is to deploy the model to large audience for the purpose of enhancing its validity, and determining its reliability, and generalizability. After these efforts, we plan to introduce the validated model to more companies and conduct ethnographic studies in companies where the tool has been introduced. The purpose is to understand how the concepts in the tool are manifested in the teams over time and how different team members conceptualize the tool.

Acknowledgement. The authors would like to thank Professor Nicola Bidwell for her assistance.

References

1. Bevan, N.: International standards for usability should be more widely used. J. Usability Stud. **4**, 106–113 (2009)
2. Boivie, I., Åborg, C., Persson, J., Löfberg, M.: Why usability gets lost or usability in in-house software development. Interact. Comput. **15**, 623–639 (2003)
3. García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M.: Approaches to promote product quality within software process improvement initiatives: a mapping study. J. Syst. Softw. **103**, 150–166 (2015)
4. Svanæs, D., Gulliksen, J.: Understanding the Context of Design – Towards Tactical User Centered Design. In: Proceedings: NordiCHI 2008, pp. 353–362 (2008)
5. Kieffer, S., Vanderdonckt, J.: STRATUS : a questionnaire for strategic usability assessment. In: Proceedings of the SAC 2016 Conference, pp. 1–8. ACM, Pisa (2016)
6. Maxim, B.R., Kessentini, M.: An introduction to modern software quality assurance. In: Mistrik, I., Soley, R., Ali, N., Grundy, J., Tekinerdogan, B. (eds.) Software Quality Assurance: In Large Scale and Complex Software-Intensive Systems, pp. 19–46. Morgan Kaufmann, Tokyo (2016)
7. Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R.: Investigating and promoting UX practice in industry: an experimental study. Int. J. Hum Comput Stud. **72**, 542–551 (2014)
8. Kuusinen, K., Mikkonen, T., Pakarinen, S.: Agile user experience development in a large software organization: good expertise but limited impact. In: Winckler, M., Forbrig, P., Bernhaupt, R. (eds.) HCSE 2012. LNCS, vol. 7623, pp. 94–111. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34347-6_6
9. Misra, S.C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. J. Syst. Softw. **82**, 1869–1890 (2009)

10. Liem, A., Sanders, Elizabeth B.-N.: The impact of human-centred design workshops in strategic design projects. In: Kurosu, M. (ed.) HCD 2011. LNCS, vol. 6776, pp. 110–119. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21753-1_13
11. Boivie, I., Gulliksen, J., Goransson, B.: The lonesome cowboy: A study of the usability designer role in systems development. Interact. Comput. **18**, 601–634 (2006)
12. Rogers, E.M.: Diffusion of Innovations. Free Press, New York (2003)
13. Earthy, J.: Usability Maturity Model: Human Centredness Scale (1998)
14. ISO: ISO/IEC TR 15504-2: Information technology – Software process assessment – Part 2: A reference model for processes and process capability (1998)
15. ISO: ISO TR 15504 Part 5, Information technology – Software Process Assessment – Part 5: An assessment model and indicator guidance (1999)
16. Liu, Z.: An organisational human-centeredness assessment at Chinese software enterprises. In: Proceedings of APCHI 2002, pp. 251–259. Academic Press, London (2002)
17. Hussein, I., Mahmud, M., Tap, A.O.: A survey of usability awareness in Malaysia IT industry. In: Proceedings of the 2011 International Conference on User Science and Engineering (i-USEr), pp. 146–151 (2011)
18. Jokela, T.: The KESSU usability design process model, Oulu (2004)
19. ISO: ISO 13407. Human-centred design processes for interactive systems, Geneve, Switzerland (1999)
20. ISO: ISO/TR 18529: Ergonomics — Ergonomics of human-system interaction — Human-centred lifecycle process descriptions, Geneva (2000)
21. Grama, C.: Adapting Lean User Experience Process for Enterprise Environment (2016)
22. Kar, S., Das, S., Kumar Rath, A., Kar, S.K.: Self-assessment model and review technique for SPICE: SMART SPICE. In: Mas, A., Mesquida, A., Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 222–232. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_20
23. ISO: ISO/IEC 15504-5: Information technology – Process Assessment – Part 5: An exemplar Process Assessment Model (2006)
24. de Bruin, T., Freeze, R., Kulkarni, U., Rosemann, M.: Understanding the main phases of developing a maturity assessment model. In: Proceedings of the 16th Australasian Conference on Information Systems, pp. 1–10. AIS, Sydney (2005)
25. Wendler, R.: The maturity of maturity model research: A systematic mapping study. Inf. Softw. Technol. **54**, 1317–1339 (2012)
26. Bruno, V.: Improving Usability Outcomes in IS Projects: The Views of Usability Practitioners (2011)
27. Kajaste, I., Mathew, D., Peltomäki, S., Poranen, T.: Using usability experts to improve software quality. In: Proceedings of the INSPIRE Conference, pp. 1–15 (2007)
28. Ogunyemi, A., Lamas, D., Sarapuu, H., da Rosa, I.B.: Current state of HCI practice in the Estonian software development industry. In: Stephanidis, C. (ed.) HCI 2015. CCIS, vol. 529, pp. 170–175. Springer, Cham (2015). doi:10.1007/978-3-319-21383-5_29
29. Ogunyemi, A.A., Lamas, D., Adagunodo, E.R., Loizides, F., Da Rosa, I.B.: Theory, practice and policy: an inquiry into the uptake of HCI practices in the software industry of a developing country. Int. J. Hum.-Comput. Interact. **32**, 665–681 (2016)
30. Fraser, P., Moultrie, J., Gregory, M.: The use of maturity models/grids as a tool in assessing product development capability : a review. In: Proceedings of the IEEE International Engineering Management Conference, pp. 18–20. IEEE, Cambridge (2002)
31. De Paula, I.C., Fogliatto, F.S., Cristofari, C.A.: Method for assessing the maturity of product development management: A proposal. Afr. J. Bus. Manag. **5**, 10285–10302 (2012)
32. ISO: Ergonomics of Human-System Interaction - Part 210: Human-Centred Design for Interactive Systems. In: ISO 9241-210:2010, pp. 1–32. ISO (2010)

33. Hussey, J., Hussey, R.: Business Research: A Practical Guide for Undergraduate and Postgraduate Students. Macmillan, London (1997)
34. Poeppelbuss, J., Niehaves, B., Simons, A., Becker, J.: Maturity models in information systems research: literature search and analysis. Commun. Assoc. Inf. Syst. **29**, 505–532 (2011)
35. Paananen, A.: Improving In-House Software Development Process: A User-Centered Approach (2014)

Cultural Issues and Impacts of Software Process in Very Small Entities (VSEs)

Tatsuya Nonoyama[1,2(✉)], Lian Wen[1,2], Terry Rout[1], and David Tuffley[1,2]

[1] Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia
{t.nonoyama,l.wen,t.rout,d.tuffley}@griffith.edu.au
[2] School of Information and Communication Technology, Griffith University, 170 Kessels Rd, Brisbane, QLD 4111, Australia

Abstract. ISO/IEC29110 is an international standard of software lifecycle for small software companies also known as Very Small Entities (VSEs). While VSEs come from a diversity of cultural backgrounds, the current ISO/IEC29110 for VSEs does not address these cultural variations. VSEs from various cultural backgrounds might therefore find it difficult to adapt such a standard. This paper raises the issue that the current ISO/IEC29110 should recognize the impacts of cultural variation on software processes and cultural suggestions. It also point out one cultural dimension has a significant impact on software processes and their efficacy. Furthermore, the concepts of cultural consideration should not be limited to regions but, also cover the management perspective of individual VSE. In this paper, we identified two opposite cultural types which affect their software processes significantly. We propose that to make software process standards more practical for VSEs from different cultural backgrounds.

Keywords: Organizational culture · Cultural diversity · Cultural variation · Very small entities · Software process · Software standards · Cultural dimensions

1 Introduction

The successful adaption of ISO/IEC29110 has improved many small software companies over the years. The standard covers a variety of software processes and check lists to support the VSE. Prior in 2014, ISO/IEC15504 was too complex and expensive for small software companies to adapt (Abe et al. 2012). A light weight standard was specially built for small software companies also known as Very Small Entities (VSE). According to International Organizational for Standardization (ISO) classification of the VSE can be a project or a small company which consists of 25 or less employees (ISO/IEC TR 29110-5-1-2 2011; Laporte et al. 2008).

However, not all VSEs has received the benefits from the current ISO/IEC29110. Some VSEs argue that ISO/IEC29110 do not adequately address the cultural backgrounds of the VSE. In fact, the standard is not compatible to fit their organizational cultures (Suryaningrum 2012; Roldan 2015). They are concerned that lack of cultural

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 70–81, 2017.
DOI: 10.1007/978-3-319-67383-7_6

support in software process can impact on their software quality and the overall software capability.

The purpose of this paper is to strengthen the knowledge and information with regards to cultural issues of VSEs from different cultural backgrounds. The paper is not intended to address all aspects of cultural issues in VSE, only to address some aspects of cultural issues to provide closely tailored software process and management concepts for VSE with different cultural backgrounds. This paper is intended for ISO standard developers and researchers to help improve the efficiency of ISO/IEC standards. Regional differentiations between VSE are not significant enough to support compatibility of ISO/IEC29110.

The structure of this paper outlines below. The significance of the problem highlights the importance that made cultural suggestions has the potential to improve the overall software capability of the VSE. The research question of this paper is divided into two important objectives to tackle the complex phenomenon of VSE's cultural background. Furthermore, in the literature review, we have selected a particular cultural dimension from Geert Hofstede's work. The cultural element of "Individualist VS Collectivist" was analyzed for addressing the research question.

The outcomes of the paper include a table that defines the characteristics of each VSE depending on their management perspective styles. Based on the six cultural dimensions framework, we have proposed tailor made cultural suggestions. The discussion, explains VSE's cultural backgrounds can have overlapping issues as well as expected outcome for applying the cultural suggestions. In conclusion, the paper recaps on how cultural suggestions can benefit ISO/IEC developers and researchers then briefly discusses future direction of this paper.

It is important for ISO/IEC standards developers and researchers to understand the complex cultural backgrounds of software development across the regions. Norbury (2003) argues that, the cultural backgrounds between VSE are different as well as across the regions (Western countries and Asian countries). Furthermore, the organizational culture highly impacts on the way people adapt new organizational culture, software processes and management concepts. Our cultural suggestion could be applied to small medium enterprises (SMEs) with minor modifications to fit the characteristic trends of VSEs.

2 Literature Review

In this section, we investigated several literature studies in relations to culture and software process. The literature review covers the background information of software standards. Furthermore, we explored the cultural dimensions framework by Geert Hofstede. The key research question of this paper is: how does cultural diversity/issues effect on VSE's implementation? The paper then divides the key question into two objectives. Objective 1: what are the cultural issues among VSE when adapting software processes? Objective 2: how does the VSE's organizational culture impact on the content of software processes? We understand that all VSEs require adjustments to the software processes from ISO/IEC29110 to meet their cultural backgrounds.

Every VSE has different ways to manage software processes, software quality and people. The feasibility to adapt ISO/IEC29110 is dependent upon VSE's cultural backgrounds (Nonoyama et al. 2016). Despite the fact that the current ISO/IEC29110 provides technical support to improve software capability, however, it does not fully address cultural variation in VSEs. It is important to address cultural backgrounds to help individual VSE to adapt. Furthermore, the standard should include cultural suggestions to resolve the VSE's cultural issues. These cultural issues include management perspective styles, work conditions and organizational culture. Facing these cultural issues not only impacts on the standard compatibility but, also creates a serious financial issue in a long term. Although, these cultural issues are not limited to VSEs, the size of software companies increases the complexity of adapting ISO standards in general. This is the reason why we need to acknowledge the various cultural backgrounds of VSEs. In comparison to larger scale companies such as SMEs, VSEs with strong cultural backgrounds tends to have their own traditional (native) approach in software development.

2.1 Software Standards for VSEs

In 2011, International for Standardization Organization (ISO) and International Electrotechnical Commission has published a set of standard called ISO/IEC29110. ISO/IEC 29110 provides a variety of profiles for VSEs. The standard was deployed only the relevant software processes from the original software standard called ISO/IEC 12207. The purpose of ISO/IEC29110 is to facilitate measurable and tangible benefits to the clients through the expected outcomes (Wen and Rout 2012). To help the VSE to satisfy the expected outcomes, it offers the guideline for establishing appropriate software processes (Nonoyama et al. 2016). The potential benefit in ISO/IEC29110 is that VSEs can establish more consistent and tailored software processes to improve their software capability level.

2.2 Hofstede's Cultural Dimensions Framework

We first analyze national culture then we narrow it down into the organizational culture of the VSE. The six cultural dimensions framework by Geert Hofstede is a good starting point to understand national culture and the impacts of software processes. The purpose of cultural dimensions framework is to provide comprehensive knowledge of how values in the workplace are influenced by national culture (Hofstede 1984). In order to understand VSE's organizational cultures, it is important to understand the values of national cultures. The framework also suggests that the organizational culture and the national culture should not be separated or cannot be separated. Sordo (2015) argue that Hofstede's cultural dimension framework provides a strong linkage between the national culture and the organizational culture. In this paper, we also want to stress that cultural variation in individualist and collectivist have some similarities in the context of software development.

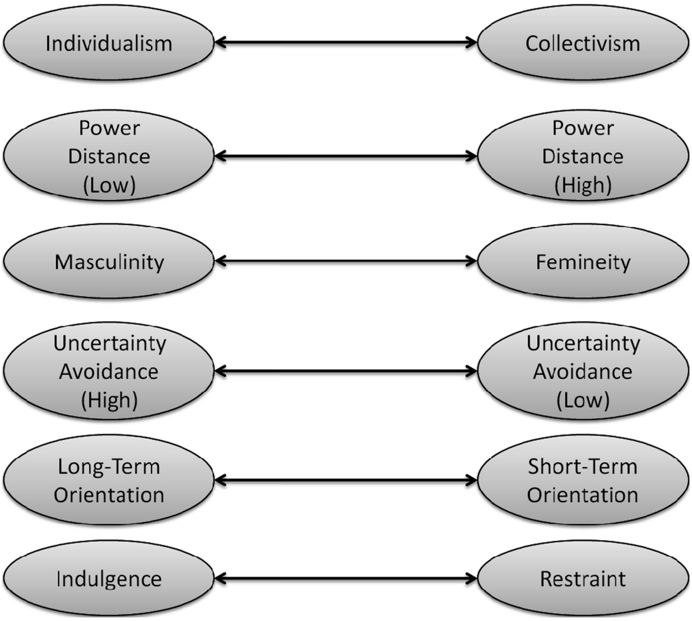

Fig. 1. Hofstede's cultural dimensions framework (Hofstede 1984; Sordo 2015).

In Fig. 1, presents the latest version of Hofstede's cultural dimensions framework. In the original framework, it only presented five cultural dimensions. A new dimension called "Indulgence vs Restraint" was introduced in the recent studies (Sordo 2015). A new cultural dimension is included in Fig. 1.

We have highlighted one relevant cultural dimension which closely relates to organizational culture in VSEs. The cultural dimension of "Collectivist VS Individualist" shows a significant reflection on how VSE manage and adapt software process (Gordon and O'Connor 2015). Adaptation of software process differs based on the dominant management perspective of the VSE.

2.3 Collectivist vs Individualist

In collectivist culture, the group work and group cohesion are highly valued. It is important for collectivist culture to belong to a particular group or community (Hofstede 1984). This also applies to VSE and their software processes. The software process includes a large amount of communications and interactions. Furthermore, employees are expected to cooperate in creating harmony between workers and clients (Suryaningrum 2012). As the result, collectivist culture VSE has slower software process time compared to the VSE with individualist culture. The software contracts and goals are negotiable depending on the client's organizational status.

On the other hand, VSE with individualist culture highly values individual's expertise and experience. They primary show interests on close relatives and friends more than a strong connection towards a particular organization or community

(Hofstede 1984). Additionally, the organizational status is not the first thing to be considered. Instead, they consider organizational policies and software capability level at initial stage of software development (Laporte et al. 2008). VSE with individual culture, have a clear relationship between clients and employees. The client communication should only be work related matters. In collectivist environment, communication needs to be on personal level (Suryaningrum 2012). However, this rarely occurs in the context of individualist culture.

For example: Japanese and Chinese VSEs are generally viewed as collectivist culture. They prefer to have longer software process which closely aligns with their guidance of team work management (management by passion) (Norbury 2003; Coleman and Basten 2015). America is a commonly used example of individualist culture. American based VSE are viewed as individualist culture when developing software. It is also known as task based management with clear individual objectives (Sordo 2015). Therefore, software development time is relatively shorter than the VSE in collectivist culture.

However, not all countries have the clear boundary between national culture and organizational culture. Another example is: Taiwan and Hong Kong generally do not share Chinese collectivist culture. The organizational cultures in Taiwan and Hong Kong are much more diverse than the mainland China. According to Norbury (2003); Roldan (2015); Saith (2008); Sordo (2015) the organizational cultures in Hong Kong and Taiwan are categorized as individualist culture whereas in China, collectivist culture is dominant in both national culture and organizational culture. Some argue that, places such as Hong Kong and Taiwan still resembles Chinese collectivist culture which shows a strong cultural association with the mainland China.

3 Cultural Suggestions

3.1 Characteristics of Management Perspective Styles

This section highlights the differences between two management perspective styles in VSEs. According to Table 1, some characteristics are overlapping each other. The table also shows some correlations from individualist and collectivist cultures which reflects on their organizational culture. However, as mentioned in Sect. 2.3, in some cases, the national culture do not reflect on how people do their businesses (organizational culture). This obviously impacts on software processes and software development in VSEs.

The purpose of Table 1 is to compare two management perspective styles “Task based and Team work based managements which commonly witnessed across the VSE’s cultural backgrounds. The characteristic of work culture indicated the major difference in prioritizing goals and objectives (Clarke and O’Connor 2011; Nonoyama et al 2016). Furthermore, task based management corresponded higher importance on archiving individual and organizational achievements. Team work management on the other hand, is focused on their organizational goals and objectives. The cultural backgrounds are differentiated and measured as characteristics 1.1 to 1.5. Detailed information of each characteristic element is presented below Table 1.

Table 1. Comparisons of task based and team work management perspective styles

Characteristic element	Task based management	Team work management
1.1 Working Hours	Over time work is not acceptable. Acceptable in emergencies	Over time work is generally accepted
1.2 Working Conditions	Work conditions are clearly established	Work conditions are clearly established
1.3 Work Culture	Clear objectives and goals	Objectives and goals are changeable depending on the clients and senior employees
1.4 Communications	Direct approach	Indirect approach
1.5 Roles & Responsibilities	Clearly defined Task allocation should not change	Clearly defined Task allocation is changeable

(Norbury 2003; Nonoyama et al 2016; Perlow 2001; Laporte et al. 2008; Roldan 2015).

1.1 Working Hours

Different management perspective has different work hours at workplace. Generally, task based management do not accept over time work unless it is an emergency (Perlow 2001). On the other hand, team work management tends to accept longer hours of commitment in general.

1.2 Work Conditions

The work conditions are established in both management perspective styles. Based on the work policy, the average work hour is 40–45 h (Perlow 2001; Norbury 2003). In some cases, employees were told to work more than 45 h per week to catch up with clients. This is common in team work management environment where employees were obligated to entertain clients or work overtime.

1.3 Work Culture

Task based management generally prefers to have clear goals, objectives and justifications. Furthermore, in task based management, shows higher importance on individual objectives and goals (Roldan 2015). Whereas in team work management, they consider their organizational goals and objectives are the most important aspects in the project (Norbury 2003). Another important aspect in team work management is that clients and senior employees have more power to influence on software development.

1.4 Communications

Direct approach is common in task based management environment. The client communication is crucial yet, only performed for business purposes (Sordo 2015). In comparison to task based management, team work management tends to interact and communicate frequently to the clients. In some cases, it is a norm to entertain high profile clients outside of work hours (Norbury 2003). For team work management, indirect communication is considered a formal approach to understand clients on the personal level.

1.5 Roles & Responsibilities

Task based management establish clear roles and responsibilities of each employee. Organizations (VSEs) are expected to accomplish individual tasks. Employees from task based management environment do not expect to accomplish tasks from outside of their boundaries (Roldan 2015; Laporte et al. 2008). However, in team work management environment, all employees are expected work towards their organizational goals. The boundary between delegated tasks and additional (external) tasks are not clearly outlined. In other words, employee's expertise and knowledge are frequently shared among other employees to accomplish the project.

3.2 Software Process Impacts in Two Cultural Types

Based on the literature review, it would be desirable to differentiate suggestions based on the dominant management perspective of the VSE. Providing suggestions based upon the regional segregation is too general and insufficient. Instead, the VSE's cultural backgrounds should determine the content of cultural suggestions.

The purpose of this Table 2 is to determine the software process impacts between individualist culture and Collectivist culture. We have separated the VSE's cultural backgrounds into two cultural types A and B. The cultural type A: as individualist and cultural type B as collectivist. It is highly recommended for ISO/IEC developers and researchers to acknowledge these characteristic elements for future improvements. Both suggestions include the basic concepts to select a suitable software process that fits their cultural backgrounds.

Table 2. The comparisons of individualist culture and collectivist culture

Impact element	Cultural type A (Individualist)	Cultural type B (Collectivist)
(2.1) Management perspective	Task based	Team work (Management by passion)
(2.2) Adaption of software process	A detailed plan, objectives and goals (Slow paced decisions)	Flexible plans to prepare for client expectations (Fast paced decisions)
(2.3) Software process choice	Fast paced	Slow paced
Country example	U.S. and Hong Kong	Japan and India

(Sharp et al. 1998; Perlow 2001; Sordo 2015).

3.3 Characteristics of Cultural Type A: Individualist

A certain culture type preferred their goals and objectives to be clearly defined. In this type of VSE, an employee dislikes the ambiguity when adapting software process and development. VSE with individualist culture would expect their software process to be well organized with a clear justification. Furthermore, VSEs with individualist culture are less likely to seek support compared to collectivist culture based VSEs. This is because; individual expertise and knowledge are highly valued. Team work and common work culture are still important yet; the individual differences are accepted in the individualist culture (Perlow 2001; Sharp et al. 1998). In order to increase consistency in software development, they tend to discuss roles & responsibilities. VSEs with individualist culture preferred to keep the client communications/interactions to the (basic) minimum.

Unlike collectivist culture based VSEs, individualist culture based VSEs tends to overly eliminates slacks including (social interactions and client communications). The elimination of slacks highly impacts on the software process time. Depending on the content of slack, it can be a positive or negative outcome (Sordo 2015). It is always important for individualist VSEs to carefully consider eliminating client interaction/communication slacks.

Additionally, clear relationship boundary between clients and employees. A moderate amount of direct communication can help to maintain the harmony between client and employee. VSEs with individualist culture have equal bargain power between client and employee. Both employees and clients are entitled to openly discuss their concerns. It is also common in individualist culture to respect each other's privacy (a personal life and non-work related matters) and keep it to the minimum.

3.4 Cultural Suggestions for Cultural Type A: Individualist

Based on these characteristic elements, VSE with cultural type A, it is highly recommended to apply these suggestions.

1. Increase interaction and communication level among clients and other fellow employees in the VSE. Show some acknowledgment of different opinions, expertise and organizational culture.
2. Learn to adjust communication approach based on the VSE/clients. Direct approach and communication can be inappropriate in some cultures or the VSE's organizational culture. In the initial stages of a project, indirectly communicate with clients. Once the harmony was created, establish clear and direct communications with clients.
3. Avoid focusing only on technical issues in the VSE (poor coding, management concepts and testing procedures). Non-technical or cultural issues (incompatibility of standards) are equally important to be resolved.

3.5 Characteristics of Cultural Type B: Collectivist

VSEs with cultural type B generally accept the ambiguities during the software process and development. It is acceptable and norms to work over time. Team work engagement and common shared beliefs are the primary traits of collectivist culture. Therefore, a software process is necessary to handle a large communication/interaction management.

A large volume of communications and interactions are frequently take place during the software development. A frequent communication helps the VSE to polishing up the details of software and client expectations. However, in some scenarios, an excessive amount of communication can be a disaster (Norbury 2003). The confusions between clients and employees can impact on the quality of software process and development.

Another important characteristic of collectivist culture is that social engagement is very common among employees and high profile clients. Most of these client engagement and entertainment occurred outside of work hours (Suryaningrum 2012). As the results of these client engagement and communication, the software process time becomes slower than individualist culture A. Detailing out client requirements and expectations can be time consuming. Additionally, a frequent change in software goals and objectives can dramatically slow down the overall software process (Yan and Li 2015). A large volume of client communications and interactions make the software process more complex to be adapted.

3.6 Cultural Suggestions for Cultural Type B: Collectivist

Based on these characteristic elements, VSE with cultural type B, it is highly recommended to apply these suggestions.

1. Minimize communications between employees and clients. Time management concept is needed to help the VSE to minimize the volume of communication and interactions with clients.
2. Improve individual expertise and knowledge of employees. It is important to provide practical guideline to train employees to be more independent. Ensure to provide potential benefits of implementing software processes.
3. Focus more on technical side of software process and software development. It is important to highlight the importance of establishing clear objectives and goals can reach out international audience.

4 Discussion

Major challenge of this paper is tackling the questions that associate with cultural issues. As mentioned before, some VSE's cultural backgrounds may overlap between individualist culture and collectivist culture. The effect of these suggestions can be time consuming and can pose risks for some VSEs. It is also worth highlighting the point that, a close alignment between ISO/IEC29110 and VSEs may be limited to a certain cultural backgrounds. Having the basic suggestions could only reduce the complexity of cultural issues. It may not be sufficient to fully understand the cultural backgrounds of VSEs. In other words, a new set of standard may be needed to convince the VSE for adapting software processes.

The literature review indicated that work condition of the VSE was standardized. Both cultural types were exposed in similar work hour conditions. In the cultural suggestions section, the Tables 1 and 2 are useful to gain knowledge in management differences and the organizational cultures of the VSE. Both cultural types also indicated some overlapping characteristic elements (Coleman and Basten 2015). In other words, both culture types could improve on a particular characteristic element. There is no clear boundary between VSE's cultures including cultural issues and management perspective styles.

The expected outcome of this paper is to reduce the cultural incompatibility when adapting ISO/IEC29110. For VSEs with cultural issues, these suggestions should facilitate in adapting more stable and suitable software process in place. Acknowledgment on cultural backgrounds could motivate VSEs to adapt ISO/IEC29110. It is important for ISO/IEC standard developers and researchers to understand different cultural backgrounds can impact on the adaptations of software processes.

The suggestions for both cultural types are designed to save resources and software reworks during the software development. It should also clarify the VSE to adapt which software process features would benefit them. For cultural type A tends to reflects on the characteristic element of task based management perspective style. In this cultural background, employees are expected to be independent and the overall software process become faster than collectivist culture based VSEs. On the other hand, collectivist VSEs has a strong relation to team work based management perspective style. This also indicated that collectivist VSEs should shorten the software process time by reducing the amount of client communication and interaction. We would like to highlight that these suggestions can facilitate a closer alignment between the concepts of ISO/IEC29110 and VSE's cultural backgrounds.

Furthermore, our suggestions can be enhanced to fit the characteristics/situational factors of larger scale software companies. It would be important to investigate the organizational cultures of SMEs and redesign the differences to tailor the software process for SME's audience group. Since SMEs may have higher resistance for change than VSEs with 25 employees or less (ISO/IEC29110 2011). The major difference between VSEs and SMEs is the size of the software company. In other words, the number of employees can impact on the software capability, software development costs and the organizational culture.

5 Conclusion

We have concluded that the cultural backgrounds should not be determined by geographical regions of the VSE alone. The VSE's cultural background should be analyzed based on their work hours, work culture and management perspective styles. Therefore, the generic ISO/IEC29110 is not sufficient to address the VSE's cultural backgrounds and their cultural issues. It is important for standard developers and researchers to understand the national cultures of the VSE as well as the generic trends of organizational culture. We consider these cultural suggestions would help the current ISO/IEC29110 to more feasible for VSEs with strong cultural backgrounds. The paper highlights that acknowledgment in cultural backgrounds will help ISO standard developers to suggest tailor made software processes for VSEs. Based on our literature review, the organizational cultures in different regions of VSEs were different yet; some overlaps between cultural types. In addition, some characteristics in VSEs can overlap with SME's cultural backgrounds.

In this paper, we want to stress that determining the national cultures of the VSE would be the initial stage to gain the basic knowledge in the VSE's organizational cultures. We have provided Hofstede's cultural dimension frameworks to link the cultural impacts and the VSE's characteristic trends. The paper narrowed it down the national culture and applied to organizational cultures of the VSE. The comparison table of task based management and team work based Management (Table 1) was developed to highlight the major differences in VSEs.

Based on the findings of literature review, we have proposed a Table 1 for two management perspective styles. In this paper, we linked these perspective styles with two cultural types. Cultural type A includes the major characteristic of task based management perspective style. On the other hand, cultural type B, resembled as a team work management perspective style.

Furthermore, we have developed a Table 2 which consists of cultural suggestions for VSEs. The suggestions are tailored into cultural type A based VSEs and B as collectivist based VSEs. The cultural type A indicated that the overall software process should be designed to spend more time and resources on client interactions. The overall process may become slower yet, capable to capture greater details of client requests. As the results, it would gradually sustain the balance between client requirements and software requirements. On the other hand, the cultural type B should focus on the technical side of software process. In other words, the software process should be designed to reduce a large amount of communications. It is highly recommended to design the software process with a basic management concept to clarify goals and software objectives. It is also important to eliminate unnecessary slacks including any unnecessary client entertainment and over communicating clients on the personal level. The Cultural type B should spend more time and resources to establish more consistent software processes.

Our future research would involve strengthening the knowledge in the VSE's national culture and the organizational culture. This is important for our future works because, we can develop more detailed suggestions to help ISO/IEC29110 for more specific to resolve cultural diversity. This includes enhancing our cultural suggestions towards SMEs as well. We would also want to stress that, the future work should address the following challenges. The first challenge was related to the cultural linkage between Hofstede's cultural dimension frameworks and the concepts of software processes and software development. The frameworks were originally adopted for addressing the national culture between different countries. Having said that, we needed to find the evidence to suggest that cultural factor highly affects the way people adapt software processes. The second challenge was related to analyzing the frameworks of Geert Hofstede. It was particularly difficult to select which cultural dimension had a strong impact on the characteristics of the VSE. Each dimension has more or less impacts on their software development yet; requires time-consuming research studies. Lastly, addressing these challenges would help align current ISO/IEC29110 to be more practical and relevant for VSEs with compatibly concerns and other cultural issues.

References

Abe, M., Troilo, M., Batsaikhan, O.: Financing small and medium enterprises in Asia and the Pacific. J. Entrepreneur. Public Policy **4**(1), 2–32 (2012)

Coleman, D., Basten, S.: The death of the west: an alternative view. Popul. Investigat. Committee **69**(1), 107–118 (2015)

Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: towards a comprehensive reference framework. Inf. Softw. Technol. **54**(5), 433–447 (2011)

Gordon, M.S., O'Connor, R.V.: Understanding the gap between software process practices and actual practice in very small companies. Softw. Qual. J. **9282**(6), 1–22 (2015)

Hofstede, G.: The cultural relativity of the quality of life concept. Acad. Manage. **9**(3), 389–398 (1984)

ISO/IEC TR 29110-5-1-2: Software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile (2011)

Laporte, C.Y., Alexandre, S., Renault, A.: Developing international standards for very small enterprises. IEEE Comput. Soc., Willowdale (2008)

Norbury, P.: A Traveller's Guide to Custom and Culture. Graphic Arts Center, Kuperard (2003)

Perlow, L.A.: Time to coordinate toward an understanding of work-time standards and norms in a multicountry study of software engineers. SAGE J. **28**(1), 91–111 (2001)

Roldan, M.D.: Sustaining "Lilliputs" in the global knowledge-based economy: prospects for micro, small, and mediumscale enterprises in the developing world. Europ. J. Sustain. Develop. **4**(2), 269–274 (2015)

Nonoyama, T., Wen, L., Rout, T.: Current challenges and proposed software improvement process for VSEs in developing countries. Commun. Comput. Inf. Sci. **609**, 437–444 (2016)

Saith, A.: China and India: the institutional roots of differential performance. Inst. Soc. Stud. **39**(5), 723–757 (2008)

Sharp, H.C., Hovenden, F.M., Woodman, M.: Factors affecting the adoption and evolution of software quality management systems. Technical report, Computing Department, the Open University, TR 98/17 (1998)

Sordo, A.I.: Beyond hofstede's cultural dimensions theory: approaching a multicultural audience. Skyword, Web site (2015). https://www.skyword.com/contentstandard/marketing/beyond-hofstedes-cultural-dimensions-theory-approaching-a-multicultural-audience. Accessed 26 May 26 2017

Suryaningrum, D.H.: Knowledge management and performance of small and medium entities in Indonesia. Int. J. Innovat. Manage. Technol. **3**(1), 35–41 (2012)

Wen, L., Rout, T.: Using composition trees to validate an entry profile of software engineering lifecycle profiles for very small entities (VSEs). In: Mas, A., Mesquida, A., Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 38–50. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_4

Yan, H., Li, Y.: Research on the Relationship of Government subsidies and Enterprise performance. In: 3rd International Conference on Management, Education, Information and Control, p. 1806 (2015)

SPI and Assessment

The Maturity of Usability Maturity Models

Carmen L. Carvajal(✉) and Ana M. Moreno

Facultad de Informática, Universidad Politécnica de Madrid,
Campus de Montegancedo S/N Boadilla Del Monte, Madrid, Spain
carmen.carvajal.jimenez@alumnos.upm.es,
ammoreno@fi.upm.es

Abstract. The integration of usability practices in software development is not a straightforward process. In this context, the application of usability maturity models (UMM) in a software organization can provide insightful information to improve such integration. This paper discusses the design and application characteristics of the UMMs used over the last decade. The analysis of recent UMMs confirms that, even when the UMM field is a matter of interest and is getting adapted to new development contexts (for instance, agile or open source). UMMs lack detailed empirical evidence and supporting documentation for their objective application. In addition, our study also identifies other open issues related such as the level of prescriptiveness or mutability of UMMs. Consequently this paper identifies different opportunities for improving the maturity of UMMs. The application of mature UMMs would contribute to a better incorporation of usability and user experience practices in software organizations.

Keywords: Usability maturity models · UX maturity models · Maturity models

1 Introduction

Usability is a quality attribute defined in ISO/IEC 25010 [1] as the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. The benefits of adopting usability or UX (user experience) practices in software development have been much highlighted from the viewpoint of both users and organizations [2, 3]. However, usability methods are hardly ever integrated into software lifecycles in industrial settings [4]. In this context, usability or UX maturity models (UMMs, from now on) have an important role to play in such integration.

According to Becker et al. [5], a maturity model provides the criteria and characteristics that need to be fulfilled in order to achieve a particular maturity level in a specific area. The aim of maturity models in the field of usability is to evaluate the maturity of an organization from a usability point of view. That is, UMMs help to reflect how the usability process and practices are implemented in an organization. As a result of the evaluation, the organization can identify which aspects of usability require improvement. UMMs are thus a very useful tool for an organization to improve its software process from a usability point of view.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 85–99, 2017.
DOI: 10.1007/978-3-319-67383-7_7

There are several UMMs in the literature. Conversely, there is not much research analysing their characteristics and practical applicability. In 2012, Wendler [6] published a systematic mapping study of maturity models applied to different domains. However, he refers only briefly to the specific case of UMMs as part of his discussion of the application of maturity models in the field of software. As far as we know, the only study on UMMs was published in 2006 by Jokela et al. [S1] Jokela et al.'s survey identified 11 pre-2006 UMMs.

A decade later, our aim is to gain an up-to-date snapshot of the state of UMMs in order to identify valid models and their characteristics from both the structural and application viewpoints. To do this, we conduct a systematic mapping study of the UMM literature published over the last ten years; this includes publications about UMMs object of study during the last decade even if they have been originally published before. Practitioners looking to improve the adoption of usability in their development process may find the results useful, as they paint a picture of current UMMs together with their potential strengths and weaknesses. This information is also useful for researchers, as it suggests open lines of research.

The results of this research are reported as follows. Section 2 discusses the research previous to ours. Section 3 explains the applied systematic mapping research method (including the research questions, search process and information extraction process). Section 4 details the results. Finally, Sect. 5 discusses the results and conclusions of this research.

2 Related Work

Wendler [6] published the first ever systematic mapping study in the literature on maturity models in 2012. The study revealed a growing interest in the topic with 237 articles retrieved from 1993 to 2010. This study identified 22 domains in which maturity models have been applied, including knowledge management, information management or IT governance. According to Wendler, software development and software engineering models are the leaders, as there are significantly more articles in this than in other areas. Due to the breadth of the study, however, it mentions UMMs only briefly as an example of one of the subdomains where models are applied in the software field. Wendler highlights weaknesses with respect to model validation and stresses the need to examine the suitability of maturity models in real scenarios.

In the specific area of UMMs, in 2006, Jokela et al. [S1] identified 11 models published up to 2005. Jokela et al. studied these models from the general and practical viewpoints, analysing, for example, the number of levels that they contain or the usability elements that they evaluate. Jokela et al. [S1] concluded that hardly any of the analysed UMMs provide specific guidelines for their practical application or have been empirically validated. Additionally, they suggested a need for a cumulative research tradition on UMMs to help identify problems with existing models, understand the differences between the different models and avoid redundancies.

More recently, Salah et al.[S11] compared 11 MMUs in 2014 aiming to select the appropriated model in order to evaluate the Usability Maturity level in organizations using Agile methodologies and User centered design. In that work, the authors analyzed the models according to their applicability in agile environments. For instance, they required that the models should be light, that is, they do not require considerable amount of time in their evaluations so that they do not interrupt the dynamics of agile methodologies. As a result, although with limitations, the Corporate Usability Maturity, and the Usability Maturity Model-HCS were classified as the most suitable for their application in agile contexts.

3 Research Methodology

Our research follows the systematic mapping procedure designed by Peterson et al. [7]. The research methodology aims to provide an overview of a research area and identify the type and quantity of research conducted and the published results. This section introduces the followed steps for this study.

3.1 Identify Research Questions

As already mentioned, the main aim of this paper is to compile and analyse the studies published from 2006 to 2016 on UMM in order to gain an overview of the field. The research questions for this study are the following:

- RQ1. Which usability maturity models have been addressed by publications over the last 10 years?
- RQ2 What are the general features of the UMMs?
- RQ3 What are the design features of the UMMs?
- RQ4 What are the use features of the UMMs?

The first question (RQ1) identifies the UMMs that have been addressed by publications over the last decade. The other questions aim to delve deeper into these models. Firstly, RQ2 provides an overview of the models by analysing their general features, including their application domain or number of maturity levels used, according to the general analysis presented by Jokela et al. [S1]. RQ3 and RQ4 gather more detailed information on the models from the viewpoint of their structure and application features, respectively. For that aim, particular criteria defined for evaluating maturity models will be used.

3.2 Search Relevant Literature

Based on the above research questions, we defined a set of keywords for searching terms related to usability and usability maturity models. As a result, the search string used was: *(usability OR "human centred design" OR "user centred design" OR "user experience") AND ("usability capability maturity model" OR "usability maturity model" OR "usability maturity" OR UCMM).*

The search was conducted from June to September 2016. It was originally confined to the title and abstract fields of the papers. As the number of returned results was low, however, the search was finally extended to the entire paper. The following electronic databases were used: ScienceDirect, ACM digital library, IEEExplore and Springer

Link. We also queried the Scimago scientific journals ranking in order to make sure that we searched at least the top twenty journals listed under Q1 in the field of Human Computer Interaction – HCI. To do this, we used Google Scholar as a secondary search engine to retrieve information of interest from HCI journals like Cyberpsychology, Behavior, and Social Networking or Topics in Cognitive Science. Finally, we applied the backward snowballing sampling technique on the selected set of papers.

3.3 Select Relevant Papers Based on Inclusion/Exclusion Criteria

We screened the papers returned by the search based on the inclusion and exclusion criteria listed in Table 1. Table 2 summarizes the paper identification process. Initially, the database search using the defined searching-string returned 309 papers. After applying the basic exclusion criteria, the number of papers was reduced to 250. After screening by the title and abstract, 24 papers were left. At this stage of the process, the second author sampled the selected and excluded papers at random to confirm the results. Later, another three papers retrieved by means of the backward snowballing technique were added. The final decision on which papers were selected was taken after reading the full text of the paper. Finally, 17 papers that strictly met the objectives of our research were selected as primary studies (see Appendix A for the full list).

Table 1. Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Papers published since 2006	Papers not written in English
Conference, workshop and journal papers published in the area of software development and HCI	Papers whose full text is not available
Papers describing UMMs	Redundant papers
Papers on the results of applying UMMs	Notes, electronic presentations, poster papers, comments, patents or letters
Papers comparing different UMMs	Papers not focusing on the area of UMMs

Table 2. Paper identification process

Phase	SD	Springer	ACM	IEEE	GS	Total
Search using search string	14	45	22	17	211	309
Exclusion based on basic criteria like language or redundancy	14	43	20	17	156	250
Exclusion based on title and abstract	1	12	2	2	7	24
Inclusion based on backward snowballing	1	12	2	2	10	27
Exclusion based on full text	1	8	0	1	7	17

3.4 Build the Classification Scheme

The articles are organized base on their title, authors, publisher and date. In addition, the articles are classified as primary [8] (i.e., contains the original information), or secondary [8] (i.e., contains information based on a collection of primary studies).

Finally, the MMU title and origin (i.e., Academia or industry) are identified from the article. In addition, the articles are classified as solution, validation, evaluation, experience, philosophy, or opinion according to the criteria presented in [9].

3.5 Extract Information and Map Studies

Figure 1 illustrates a breakdown of the papers according to the classification presented in Sect. 3.4. Nine (9) out of the seventeen (17) papers (S2, S3, S4, S5, S6, S7, S8, S9 and S10) focus on defining new UMMs and are classified as solution studies. They are followed by five evaluation papers (S5, S11, S13, S14 and S17) and another two opinion papers (S15 and S16). Only one validation paper (S5), one experience paper (S12), and one secondary study (S1) have been published. Note that one paper (S5) was classified in three different categories, as it reported evidence related to a solution, validation and evaluation (for this reason, the number of papers listed in Fig. 1 totals 19). Note that the research tool used in the evaluation papers was case studies.

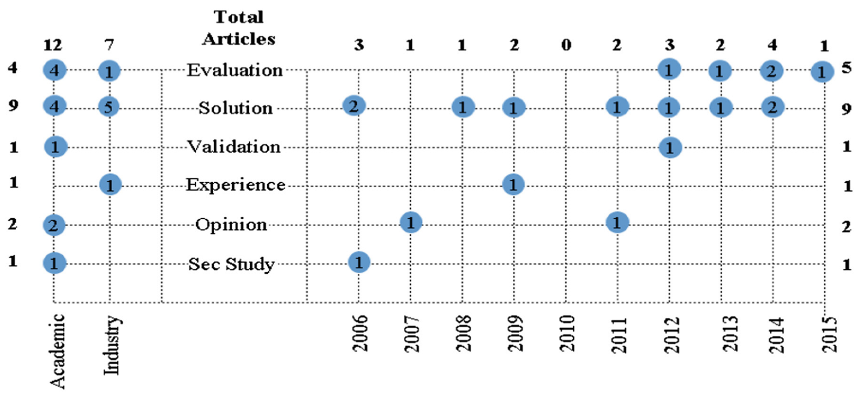

Fig. 1. Map by years and study setting

Additionally, Fig. 1 shows the distribution of the publications by year. This distribution was quite similar from year to year, except in 2010 when no papers were published and 2014 when slightly more papers were published. Figure 1 also shows the distribution of these publications by their setting: academia or industry. There are more papers from academia. This applies to all study types, except for experience papers, where we identified only one primary study conducted in industry, and solution studies, where the five studies from industry illustrated in Fig. 1 refer to four different models.

Although there are more papers from academia —the ratio is about 60 to 40—, we have found that interest in industry is more significant than in other software areas where a much smaller percentage of papers are sourced from industry like example [10].

Finally, the reader could think this study has some threats to validity. The first threat is that only four digital libraries were used (ACM, IEEE, ScienceDirect and Springer Link); however, according to Petersen et al. [17], these are the most relevant libraries for this subject and the use of IEEE and ACM as well as two indexing databases is sufficient for this research. Secondly, the reader could argue that not every study was taken into consideration for this work; however, several strategies were applied in order to mitigate this threat such as the keywords selection and the application of the backward snowballing sampling technique on the selected set of papers. Lastly, in order to counteract any subjective bias on the part of the first author, the final decision on which papers were selected was taken after reading the full text of the paper. Additionally, all the papers were reviewed by the second author separately. The results of the evaluation were compared, and disagreements were settled by negotiation.

4 Research Results

This section presents answers for the research questions stated above.

4.1 RQ1. Maturity Models Under Research Over the Last 10 Years

Table 3 shows eleven models addressed in publications over the last 10 years and their respective references. The model acronyms and the date of their first publication are shown in parentheses. Table 3 illustrates that three of the eleven models identified by Jokela et al. in 2006 have been addressed by publications in the last decade (italicized in Table 3). Although the following questions discuss the features of these models in detail, we should highlight that most of the publications on new UMMs (not italicized in Table 3) are categorized as solution papers that explain the theory underlying the model. An exception is the OS-UMM, which also reports a validation and evaluation of the model. The publications addressing models created before 2006 (italicized in Table 3) are mainly categorized as evaluation papers.

Table 3. UMMs addressed by publications over the last 10 years

UMM	Reference
AgileUX Model (Agile UX, 2014)	S3
UX Maturity Model(UX-MM, 2014)	S4
AUCDI Maturity Model (AUCDI-MM, 2013)	S2
Open Source Usability Maturity Model (OS-UMM, 2011)	S5
Health Usability Maturity Model (HU-MM, 2011)	S6
Corporate UX Maturity Model (CUX,MM, 2009)	S7
KESSU 2.2 (KESSU 2.2, 2007)	S8
Corporate Usability Maturity (CUM, 2006)	S9, S10, S11
ISO 18529 + ISO 15504 (ISO 18529 + 15504, 2000)	S13, S14, S16
Usability Maturity Model - Processes (UMM-P, 1998)	S15
Usability Maturity Model - HCS (UMM-HCS, 1998)	S12, S17

4.2 RQ2. General Features of UMMs

RQ2 aims to provide an overview of UMMs. Therefore, Table 4 summarizes a set of general characteristics of the eleven (11) models identified in our study according to the criteria used in [S1]. None of the models define the time required to achieve maturity, except for CUM. CUM's author, Nielsen, states that it takes 40 years to reach usability

Table 4. General features of UMMs

Model	Time Maturity	Target audience	Domain	Maturity Levels	Dimensions
Agile UX	Undefined	Mgmt and tech	Scrum	Not defined	Six dimensions, e.g., integration of user experience (UX) practices on development and budget for UX
UX-MM	Undefined	Mgmt	Any domain	Stage 1: Beginning to Stage 5: Exceptional	Six indicators, e.g., integration of UX and corporate processes and design thinking to drive UX
AUCDI-MM	Undefined	Mgmt and tech	Agile Org.	Level 0: Not Possible to Level 5: Cont Improvement	Four dimensions, e.g., agile-user centred design (A-UCD) integration process and UCD continuous improvement
OS-UMM	Undefined	Tech	OS Org.	Level 1: Preliminary to Level 5: Institutionalized	Four dimensions, e.g., usability assessment and usability methodology
HU-MM	Undefined	Mgmt	Health Org.	Phase 1: Unrecognized to Phase 5. Strategic	Five elements, e.g., process and infrastructure for usability
CUX-MM	Undefined	Mgmt	Any domain	Level 0: Initial to Level 4: Driven Corporation	Undefined
KESSU 2.2	Undefined	Tech	Any domain	Level 0: Incomplete to Level 5 Optimizing	Seven activities, e.g., identification of user task design and usability feedback
CUM	40 years	Mgmt	Any domain	Stage 1: Hostility to Stage 8: User-Driven Corp.	Five dimensions e.g., management attitude towards usability and strategic usability
ISO 18529 + 15504	Undefined	Mgmt and tech	Any domain	Level 0: Incomplete to Level 5: Optimizing	Seven processes, e.g., plan the HCD process and facilitate the human-system implementation
UMM-P	Undefined	Mgmt and tech	Any domain	Level 0: Incomplete to Level 5: Optimizing	Same dimensions as ISO 18529 + 15504
UMM-HCS	Undefined	Mgmt and tech	Any domain	Level X: Unrecognized to Level E: Institutionalized	Set of practices, e.g. user focus and human centred improvement of organization

maturity [S10]. Such a long time period may not be appealing for a software development organization keen for results. We should note, however, that it, in general, it is not possible to specify accurate times for improvement processes as many organization-dependent factors have a bearing on such processes. These factors include the readiness of the organization, the existence of effective processes and infrastructure to support a programme, and the skills and knowledge of the organization's people [11].

As regards the targeted audience, the results of the evaluation for four models will be mainly useful to management. For example, the UX-MM model focuses on indicators like UX expertise and resources or leadership and culture in the company. On the other hand, two models focus on technology. For instance, KESSU sets out to evaluate the performance of different usability activities conducted by the development team. Finally, the other models combine management and technology issues.

As regards the model application domain, most are generally applicable, that is, can be applied in any type of organization. However, two models are for very specific domains. HU-MM was developed in response to usability problems detected in health-related products. In addition, the OS-UMM model was developed for open source models (OSS). Finally, two of the models were specially designed for organizations enacting an agile development approach (AGILEUX and AUCDI-MM).

Another key feature is the number of levels or stages to achieve maturity in usability. According to Fraser et al. [12], a model usually defines up to six maturity levels. Most of the retrieved models are within this range, except KESSU and CUM with seven and eight stages, respectively. Still, this is not a major deviation. On the other hand, the information reported in the publication that we retrieved about the AGILEUX model is partial, as it only describes level 2 and does not refer to the total number of levels to be considered.

Finally, all the models, except CUX-MM, define areas, dimensions or criteria (depending on the model) that identify key structural elements in the field of usability. They are used to ascertain the usability maturity within an organization. The results of the evaluation of these areas illustrate maturity as a whole and separately for each of the evaluated areas or dimensions. Table 4 shows examples of these dimensions for the different UMMs.

4.3 RQ3. UMM Design Features

The design or structural features of a maturity model are used to describe the form and organization of the model. As already mentioned, we use the design attributes proposed by Mettler et al. [13] for maturity models applied to the information system field. Additionally, the values of some of these criteria were complemented by other research as mentioned below:

- Maturity concept defines the approach of the model:
  - Process maturity, that is, the extent to which a specified process is specifically defined, managed or controlled.
  - Object maturity, that is, the extent to which a particular object, for example, a software product reaches a predefined level of sophistication.

  - Workforce maturity, that is, how proficient a team of people are at building knowledge and improving skills.
- Composition is, according to Fraser et al. [12], divided into three types:
  - Maturity grids usually have a narrative text describing the activities for each maturity level; their design complexity is moderate.
  - Likert-like questionnaires aim to rate specified statements on good practices at different maturity levels.
  - CMM-type models have a more formal architecture and are more complex because a broad spectrum of scales and subscales should be implemented to evaluate maturity.
- Reliability defines two categories:
  - Validated: a model can be validated, qualitatively by means of case studies or using quantitative questionnaires.
  - Verified: a quite accurate conceptual description and specification of the model is given without evidence of its practical use.

If there is no detailed information in this respect, the model is catalogued as "Not fully described".

- Mutability defines two categories:
  - Form refers to whether the model accounts for changes in the description of maturity levels and requirements in order to assure model standardization.
  - Operation refers to changes defined by the model on how maturity is measured at each stage.

Table 5 is a summary of design features of the analyzed UMMs. In this case, all the models are oriented to the usable software construction process.

Table 5. Maturity models design features

Model	Maturity	Composition	Reliability	Mutability
AgileUX	Process	CMM family	Verified	Not mentioned
UX-MM	Process	Maturity grid	Not fully described	Not mentioned
AUCDI-MM	Process	Likert-like	Verified	Not mentioned
OS-UMM	Process	Likert-like	Validated, case studies	Not mentioned
HU-MM	Process	Likert-like	Verified	Not mentioned
CUX-MM	Process	Not mentioned	Not fully described	Not mentioned
KESSU 2.2	Process	Likert-like	Verified	Not mentioned
CUM	Process	Maturity grid	Validated, case studies	Not mentioned
ISO 18529 + 15504	Process	Likert-like	Validated, case studies	Not mentioned
UMM-P	Process	Likert-like	Validated, case studies	Not mentioned
UMM-HCS	Process	Likert-like	Validated, case studies	Not mentioned

As regards composition, six models have a Likert-like composition, where the model authors select the scoring scheme at their discretion. Without a clear description, however, these scoring schemes can be confusing, ambiguous, and lead to mistaken results. In this respect, Salah et al. [S17] claim that the description provided for the UMM-HCS scoring scheme is unsatisfactory. On the other hand, the UX-MM and CUM models have maturity grid composition. Therefore, the result of the evaluation largely depends on how the evaluator interprets the model. AGILEUX is based on the CMM model, but reports only information for maturity level 2. Finally, we were unable to gather enough information from the literature to determine the composition of CUX-MM. Note that the model's composition type is not necessarily a strength or weakness a priori; it depends on what facilities the evaluator is given for applying the respective model.

With respect to the reliability attribute, we found that there is evidence about the use of five out of the eleven models based on case studies. According to Mettler et al.'s terminology, therefore, five models have been validated. There is no empirical evidence for the other six models. Our study did not retrieve any papers containing evidence about the UMM-P model. However, Jokela et al. [S1] pointed out that several case studies were conducted prior to 2006, albeit with contradictory results. On this ground, our study considers this model to have been validated. Note that the fact that model has been validated does not necessarily mean that the results of the validation were successful. On the other hand, Table 5 classifies four models (AgileUX, AUCDI-MM, HU-MM and KESSU 2.2) as verified. These models have an accurate conceptual specification. The conceptual accuracy regarding the UX-MM and CUX-MM models was not found to be good enough in the retrieved literature. On this ground, they have been classified as not fully described.

4.4 RQ4. UMM Use Features

RQ4 is related to the practical application of the model. The attributes identified by Mettler et al. in this respect are complemented with others also provided in literature as follows:

- The method of application defines who applies the model. This can be classified as a self-assessment, or a third-party assisted assessment.
- The type of support to which the model user has access. Three options are given for this attribute: (1) the user is not given any support material; (2) the user is offered a textual description about how to conduct the evaluation; (3) the user is offered a software tool to conduct the evaluation.
- The Purpose of use, defined by De Bruin et al. [14] as:
  - Descriptive: the purpose is to evaluate the current status of the organization.
  - Prescriptive: the purpose, apart from evaluating the organization's current status, is to suggest improvement guidelines in order to progress to the next maturity level. According to Pöppelbuß et al. [17], maturity models claiming to serve a prescriptive purpose of use must provide at least: (1) a set of improvement measurements and recommendations; (2) a decision calculus to help to evaluate different alternatives; and (3) a procedure on how to specify and adapt the improvement measures. In our study, the models that comply with all three characteristics are catalogued as fully prescriptive, whereas models meeting at least one will be classed partially prescriptive. Additionally, according to Mettler

et al., the improvement recommendations may be explicit, that is, detail exactly what to do to improve an activity or process, or implicit that is, they are embedded in other general and non-specific comments.

Table 6 summarizes the characteristics of UMM usage. As regards the method of application, five models were identified as self-assessment models. The authors of OS-UMM and HUMM clearly state that these models are self-assessment ones. AUCDI-MM and KESSU are said to have been designed for a non-specialized audience (that is, evaluators) and should not consume too much time or external resources. Additionally, UMM-HCS [16] was used by Salah et al. as a self-assessment method [S17]. The authors of UMM-P state that their guides were designed for expert personnel like process improvement consultants. Although no mention is made of the application method for the ISO 18529 + ISO 15504 model, it is, according to [S16], a complex model, and the formal use of ISO 18529 is a job for a professional. On this ground, it has also been classified as a third-party assessment model. We were unable to establish the method of application of the other models from the retrieved information.

With regard to the type of model application support to which the user has access, HU-MM is the only model offering a software tool to conduct the evaluation. As shown in Table 6, another five models provide a narrative description of the activities to be evaluated, an explanation of the scoring scheme and a recording form. We did not find any references to possible evaluator support material for the other models.

Finally, as regards the purpose of use, we found that eight of the models are descriptive. On the other hand, three models are classed as partially prescriptive since provide a set of improvement measures. Note, however, that the recommendations are implicit. The HU-MM merely mentions that it will offer some suggestions. As it does not outline these recommendations in the published document, it was catalogued as descriptive.

Table 6. Maturity model use features

Model	Method of Application	Support of Application	Purpose of use
AgileUX	Not mentioned	None offered	Part. prescriptive, implicit
UX-MM	Not mentioned	None offered	Part. prescriptive, implicit
AUCDI-MM	Self-assessment	Textual description	Descriptive
OS-UMM	Self-assessment	None offered	Descriptive
HU-MM	Self-assessment	Software tool	Descriptive
CUX-MM	Not mentioned	None offered	Descriptive
KESSU 2.2	Self-Assessment	Textual description	Descriptive
CUM	Not mentioned	None offered	Part. prescriptive, implicit
ISO 18529 + 15504	Third party assisted	Textual description	Descriptive
UMM-P	Third party assisted	Textual description	Descriptive
UMM-HCS	Self-assessment	Textual description	Descriptive

5 Discussion and Conclusions

In this work, we aim to characterize the UMMs that have been researched in the literature over the last decade. Three of the eleven retrieved UMMs were designed prior to 2006, whereas eight new UMMs have been created since 2006.

Generally, one of the differences detected between the models identified more than one decade ago and the more recent ones is that, new UMMs have been proposed for specific contexts like agile developments or open software over the last decade. This is an interesting development, as it may result in a more efficient evaluation targeting the specific features of such domains.

Several points must be addressed on UMMs. At first, from a practical point of view is their reliability, that is, whether there is evidence about their application in real environments. In this respect, our study is consistent with earlier studies highlighting that the cross-checking of maturity models is insufficient. This study has found information on only five empirically tested models (36%). Model checking was qualitative based on case studies, which is consistent with Wendler's and Jokela et al.'s findings. In our study, ISO 18529 + ISO 15504 is the model for which there is most empirical evidence. However, as discussed in Sect. 4.4, the model is complex, and the assessment has to be made by experts in maturity models. This can, according to [S16], be a major drawback for the practical application of the model. This model is followed by UMM-HCS and CUM, although we identified some deficiencies in CUM scoring or inconsistencies in CUM terminology.

Another important discussion point is the support provided by the models for evaluators. In our study, we have found that five out of the 11 retrieved models (45%) do not offer specific guidance for identifying the usability maturity levels in an organization. The other six models have a narrative description of how to perform this evaluation. Upon evaluation, however some were regarded as hard to interpret. On the other hand, we identified only one software tool supporting evaluation for the HU-MM model. Although there is no guarantee of model application being objective, since this depends on the quality of the material, any support material or even a support tool for evaluation is better than none. Briefly, in this regard, our study pinpointed the same weakness already identified by Jokela et al. [S1] for pre-2006 models, where 46% of the identified models offered no specific guidance to give practitioners insight into how to apply the models.

On the other hand, most models studied serve the purpose of description, that is, output a view of the company's usability status. Our study did not retrieve any fully prescriptive models; however, we did identify three partially prescriptive models with implicit recommendations or improvement practices. Although the studies by Wendler et al. [8] and Jokela et al. [S1] do not refer to the purpose of use, it is useful for establishing which models not only offer information on the organization's usability level but take a step further into practice. Although by no means straightforward, prescriptive models enabling a company to move to the next usability maturity level is an important research issue to cover to promote an effective integration of usability practices into the development process.

Another striking finding is that none of the models refer to the mutability feature, thereby accounting for the possible adaption of the model to new usability practices or process changes. This is also an important feature as software development is a live process and new techniques and practices should emerge as part of continuous improvement. However, we think that mutability would be a desirable feature once the above weaknesses related mainly to model validation, support and improvement recommendations have been resolved. Note that, even if these constraints are overcome, it may not be easy to state that one particular model is better than others, basically because the choice of model is dependent on the features and priorities of each organization. For example, the ISO 18529 + ISO 15504 model has advanced design and use features, but it is a complex model and would not be suitable for application in a small organization.

In summary, as discussed throughout this paper, the field of UMMs cannot yet be considered mature, even though the first UMMs date back over 20 years. Our research aims to contribute to building cumulative research on UMMs as suggested by Jokela et al. in 2006. Although it is not easy to offer practitioners clear recommendations on the best UMMs, the characterization outlined here is a potential decision-making aid. From a research point of view, our characterization is based mainly on criteria already defined in literature for analysing maturity models. Therefore, its application to UMM provides a more robust analysis complementing previous research. On the other hand, we have highlighted open issues and opportunities for research to bring forward the area of UMMs. Mature UMMs will contribute to improve the integration of usability and user experience techniques in the software development process.

Acknowledgments. This work has been developed under the context of project TIN2010-19077 founded by the Spanish Science Foundation.

Appendix A: Mapping Studies

[S1] Jokela, T., et al.: A survey of usability capability maturity models: implications for practice and research. Behaviour & Information Technology. 25(3), 263–282 (2006)

[S2] Salah, D., Paige, R., & Cairns, P.: A Maturity Model for Integrating Agile Processes and User Centred Design. In: International Conference on Software Process Improvement and Capability Determination. Springer International Publishing (2016)

[S3] Peres, A.L., et al.: AGILEUX model: towards a reference model on integrating UX in developing software using agile methodologies. In: 2014 Agile Conference, AGILE 2014, pp. 61–63 (2014)

[S4] Chapman, L., Plewes, S.: A UX Maturity Model: Effective Introduction of UX into Organizations. In: International Conference of Design, User Experience, and Usability, pp. 12–22. Springer International Publishing (2014)

[S5] Raza, A., Capretz, L.-F., Ahmed, F.: An Open Source Usability Maturity Model (OS-UMM). Computers in Human Behavior. 28(4), pp. 1109–1121 (2012)

[S6] HIMSS Usability Task Force.: Promoting usability in Health Organisations: Initial Steps and Progress towards a Healthcare Usability Maturity Model. Health Information and Management Systems Society (2011)

[S7] Van, S.: Corporate User-Experience Maturity Model. In: Human Centered De-sign, pp. 635–639. Springer, Berlin Heidelberg (2009)

[S8] Jokela, T.: Characterizations, requirements, and activities of user-centered de-sign—the KESSU 2.2 model. In: Maturing Usability, pp. 168–196. Springer, London (2008)

[S9] Nielsen, J.: Nielsen's Alertbox: Corporate Usability Maturity: Stages 1–4 (2006)

[S10] Nielsen, J.: Nielsen's Alertbox: Corporate Usability Maturity: Stages 5–8 (2006)

[S11] Salah, D., Paige, R., Cairns, P.: Integrating agile development processes and user centred design- a place for usability maturity models? In: HCSE 2014. LNCS, vol. 8742, pp. 108–125. Springer, Heidelberg (2014) In: Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds.) HCSE 2014. LNCS, vol. 8742, pp. 108–125. (2014)

[S12] Ashley, J., Desmond, K.: Usability maturity: a case study in planning and de-signing an enterprise application suite. In: Human Centered Design, pp. 579–584. Springer, Berlin Heidelberg (2009)

[S13] Bevan, N.: Comparison of Case Studies of Usability Maturity Assessment and Process Improvement. In: CHI 2013 Conference, pp. 54–59 (2013)

[S14] Rodríguez, A.: Extending OpenUP to Conform with the ISO Usability Maturity Model. In: Human-Centered Software Engineering, pp. 90–107. Springer, (2014)

[S15] Joshi, A., Gupta, S.: Usability in India. Global usability, pp. 153–168. Springer (2011)

[S16] Jokela, T., Kantola, N.: Name of the method: ISO 18529 (+ISO 15504). R3UEMs: Review, Report and Refine Usability Evaluation Methods, pp. 106–108 (2007)

[S17] Salah, D., Paige, P., Cairns, P.: Observations on utilizing usability maturity model-human centrdness scale in integrating agile development processes and user-centred design. In: Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2015. Communications in Computer and Information Science, vol. 526, pp. 159–173. Springer, Switzerland (2015)

References

1. ISO, International Software Quality Standard, ISO/IEC 25010. Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) (2011)
2. Hoo, M.H., Jaafar, A.: Usability in practice: perception and practicality of management and practitioners. In: International Conference on Pattern Analysis and Intelligent Robotics (ICPAIR), vol. 2, pp. 211–216 (2011)
3. Nielsen, J., et al.: Return on Investment (ROI) for Usability, 4th edn. Nielsen Norman Group, Fremont (2008)

4. Bornoe, N., Stage, J.: Usability Engineering in the Wild: How Do Practitioners Integrate Usability Engineering in Software Development? In: Sauer, S., Bogdan, C., Forbrig, P., Bernhaupt, R., Winckler, M. (eds.) HCSE 2014. LNCS, vol. 8742, pp. 199–216. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44811-3_12
5. Becker, J., Knackstedt, R., Pöppelbuß, D.: Developing maturity models for IT management. Bus. Inf. Syst. Eng. **1**(3), 213–222 (2009)
6. Wendler, R.: The maturity of maturity model research: a systematic mapping study. Inf. Softw. Technol. **54**(12), 1317–1339 (2012)
7. Petersen, K., et al.: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, vol. 17 (1) (2008)
8. Hox, J.J., Boeije, H.R.: Data collection, primary vs. secondary. In: Encyclopedia of Social Measurement, vol. 1, pp. 593–599 (2005)
9. Wieringa, R., et al.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Eng. **11**, 102–107 (2006)
10. Condori, N., et al.: A systematic mapping study on empirical evaluation of software requirements specifications techniques. In: 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 502–505. IEEE Computer Society (2009)
11. Cmmifaq. http://www.cmmifaq.info
12. Fraser, P., Moultrie, J., Gregory, M.: The use of maturity models/grids as a tool in assessing product development capability. In: Engineering Management Conference, pp. 244–249. IEEE International (2002)
13. Mettler, T., Rohner, P., Winter, R.: Towards a classification of maturity models in information systems. In: Management of the interconnected world, pp. 333–340. Physica-Verlag HD (2010)
14. De Bruin, T., et al.: Understanding the main phases of developing a maturity assessment model. In: 16th Australasian conference on information systems. Sydney (2005)
15. Pöppelbuß, J., Röglinger, M.: What makes a useful maturity model? A framework of general design principles for maturity models and its demonstration in business process management. In: 19th European Conference on Information Systems. Finland (2011)
16. Earthy, J.: Usability Maturity Model: Human Centredness Scale. INUSE Project deliverable D5.1.4(s). Version 1.2., Lloyd's Register of Shipping, London IE2016 INUSE Deliverable D5.1.4s (1998)
17. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. Inf. Soft. Technol. **64**, 1–18 (2015)

Comparative Study of Cybersecurity Capability Maturity Models

Angel Marcelo Rea-Guaman[1], Tomás San Feliu[1], Jose A. Calvo-Manzano[1(✉)], and Isaac Daniel Sanchez-Garcia[2]

[1] Universidad Politécnica de Madrid, ETS Ingenieros Informáticos, Madrid, Spain
marcelo.rea.guaman@alumnos.upm.es, {tomas.sanfeliu,joseantonio.calvomanzano}@upm.es
[2] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Mexico City, Mexico
issanchez@ipn.mx

Abstract. According to ESET, cybersecurity can be defined as the protection of information assets, through the treatment of threats that put at risk the information that is processed, stored and transported by information systems that are interconnected; and a process that involves prevention, detection and reaction or response. This article aims to describe and compare the most used cybersecurity capability maturity models, as a result of a systematic review (SR) of published studies from 2012 to 2017. For this, a taxonomy for comparing cybersecurity capability maturity models was developed, based on Halvorsen and Conradi's taxonomy. Also, the taxonomy is adapted and applied to the cybersecurity capability maturity models identified in the SR. It was observed that the cybersecurity capability maturity models have similar elements because they use processes and levels of maturity, they also manage the risk, although at different levels of depth. Finally, it has been observed that each model due to its particularity has different fields of application.

Keywords: Cybersecurity · Maturity model · Comparative taxonomy

1 Introduction

Cybersecurity is a topical issue and one of the main concerns in organizations, due to the increasing incorporation of technologies in them. In its beginnings, cybersecurity was relatively simple, basically focused on viruses and malicious code. Today, it has become a complex activity; currently, there are persistent attacks on a large scale that allow access to internal corporate networks, generating economic losses, theft of critical information, loss of services, and even reaching the loss of image and prestige of the company [1] and [2].

According to the Oxford dictionary [3], secure is defined as "certain to remain safe and unthreatened". Thus, appropriate protection measures should be implemented according to the importance and criticality of the information. This is the area of Information Security.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 100–113, 2017.
DOI: 10.1007/978-3-319-67383-7_8

Cybersecurity is a widely-used term that talks about the security of systems and data, but it has several definitions. According to ESET security community [4], cybersecurity is defined as "protection of information assets, through the treatment of threats that put at risk the information that is processed, stored and transported by information systems that are interconnected". Therefore, it is a discipline that involves technology, people, information and processes to enable safe operations, which is achieved through the implementation of cybersecurity best practices.

In order organizations can improve their cybersecurity practices, industry and the technical community have developed cybersecurity capability maturity models that allow to measure the cybersecurity capabilities of organizations and position them at different levels. There are different cybersecurity capability maturity models developed by the industry, in many cases developed by state entities with the purpose of being national/international standards. Therefore, organizations have decided to develop maturity models of cybersecurity capabilities that respond to their particular needs. Therefore, it is important to answer the following questions:

- What are the main cybersecurity capability and maturity models used in research studies? This question is answered in [5].
- What are the differences between the main cybersecurity capability and maturity models? This question is the focus of this paper. So, the present work is carried out to identify the main differences, advantages and disadvantages of the maturity models most used in research studies, starting from the systematic review (SR) [5].

Thus, Sect. 2 presents the concept of cybersecurity capability maturity model; Sect. 3 presents the research methodology of the comparative study, and the features for the comparison of the cybersecurity capability maturity models; Sect. 4 shows the description and structure of the cybersecurity capability maturity models identified in the systematic review [5]; Sect. 5 shows the results obtained from the comparison; and Sect. 6 shows the conclusions obtained from the comparative study of the cybersecurity capability maturity models.

2 Cybersecurity Capability Maturity Models

Capability Maturity Model (CMM) broadly refers to a process improvement approach that is based on a process model. CMM also refers specifically to the first such model, developed by the Software Engineering Institute (SEI) in the mid-1980 s, as well as the family of process models that followed. A process model is a structured collection of practices that describe the characteristics of effective processes; the practices included are those proven by experience to be effective [6]. Because the previous definition has been adapted to different fields where the capability maturity models have been used, this concept is also applied to cybersecurity capability models, considering the processes involved in cybersecurity.

Therefore, a cybersecurity capability maturity model provides a benchmark by which an organization can assess the current level of maturity of its practices, processes, and set goals and priorities for improvement in cybersecurity. The cybersecurity capability maturity models are usually structured through the following elements:

- *Areas or Dimensions*: an area groups common concepts of organizational processes, and each area is not necessarily independent of the others.
- *Factors and Indicators:* factors are the objectives that have to be fulfilled in each of the areas of the model, and the indicators serve to visualize the progress towards the objectives.
- *Levels of Maturity:* it is the result of the assessment of the fulfilment of the factors and indicators within the areas or dimensions of the organization. The levels of maturity range from an initial level where an organization may have just begun to consider cybersecurity, to a dynamic comparison where an organization is able to adapt rapidly to changes in the cybersecurity landscape about threats, vulnerabilities, risks, economic strategy or changing organizational needs.

In order to identify the main cybersecurity capability maturity models, the results of the systematic review related to the cybersecurity capability maturity models used in the scientific articles and research carried out up to the present has been compiled [5]. Based on the results of this systematic review, the most relevant cybersecurity capability maturity models were identified, namely: SSE-CMM (Systems Security Engineering Capability Maturity Model) [7], C2M2 (Cybersecurity Capability Maturity Model) [8], CCSMM (Community Cyber Security Maturity Model) [9], and NICE (National Initiative for Cybersecurity Education – Capability Maturity Model) [10].

Also, in the systematic review other cybersecurity capability maturity models were found, but they were not considered in this paper because only the most referenced models were taken into account. Therefore, some of the models that were not considered are: ISM3 (Information Security Management Maturity Model) [11] and COBIT (Control Objectives for Information and related Technology) [12]. ISM3 is a model that manages information security metrics, which help to keep the organization at an acceptable level of risk, although it is adaptable for specifics needs such as cybersecurity; however, its focus is on information security and not on cybersecurity. COBIT is a model that does not fully address the issue of cybersecurity, but focuses on IT governance. Likewise, no models were included that were not used in research studies or did not have a relevant mention.

Emphasize that ISO/IEC 27001 provides the guidelines for establishing an information security management system in a company, however it has not been considered in the results of the systematic review because it does not offer a capability maturity model associated with cybersecurity [13].

In this paper, the particular description of the most referenced cybersecurity capability maturity models that were identified in the previous systematic review and a comparative study are described [5].

3 Methodology Used for Performing the Comparative Study

From the results of the previous systematic review [5], the most relevant cybersecurity models were identified, namely: SSE-CMM, C2M2, CCSMM, and NICE. The methodology to be able to carry out the comparative study of the mentioned models has been based on the taxonomy of software improvement environments proposed by Halvorsen and Conradi [14].

The taxonomy described by Halvorsen and Conradi [14] provides a list of twenty-five relevant features for the comparison of software process improvement frameworks. The features are grouped into five (5) categories. The five categories are described below:

- *General: this category includes the features that describe the general attributes of the improvement environment.*
- *Process*: this category includes the features that describe how the environment is used.
- *Organization:* this category includes the features that describe how is the relationship between the features related to the attributes of the organization and the environment in which it is used.
- *Quality:* this category includes the features related to the quality dimension indicating: aspects for quality measurement, what quality perspective is used, and which means quality in terms of quality indicators.
- *Result:* this category includes the features that describe the results of using the environment, the costs of achieving the results, and the methods used for its validation.

Figure 1 shows the categories and features in the original taxonomy of Halvorsen and Conradi [14], designed to compare process improvement environments. This taxonomy was adapted to be applied in the comparison of the cybersecurity capability maturity models. For the adapted taxonomy, the categories of *quality* and *result* were rejected because they did not allow the comparison of the cybersecurity capability maturity models. Moreover, all the models have in common:

- Handle processes of assessment and improvement, focus on the continuous improvement and provide results that allow the taking of decisions, and
- Have similar features both in quality and in the type of results obtained.
- Have similar processes.

The features of the *General*, *Process* and *Organization* categories were redefined. In the *General* category, the following features were defined:

- *Cybersecurity* oriented: if the maturity model is designed for cybersecurity. This feature is fundamental to consider which models were designed for cybersecurity and which are not.
- *Year of last revision:* last review of the model. This point can provide information about the current evolution of the model due to the constant change that exists in cybersecurity.
- *Organizational environment*: if the cybersecurity capability maturity model is focused on the whole organization or not. This point provides information on whether the model was generated for a specific cybersecurity need or for all organizational environments.
- *NIST framework compatibility:* if the cybersecurity capability maturity model has a security framework associated with it. Based on the systematic review [5], we select the NIST framework because it is the most used with the cybersecurity capability maturity models.

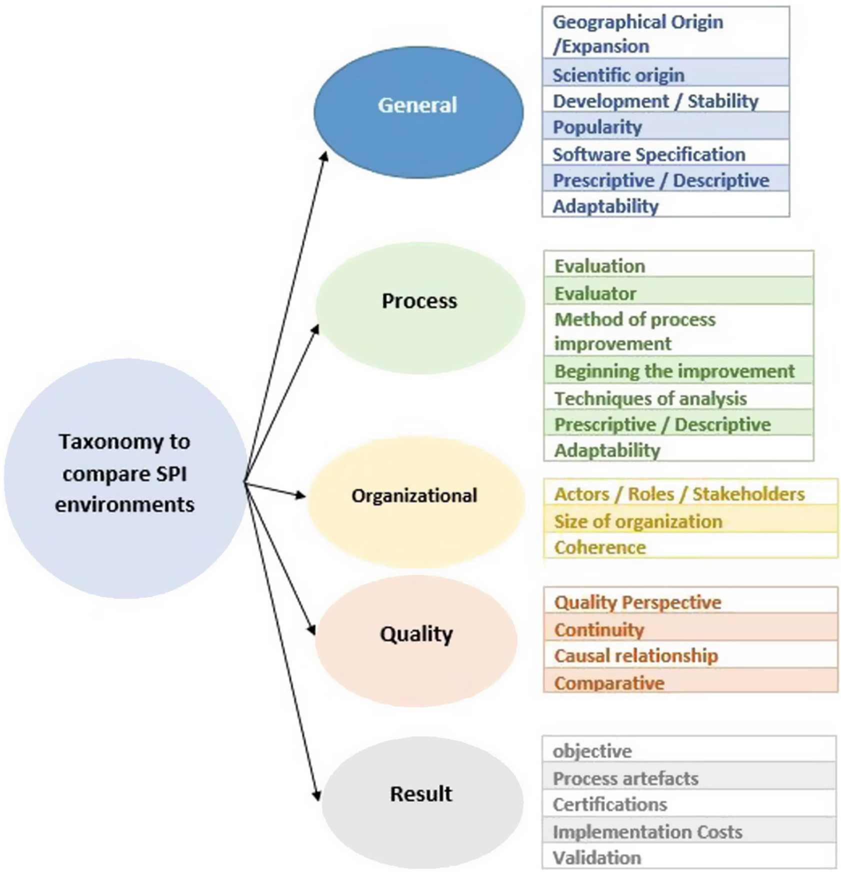

Fig. 1. Taxonomy to compare process improvement environments from Halvorsen and Conradi [14].

- *Measures risk management (threats and vulnerabilities):* if the maturity model measures risk management in a specific or general way.

Related to the Process category, the following features have been taken into account:

- *Application sector:* the area of implementation of the model that is useful to understand the objectives of the model.
- *Depth:* it depends on the complexity of the validation used. This point helps us to differentiate between the models that have a greater detail in the respective levels of maturity and the ones that are simple in this aspect.

Related to the Organizational category, the following features have been taken into account:

- *Definition of roles and responsibilities:* If the cybersecurity capability maturity model has well defined roles and *functions*. This helps us to know which model has a better structure of who should do what.
- *Level of documentation for implementation:* if there is some documentation to support and help the implementation *of* the model. In this way, we can know at what level of detail there is information for the implementation of the model.

Each feature was evaluated as follows:

- *Cybersecurity Oriented.* This feature is evaluated with a "YES" if it is a model oriented towards cybersecurity, and with a "NO" otherwise.
- *Year of last revision.* In this feature, the last year of the review is evaluated, the more recent is the better is.
- *Organizational environment.* If the model is focused on the entire organization, it is evaluated with "YES". In the case that it is focused on a specific area of the organization, it is evaluated with "NO".
- *NIST framework compatibility.* In case the model is compatible with the NIST framework it is evaluated with "YES" and in the opposite case with "NO".
- *Application sector.* This feature is represented by the name of the area(s) to which the model was directed since its creation.
- *Depth.* This feature is evaluated as "GENERAL" if there is only a first level assessment within the maturity levels. It is considered as "SPECIFIC" if the model has a second level of assessment in the maturity levels.
- *Measures risk management (threats and vulnerabilities).* This characteristic is considered as "GENERAL" if a direct risk assessment is not performed. In the case of direct risk assessment is evaluated as "SPECIFIC".
- *Defining roles and responsibilities.* If there are well-defined roles and profiles in the model, it is evaluated with "YES", otherwise with "NO".
- *Level of documentation for implementation.* The document level is considered "HIGH" if you have a white paper, Implementation Guide, and Related Documents. The level of documentation is considered "MEDIUM" if it has a white paper and related documents. The level of documentation is considered "LOW" if it only has introductory documents.

4 Comparison of the Selected Cybersecurity Capability Maturity Models

In this section, the main components and structure of the maturity models that are focused on cybersecurity and those that were adapted are described. The cybersecurity capability maturity models were obtained from the systematic review [5]. According to this review, the cybersecurity capability maturity models most mentioned in scientific articles and research papers are C2M2, SSE-CMM, CCSMM and NICE.

4.1 Cybersecurity Capability Maturity Model (C2M2)

The US Department of Energy, in collaboration with Carnegie Mellon University, published the maturity and capability model in cybersecurity. The last version (1.1) of the model was published in February of 2014.

The model is organized into ten domains, and each domain is a logical grouping of cybersecurity practices. Practices within each domain are organized into objectives, which represent achievements within the domain. The domains and objectives are enumerated in Table 1.

The model defines four maturity levels, from level 0 to level 3, which are applied independently to each model domain. The description of each level is shown in Table 2.

The C2M2 model provides descriptive rather than prescriptive guidance. The content of the model is presented at a high level of abstraction, so that it can be interpreted by organizations of various types, structures, and sizes.

4.2 Systems Security Engineering Capability Maturity Model (SSE-CMM)

Initially, it was sponsored by the US National Security Agency (NSA). The first version of the model was published in October 1996 and the last version (3.0) of the model in June 2003.

The SSE-CMM has two dimensions, "domain" and "capability". The domain dimension consists of all practices that collectively define security engineering, and these practices are called "base practices". The capability dimension represents the practices that indicate management capability and institutionalization of the process, and these practices are called "generic practices". Generic practices represent activities that should be undertaken as part of doing the base practices.

The SSE-CMM contains 129 base practices, organized in 22 process areas. Of these, 61 base practices, organized in 11 process areas, cover all major areas of Security Engineering. The other 68 base practices (organized in the other 11 process areas) related to the Project and Organization are shown in Table 3. Base practices are organized in process areas, and each process area has a set of objectives that represent the expected state of an organization that is performing the process area successfully. An organization that executes the base practices of the process area should also achieve its objectives.

Generic practices are grouped into logical areas called "Common Features", which are organized into five "Maturity Levels", which represent the increased capability of the organization.

Common features are designed to describe major changes in the organization's typical way of performing work processes, and each common feature having one or more generic practices.

The SSE-CMM has five maturity levels, as shown in Table 4. The model described is considered a model not focused on cybersecurity, but it is a model that has been adapted for that purpose due to the lack of models specific to cybersecurity.

Table 1. Domains and objectives of the C2M2 model.

Domains	Objectives
Risk Management	Establish Cybersecurity Risk Management Strategy Manage Cybersecurity Risk Management Activities
Asset, Change, and Configuration Management	Manage Asset Inventory Manage Asset Configuration Manage Changes to Assets Management Activities
Identity and Access Management	Establish and Maintain Identities Control Access Management Activities
Threat and Vulnerability Management	Identify and Respond to Threats Reduce Cybersecurity Vulnerabilities Management Activities
Situational Awareness	Perform Logging Perform Monitoring Establish and Maintain a Common Operating Picture Management Activities
Information Sharing and Communications	Share Cybersecurity Information Management Activities
Event and Incident Response, Continuity of Operations	Detect Cybersecurity Events Escalate Cybersecurity Events and Declare Incidents Respond to Incidents and Escalated Cybersecurity Events Plan for Continuity Management Activities
Supply Chain and External Dependencies Management	Identify Dependencies Manage Dependency Risk Management Activities
Workforce Management	Assign Cybersecurity Responsibilities Control the Workforce Life Cycle Develop Cybersecurity Workforce Increase Cybersecurity Awareness Management Activities
Cybersecurity Program Management	Establish Cybersecurity Program Strategy Sponsor Cybersecurity Program Establish and Maintain Cybersecurity Architecture Perform Secure Software Development Management Activities

Table 2. C2M2 maturity levels.

Maturity Indicator Level (MIL)	Level description
MIL 0	The model contains no practices for MIL0. Performance at MIL0 simply means that MIL1 in a given domain has not been achieved
MIL 1	In each domain, MIL1 contains a set of initial practices. To achieve MIL1, these initial activities may be performed in an ad hoc manner, but they must be performed
MIL 2	The organization's performance of the practices is more stable. At MIL2, the organization can be more confident that the performance of the domain practices will be sustained over time
MIL3	At MIL3, the practices in a domain are further stabilized and are guided by high-level organizational directives, such as policy

Table 3. Process areas of security engineering, and project and organizational.

Security engineering	Project and organizational
PA01 Administer Security Controls	PA12 Ensure Quality
PA02 Assess Impact	PA13 Manage Configuration
PA03 Assess Security Risk	PA14 Manage Project Risk
PA04 Assess Threat	PA15 Monitor and Control Technical Effort
PA05 Assess Vulnerability	PA16 Plan Technical Effort
PA06 Build Assurance Argument	PA17 Define Organization's Systems Engineering Process
PA07 Coordinate Security	PA18 Improve Organization's Systems Engineering Process
PA08 Monitor Security Posture	PA19 Manage Product Line Evolution
PA09 Provide Security Input	PA20 Manage Systems Engineering Support Environment
PA10 Specify Security Needs	PA21 Provide Ongoing Skills and Knowledge
PA11 Verify and Validate Security	PA22 Coordinate with Suppliers

4.3 Community Cyber Security Maturity Model (CCSMM)

Developed by the Center for Infrastructure Assurance and Security (CIAS) of the University of San Antonio, Texas, the Community Cyber Security Maturity Model (CCSMM) is designed to address the needs of states and communities to develop a viable and sustainable program of cybersecurity. The only version (1.0) of the model was published in the year 2006.

The model identifies the characteristics of communities and states as their cybersecurity programs mature. It uses aspects such as knowledge of cybersecurity, security policies and procedures, exchange of information within and between organizations, and cybersecurity training and education.

Table 4. Maturity levels of the SSE-CMM model.

Maturity level	Level description
Level 1, "Performed Informally"	Base practices of the process area are generally performed. The performance of these base practices may not be rigorously planned and tracked
Level 2, "Planned and Tracked"	Performance of the base practices in the process area is planned and tracked. Performance according to specified procedures is verified
Level 3, "Well Defined"	Base practices are performed according to a well-defined process using approved, tailored versions of standard, documented processes
Level 4, "Quantitatively Controlled	Detailed measures of performance are collected and analysed. This leads to a quantitative understanding of process capability and an improved ability to predict performance
Level 5, " Continuously Improving"	Quantitative performance objectives (targets) for process effectiveness and efficiency are established, based on the business goals of the organization

States are made up of communities and communities are made up of organizations, and the model responds to the linkage that exists among the state, the community and the organizations. The model is represented in a three-dimensional way. There are five maturity levels in the CCSMM model for organizations, communities, and states that progress through each of them in the order shown in Table 5. This model performs a high-level assessment because it is focused on states, communities and organizations.

Table 5. Maturity levels of the CCSMM model.

Level of maturity	Level description
Level 1, "Initial"	Organizations, communities and states at this level have little or no cybersecurity awareness, analysis and evaluation
Level 2, "Established"	The leadership of organizations, communities and states at this level is aware of cyber threats, problems and the imperative of adopting cybersecurity. They also recognize the need for cooperative training and education in cyber security
Level 3, "Self-assessed"	At this level, leaders within organizations, communities and states actively promote cybersecurity awareness and cooperate with others in establishing training and education programs
Level 4, "Integrated"	When cyber security is integrated, it is incorporated into every process that an organization, community or state has well-defined programs
Level 5, "Vanguard"	For organizations, communities and states at this level, cyber security is a business imperative. Entities at this level are able to teach others

4.4 National Initiative for Cybersecurity Education – Capability Maturity Model (NICE)

The National Cybersecurity Education Initiative (NICE) evolved from the Integral Cyber Security Initiative (CNCI), Initiative 8 - Expand Cybernetics Education, which was established by the President of the United States George Bush in the Presidential Directive of National Security in January of 2008, to develop a staff with a technological profile in cybersecurity, with the appropriate knowledge and skills. Towards these objectives, the NICE Component 3 focuses on the cybersecurity structure of the staff, specifically in talent management and the role of staff planning. The only version (1.0) of the model was published in August 2014.
The NICE maturity model segments key activities in three main areas:

- *Process and Analytics:* Process represents those activities associated with the actual steps an organization takes to perform workforce planning and how those steps are integrated with other important business processes throughout the organization. Analytics represents those activities associated with supply and demand data and the use of tools, models, and methods to perform workforce planning analysis.
- *Integrated governance:* Represents those activities associated with establishing governance structures, developing and providing guidance, and driving decision-making. It is the building block to an organization's overall workforce planning strategy and vision as well as assignments of responsibility, promotion of integration, and issuing of planning guidance.
- *Trained professionals and enabling technology:* Represents the activities associated with establishing a professional cadre of workforce planners within an organization. Enabling Technology represents the activities related to the accessibility and use of data systems.

The NICE maturity model has three levels of maturity. These levels are shown in Table 6.

Table 6. NICE model maturity levels.

Level of maturity	Level description
Limited level	Limited is the most basic level, portraying an organization with areas of its cybersecurity workforce planning capability in its infancy. This key area of the organization is at the beginning of its development, for example having a limited establishment of processes, lacking clear guidance, without structured data and analysis methods
Progressing level	The progressing level describes some aspects of cybersecurity workforce planning throughout the organization that has started to perform and establish some infrastructure to support efforts
Optimized level	Depicts key areas of workforce planning capabilities in an organization that are fully developed, are integrated with other business processes, and can support different levels of workforce and workload analysis, the results of which drive short- and long-term decision making for the cybersecurity workforce

To use this model, organizations must have an accurate understanding of their current staffing capabilities as they relate to the three areas of the segment and with the ability to show specific evidence of the activities described in the model.

5 Results and Analysis

The features that were defined to evaluate the models were defined previously at the end of Sect. 3. After analysing the cybersecurity capability maturity models obtained from the systematic review [5], a table was made summarizing the comparison among them. Table 7 shows the value of the features for each of the models (C2M2, NICE, CCSMM and SSE-CMM) described in the previous section.

Table 7. Comparison among cybersecurity capability maturity models.

Features	C2M2	NICE	CCSMM	SSE-CMM
General category				
Cybersecurity oriented	YES	YES	YES	NO
Year of last revision	2014	2014	2006	2008
Organizational environment	YES	NO	NO	YES
NIST framework compatibility	YES	NO	YES	NO
Measures risk management (threats and vulnerabilities)	SPECIFIC	GENERAL	GENERAL	SPECIFIC
Process category				
Application sector	ENERGY AND FUELS	WORKFORCE	COMUNITIES	SECURITY ENGINEERING
Depth	SPECIFIC	GENERAL	GENERAL	SPECIFIC
Organization category				
Defining roles and responsibilities	YES	YES	NO	YES
Level of documentation for implementation	MEDIUM	MEDIUM	LOW	HIGH

The comparative study shows that the cybersecurity capability maturity models have a significant similarity. The main difference is identified in the area to which they are focused on and the level of depth of the best practices to be implemented. The main results found in the comparison are the following:

- We searched in the original papers the areas where the models are focused on, if they were conceived for cybersecurity or in another area, and in additional papers, if they were adapted for using in cybersecurity [7–10].
- Models that are more generic (SSE-CMM and C2M2), covering all areas of the organization, cover all the attributes of security (confidentiality, integrity, and availability) [7, 8].
- In the analysed models [7–10], there are models whose evaluation features are very generic (NICE, CCSMM) compared to others (C2M2, SSE-CMM). More specific models provide more information for the proper classification and evaluation of their practices, and provide more detailed guidelines to improve the maturity indicators levels.

In addition to the comparisons described in the present work, we supplemented the systematic review cited [5], using the search string "cybersecurity maturity model" in the "google academic" engine. Only through the application of the freely accessible documents. It was obtained 6,990 documents, and it was identified that the trend of the most mentioned models are the ones compared in this document. Additional models or models that were not used in the systematic review like Resilia, CERT-RMM and SUNY ISI [15–17] were found.

6 Conclusions

All models can be adapted to different types of organizations; however, they need some level of customization. There are models such as C2M2 and CCSMM, which are designed to be implemented in conjunction with the NIST framework.

We have not identified recent updates of the models compared in the present work in the last 3 years.

The only cybersecurity capability maturity model that is focused on cybersecurity, is updated and focused on the entire organization is the C2M2.

The SSE-CMM Model is a model that already has several years in the market but is not focused directly on cybersecurity, although it has been used in this area.

Moreover, it was identified that all cybersecurity capability maturity models need a level of customization to be implemented in an organization.

All cybersecurity capability maturity models are based on cybersecurity risk management, but only SSE-CMM and C2M2 measure risk management in a more specific way.

References

1. Ponemon Institute. http://engage.hpe.com/PDFViewer?ID={81e3f9d9-32fc-43ba-907a1fda52800f8a}Cost_of_Cyber_Crime
2. Donaldson, S., Siegel, S., Williams, C.K., Aslam, A.: Enterprise Cybersecurity: How to Build a Successful Cyberdefense Program Against Advanced Threats. Apress, Berkeley (2015)
3. Oxford University Press. https://en.oxforddictionaries.com/definition/secure

4. welivesecurity. https://www.welivesecurity.com/la-es/2015/06/16/ciberseguridad-seguridad-informacion-diferencia/
5. Rea-Guaman, A.M., Sánchez-García, I.D., San Feliu, T., Calvo-Manzano, J.A.: Maturity models in cybersecurity: a systematic review. In: 12a Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI 2017), Lisbon (2017)
6. Select Business Solutions. http://www.selectbs.com/process-maturity/what-is-the-capability-maturity-model
7. SSE Project Team: System Security Engineering Capability Maturity Model (SSE-CMM): Model Description Document Version 3.0. Technical report, SSE-CMM (2003)
8. Department of Energy.: Cybersecurity Capability Maturity Model (C2M2): Version 1.1. Technical report, Department of Homeland Security (2014)
9. White, G.B.: The community cyber security maturity model. In: IEEE International Conference on Technologies for Homeland Security, pp. 173–178. IEEE Press, Wakefield (2011)
10. US Department of Homeland Security.: Cybersecurity Capability Maturity Model: Version 1.0. White paper, Department of Homeland Security (2014)
11. The Open Group.: Open Information Security Management Maturity Model (O-ISM3). Technical report, Open Group (2011)
12. ISACA (COBIT 5). http://www.isaca.org/COBIT/Pages/COBIT-5-spanish.aspx
13. International Organization for Standarization. https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1
14. Halvorsen, C.P., Conradi, R.: A taxonomy to compare SPI frameworks. In: Ambriola, V. (ed.) EWSPT 2001. LNCS, vol. 2077, pp. 217–235. Springer, Heidelberg (2001). doi:10.1007/3-540-45752-6_17
15. Axelos Global Best Practices. https://www.axelos.com/Corporate/media/Files/Syllabi/RESILIA-Practitioner-2015-Exam-Syllabus-v1.pdf. RESILIA Practitioner Examination Syllabus
16. Matthew, J.B.: Advancing Cybersecurity Capability Measurement Using the CERT ® - RMM Maturity Indicator Level Scale: Version 1.1. Technical report, Carnegie Mellon University (2013)
17. MM Lessing: Best practices show the way to Information Security Maturity. http://researchspace.csir.co.za/dspace/bitstream/handle/10204/3156/Lessing6_2008.pdf?sequence=1&isAllowed=y

The Evolution of the TIPA Framework: Towards the Automation of the Assessment Process in a Design Science Research Project

Béatrix Barafort[1], Anup Shrestha[2(✉)], and Stéphane Cortina[1]

[1] Luxembourg Institute of Science and Technology,
5 Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
{beatrix.barafort,stephane.cortina}@list.lu
[2] Faculty of Business, Education, Law and Arts,
School of Management and Enterprise, University of Southern Queensland,
Toowoomba, QLD 4350, Australia
anup.shrestha@usq.edu.au

Abstract. Managing processes remain a key challenge for most organizations which need to preserve competitiveness. Process assessment frameworks can help by providing instruments guiding process improvement and regulation alignment. Several process assessment frameworks such as TIPA® are based on the ISO Process assessment standard series ISO/IEC 15504, currently revised in the ISO/IEC 330xx family of standards. Following a Design Science Research (DSR) methodology, this paper visits the TIPA Framework evolution throughout iterative cycles in terms of design, rigour and relevance. It investigates how original and new artefacts are being developed and improved over the period of ten years, in particular the new strategic move towards the automation of the assessment process. By demonstrating the evolution of the TIPA framework using a DSR perspective, this paper explicates design knowledge regarding the role and value of the framework within the ISO standards community and in practice.

Keywords: TIPA framework · Automation · Process assessment · TIPA for ITIL application · ISO/IEC 330xx process assessment standards series · Design Science Research

1 Introduction

As all markets and industry sectors are confronted by compliance requirements and innovation challenges, companies operating on such environments are struggling to investigate their unique value proposition in order to gain market share and increase their competitive advantages. Stabilizing and improving organizations and their operational business processes remain a major concern. Managing processes in a way that contribute to the governance and decision making is a key factor for organizations. In order to facilitate governance and management from a process approach perspective, structured frameworks are required for assessing processes. Such frameworks can help determining risks related to processes from significant gaps between the "as-is"

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 114–127, 2017.
DOI: 10.1007/978-3-319-67383-7_9

situation and a targeted “to-be” profile, determining areas for improvement and/or determining gaps in terms of requirements not fulfilled from a regulation perspective.

In the software engineering community, back at the beginning of the nineties, several initiatives were introduced for process assessment: the emergence of the Capability Maturity Model (CMM) [1] originally developed as a tool for objectively assessing the ability of government contractors’ processes to implement a contracted software project, and at the International Standardization Organization (ISO) level where a Study Group [2] established in 1991 reported on the needs and requirements for a software process assessment standard. With many process assessment initiatives emerging at that time, and increasing needs for such measurement instruments on the market, the development of a process assessment standard series started: the ISO/IEC 15504. After a first set of published standards dedicated to software process assessment as Technical Reports, validated throughout Trial Phases [3, 4], a full set of International Standards [5] was developed and published between 2003 and 2006, generalizing the process capability assessment approach to any kind of process, whatever the type and size of organization. Exemplar process assessment models for software (Part 5) and system (Part 6) lifecycles were part of the standard series [5]. Aligned with the ISO standards revision policy, the ISO/IEC 15504 standards series have been reconsidered and revised: the ISO/IEC 330xx family of standards have been developed and started to be published from 2014 [6]. This major revision encompasses harmonization and rigour aspects, generic requirements for building new measurement frameworks and for addressing characteristics other than process capability, along with more guidance and process assessment models in the new domains.

In parallel to the ISO standards for process assessment development, CMM became CMMI (Capability Maturity Model Integrated) in 2002 [1], to address the following areas: Product and service development with CMMI for Development (CMMI-DEV), Service establishment, management with CMMI for Services (CMMI-SVC), and Product and service acquisition with CMMI for Acquisition (CMMI-ACQ).

In the software community, several initiatives have been developed over the years targeting various sectors: Automotive SPICE [7] for software development in the automotive industry, SPICE4SPACE [8] in the space industry, and MDevSPICE [9] in the medical device industry to quote a few ones which are based on the ISO/IEC 15504/330xx process assessment family of standards. From a general organizational perspective, the international Enterprise SPICE [10] initiative also gave birth to a process assessment model that has been published as a Publicly Available Specification (PAS) at ISO [11]. Other examples of non-IT application of the ISO process assessment standards can be cited for innovation, knowledge and technology transfer purposes with innoSPICE [12] and Operational Risk Management [13].

The IT Service Management (ITSM) community is a service oriented IT management framework that advocates best practice processes based on IT Infrastructure Library (ITIL®) to ensure that IT delivers quality service to organizations. In the ITSM community, the Luxembourg Institute of Science and Technology (LIST; public research institute) has developed an ITIL-based process assessment model [14] in the context of a R&D initiative named and branded TIPA® as a framework, with a TIPA for ITIL application [15]. As many other previously mentioned initiatives, it became a widely recognized framework within the ITSM community around the world.

The TIPA Framework is the combined use of a clearly defined process assessment method with a process model. It is documented in a published handbook [16], supported by further guidance (a toolbox), and commercially disseminated to the market through the TIPA training and certification scheme.

The development of the TIPA framework followed a Design Science Research (DSR) approach [17] during the building and evaluation of the assessment artefacts (process models, method, training course, toolbox). We have iteratively applied the three cycle activities of DSR [18] into our TIPA journey of over a decade. With a longstanding history of research and commercial activities, we are now in a position to present our **design cycle** in terms of artefact development and evaluation; backed up with the **rigour cycle** (grounding of the scientific methods and related work) and the **relevance cycle** (alignment with the international standards, industry and best practices).

In the context of incremental scientific innovation (rigour cycle) as well as responding to the market demands for effective and less costly instruments for quality products and services (relevance cycle), this paper investigates the evolution and improvement of the TIPA framework for creating and improving artefacts and supporting TIPA practitioners. After this introduction, Sect. 2 presents a background introduction to the DSR approach; Sect. 3 is associated with the rigour cycle with an explanation of scientific foundations and related works in this area; and Sect. 4 relates to the relevance cycle with key discussions on the state of practices regarding ISO/IEC 330xx requirements and the TIPA framework alignment. Section 5 discusses how the design cycle has enabled the TIPA evolution - the development and ongoing improvements within the TIPA community; then Sect. 6 presents the conclusion with future research and impact of the ongoing TIPA initiatives.

2 Design Science Research

The Design Science Research (DSR) methodology [19] focuses on the development of a new artefact which is particularly suitable for the process assessment discipline being a practice-based research since DSR "…should not only try to understand how the world is, but also how to change it" [20]. A DSR project can follow different guidelines including the use of kernel theories [21], case studies [22] or systematic literature reviews [20]. Moreover, in a socio-technical context the artefact is influenced by the environment in which it operates. Using the extant knowledge, an artefact can be represented as a practical solution so that its contribution to the body of knowledge can be supported. As a result, artefacts with superior utility can be reinvented in an iterative cycle [23]. Along the same lines of thought, Hevner [18] reinforced the need to maintain a balance between academic rigour and industry relevance while representing the artefact as a major outcome of any DSR project.

Our research draws on the DSR methodology for information systems research suggested by Hevner [18]. The DSR methodology, which combines both behavioral and design science paradigms, comprises three interlinked research cycles: relevance, rigour and the central design cycle [18]. The **relevance cycle** inputs requirements (*capability determination and process improvement*) from the relevant process assessment standards and the concerned industries (such as ITSM, Risk Management, Information Security Management) into the research and introduces the research artefacts (*collectively referred as the TIPA framework*) into real-world application. The **rigour cycle** develops the methods (*assessment frameworks and methods*) along with resources and expertise from the body of knowledge (*ISO/IEC 330xx standards series and research team expertise*) for the research. The **design cycle** supports the loop of research activities that provides the development, evaluation and improvement of the research artefacts. During the research journey of TIPA evolution, we also used DSR insights from Peffers et al. [24] for additional guidance. The three research cycles that demonstrates the evolution of the TIPA framework are discussed next.

3 Rigour Cycle: Scientific Foundations and Related Works

The DSR background associated with our TIPA evolution were explained in Sect. 2. The TIPA framework development following the DSR method comprises a set of artefacts that contribute to and support process assessment. A process assessment framework can be composed of process models, process assessment method, training courses, certification scheme for assessors and lead assessors, and a software tool for supporting the method as potentially valuable artefacts.

The TIPA Framework's set of artefacts followed the DSR rigour cycle, and is strictly aligned on ISO/IEC 15504-330xx requirements, and on guidance for implementing theories in a way that is adapted to practitioners. For the Process Models development part, Goal-Oriented Requirements Engineering (GORE) techniques [25, 26] have been applied in order to obtain the TIPA for ITIL Process Assessment Model. The model has been validated throughout various improvement loops with mechanisms including ITIL and ISO/IEC 15504/330xx expert reviews, experimentations with early adopters and real life process assessment projects. The TIPA Process assessment method published in the TIPA handbook and supported by a toolbox, along with the TIPA for ITIL Assessor and Lead Assessor training courses have also been developed in a rigorous cycle with feedback collected from early adopters and real process assessment projects accumulated over the period of ten years.

With a view to discuss related work, besides TIPA, several process assessment frameworks and tools that are based on ISO/IEC 15504-330xx requirements and guidance, in both IT and non-IT application domains are explained next.

One of the first process assessment frameworks were SPICE for SPACE (S4S) and Automotive SPICE. S4S was developed in the year 2000 and supported the Space industry in Europe for enabling the European Space Industry to select suppliers mastering their processes up to a certain targeted capability level. The S4S Process Model was based on the ISO/IEC 15504-5 Exemplar process assessment model for software lifecycle processes, with specific adaptations and processes dedicated to the Space industry needs [8].

Automotive SPICE was developed throughout the support of car industry stakeholders [27, 28]. The Automotive SPICE process assessment and process reference models have initially been developed under the Automotive SPICE initiative by consensus of the car manufacturers within the Automotive Special Interest Group (SIG), a joint special interest group of Automotive Original Equipment Manufacturers, the Procurement Forum and the SPICE User Group. It has been revised by a Working Group of the Quality Management Center (QMC) in the German Association of the Automotive Industry with the representation of members of the Automotive Special Interest Group, and with the agreement of the SPICE User Group [29]. Besides the Automotive SPICE Process Model, a certification scheme has been developed with training courses which enabled to train a community of SPICE competent assessors and lead assessors under the authority of the INTACS association for certifying assessors. Some consultants have automated the assessment process with software tools. The community of interests of both S4S and Automotive SPICE contributed to the validation and improvement of various artefacts within their respective framework.

More recently, the MDevSPICE initiative has been developed: it aims at proposing a Process Assessment Framework for the Medical Device community, aligned with many regulations of the sector. A set of artefacts compounds the framework [9]. A Brazilian initiative developed by researchers and applied in the software engineering market in Brazil is also proposing a framework, with a process model, a method, a supporting tool and competence development support related to process assessment [30, 31].

Many Process Assessment Frameworks are targeting software engineering processes because it was the initial community of interest of the ISO standard. But the generic nature of the Process assessment and measurement framework principles enable their application to any kind of industry and as a consequence, several other applications have emerged. From a general enterprise perspective, the Enterprise SPICE initiative has proposed a Process Assessment Model with a consortium which participated in the development and validation of the model [10]. This model has been introduced, positively voted and then published in ISO as a Publicly Available Specification [11].

In the IT Service Management community, an Australian Public-Private Partnership has enabled the development of a software-mediated process assessment approach for IT service management processes, which is based on the ISO/IEC 15504-8 exemplar process assessment model and ISO/IEC TR 20000-4 process reference model for IT service management and using ITIL for process improvement [32]. It provides sound insights both from a scientific background and practitioner's point of view, as it proposes an automated framework, which is similar to TIPA Framework evolution towards the development of an automated tool, as later discussed in Sect. 5.

4 Relevance Cycle: Process Assessment Standard and TIPA Framework Alignment

The relevance cycle of the DSR method is demonstrated with a detailed account of our involvement with the ISO community during the development and revisions of the process assessment standard and how we aligned our TIPA framework with the standard using the experience within the ISO community as well as in industry. Our involvement with the ISO community enables us to have access to working drafts and to anticipate changes until their publication.

The ISO/IEC 15504 standards series [5] has been revised and is progressively replaced [33] by the ISO/IEC 330xx family of standards [34]. The generic features of the process assessment mechanisms are emphasized in order to enable, inter alia,

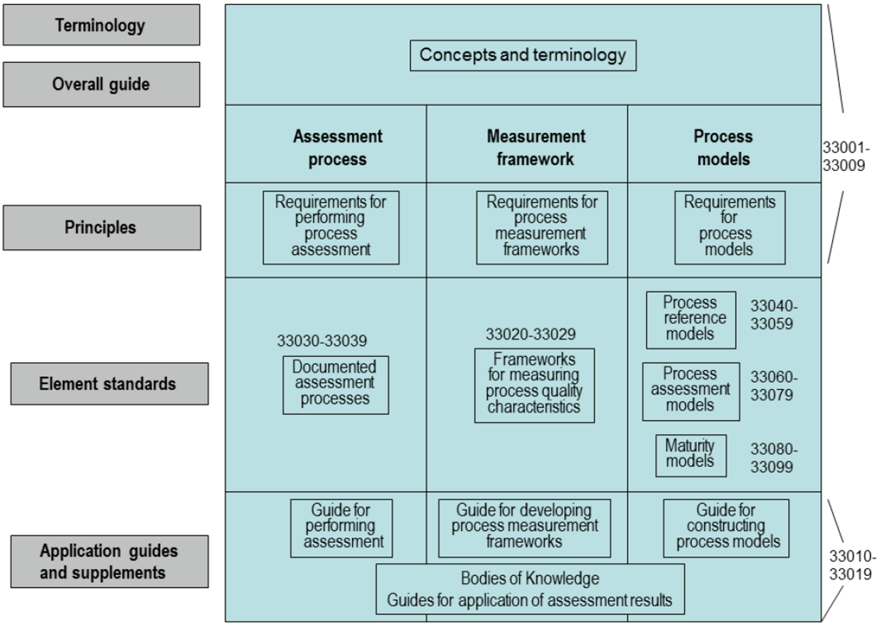

Fig. 1. Structure of the set of standards for process assessment (source: ISO/IEC 33001)

the definition of new measurement frameworks, and to target quality characteristics other than capability. Figure 1 depicts the global structure of the ISO/IEC 330xx family of standards.

The correspondence between the ISO/IEC 15504 series and the ISO/IEC 330xx is summarized in Table 1, by citing the main documents which are of direct interest for the TIPA Framework.

With the revision of the ISO/IEC 15504 series resulting in the new ISO/IEC 330xx series, several changes and adaptations were undertaken. We have systematically compared the ISO/IEC 15504 and ISO/IEC 330xx corresponding parts, and checked how they are currently reflected in the TIPA Framework, making necessary changes as we revise the updates. As several team members of the TIPA framework are actively involved with the relevant ISO standards development and revision, most changes are organic and streamlined. Here is an overview of the mapping between the TIPA for ITIL artefacts and the corresponding ISO standard documents. The revisions were taken into account by aligning with the standards when published by ISO and/or anticipating the revised published documents where possible while working with the ISO community.

By demonstrating the matching of ISO/IEC 15504 and ISO/IEC 330xx standards (Table 1) and then the mapping of ISO/IEC 330xx with TIPA for ITIL artefacts (Table 2), we explained the relevance cycle as these activities identify the needs and requirements (i.e. opportunities and problems) as inputs to our current DSR work towards the evolution of the TIPA framework.

Table 1. Correspondence table between ISO/IEC 15504 and ISO/IEC 330xx

ISO/IEC 15504 document	Replaced by corresponding document(s) in the ISO/IEC 330xx family
Part 1 - Concepts and vocabulary (2004)	33001 (2015) – Concepts and terminology
Part 2 – Performing an assessment (2003)	33002 (2015) - Requirements for performing process assessment
	33003 (2015) - Requirements for process measurement frameworks
	33004 (2015) - Requirements for process reference, process assessment and maturity models
	33020 (2015) - Process measurement framework for assessment of process capability
Part 3 - Guidance on performing an assessment	33010 (working draft) – Guide to performing assessment
	33030 (2017) - An exemplar documented assessment process
Part 4 - Guidance on use for process improvement and process capability determination (2004)	33014 (2013) - Guide for process improvement
Part 5 - An exemplar Process Assessment Model (software lifecycle processes) (2012)	33061 (working draft; waiting progress from ISO/IEC 12207 (2008) Software lifecycle processes) - Process capability assessment model for software lifecycle processes
Part 7 - Assessment of organizational maturity	Partially replaced by ISO/IEC 33002 and ISO/IEC 33003 (see above)
Part 8 - An exemplar process assessment model for IT service management (2012)	33062 - Process capability assessment model for IT service management (waiting for progress on ISO/IEC 20000-4 PRM for IT service management)

5 Design Cycle: Evolution of the TIPA Framework

While reporting the TIPA evolution, it is important to highlight that the iterative nature of the design cycle ensured that the TIPA framework built after several "build-evaluate" cycles has utility and validity.

In order to exemplify the design cycle of the DSR method, the TIPA Framework is based on ISO/IEC 15504-330xx standards series in terms of the requirements and guidance. The ISO/IEC 15504-330xx standards requirements are grounded in Quality Management theories for structuring the capability and maturity scale and on Measurements theories for the assessment of practices [2]. The TIPA framework from the latest design cycle is represented in Fig. 2, with a focus on the TIPA for ITIL application in the domain of ITSM. The reason to highlight TIPA for ITIL application is due to the long-standing history and commercial success of this application during our TIPA journey.

Table 2. Mapping between the TIPA for ITIL artefacts and the corresponding ISO/IEC 15504/330xx standard documents

TIPA for ITIL component	ISO/IEC 15504 Part	ISO/IEC 330xx document	Comment
TIPA for ITIL Process assessment model	Part 2 clauses for describing PRM and PAM, and Part 5 for the Measure-ment Frame-work dimension	33004	There is not yet a dedicated 330xx guide for constructing process reference, process assessment and maturity models; the TIPA Transformation Process and GORE techniques for designing PRMs and PAMs provided a reliable, structured and systematic approach for quality results.
TIPA Method (described in a published handbook)	Part 2 and Part 4	33002, 33004, 33014	Classes of Assessment were added and described in a published whitepaper and factsheets (a new version of the handbook has not been re-published yet); a significant part of the TIPA Method is embedded in a SaaS tool, currently being developed.
TIPA for ITIL Office Toolbox	Parts 2, 3, 4	33002, 33004, 33014	The TIPA for ITIL Toolbox had been upgraded with classes of assessment; the current major upgrade is the provision of the SaaS Tool replacing the Office Toolbox.
TIPA for ITIL Training Course	Parts 2, 3, 4	33001, 33002, 33004, 33014	Classes of Assessment were added with all impacted tools of the Office Toolbox; the SaaS Tool will support assessment training in near future.
TIPA for ITIL Assessor and Lead Assessor Certification scheme	Part 3	33002	TIPA is making a clear distinction between the skills required by the Lead Assessor (the one accountable for the assessment results), and those required by the Assessors.

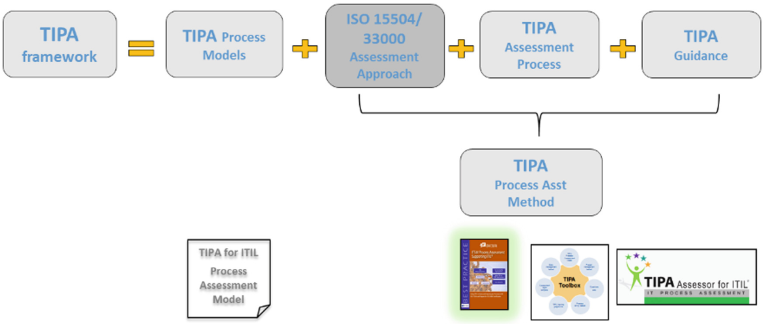

Fig. 2. The TIPA Framework components with TIPA for ITIL artefacts

The TIPA Framework has been used by trained TIPA assessors and Lead assessors over the years. Originally, guidance for supporting assessment projects has been provided via static documents (typically using the Microsoft Office files – Word, Excel and PowerPoint). While the structure and guidance provided by this solution was effective, it was not comprehensive, predominantly due to the lack of maintainability and security of the files. To address these weaknesses, we worked on a major evolution of the TIPA Framework to develop a cloud-based software-as-a-service (SaaS) tool in order to automate and support the assessment process as well as for the storage of assessment data for benchmarking and trend analysis.

The SaaS Tool is designed to enable cost-effective and repeatable process assessments. Therefore, the time and resource requirements to organize process assessments could be shortened. The SaaS Tool has the potential to automate key process assessment activities including assessment data collection, analysis and reporting. For SaaS tool of the TIPA framework, we followed the DSR approach using a set of six activities described by Peffers [24], viz.: (1) problem identification and motivation, (2) Define the objectives for a solution, (3) Design and development, (4) Demonstration, (5) Evaluation, and (6) Communication. Currently, the first three activities have been completed and we are in activity 4 Demonstration stage. Activity 5 Evaluation is carefully planned and works are being done as part of Activity 6 Communication including this paper. Further discussion of this latest round of TIPA evolution in terms of DSR mindset and actual project activities is discussed next using the guidance from Peffers et al. [24].

5.1 Problem Identification and Motivation

DSR mindset: This activity aims at defining the specific research problem and justifying the value of a solution. The problem definition will be used to develop an artefact that can provide a solution. In order to motivate the value of a solution, this set of activities includes knowledge of the state of the problem and the importance of its solution.

Project Activity: All relevant business and market constraints were investigated during this stage. This was done via interviews of business practitioners (mainly LIST assessors and Lead assessors, and TIPA certified TIPA assessors and Lead assessors) and by benchmarking existing similar tools on the market. The outcome was a cartograph that we mapped from the TIPA business activities in order to represent business problems that highlight the importance of the solution (tool).

5.2 Define the Objectives for a Solution

DSR mindset: This activity aims at inferring the objectives of a solution from the problem definition and knowledge of what is possible and feasible.

Project Activity: In order to define the objectives for the future solution, a methodological approach based on User Experience (UX) principles was followed. An experience map was produced with a service design mind set in order to determine "pain" points demonstrating what brings value – by answering what are the positive and negative points and why? Personas were created to assist in solution design. A Persona, in user-centered design and marketing, is a fictional character created to represent a user type that might use a site, brand, or product in a similar way. In our case, personas were used as part of the user-centered design process to design the software (we referred them as "ProtoPersonas" which are an adaptation of the real world users). The personas enabled us to create archetypes of users, with a focus on the users who are bringing the most value to the product (from a financial value perspective).

A questionnaire was sent to the current users of the TIPA toolbox in order to validate Personas, and to prioritize the usage of the twenty tools of the initial toolbox (it is important to quote that some tools are compulsory because they rely on ISO/IEC 15504-330xx requirements but others are "nice to haves").

5.3 Design and Development

DSR mindset: This activity aims at creating the artefact(s). A design research artefact can be any designed object in which a research contribution is embedded in the design.

Project Activity: A hierarchy of the information requirements was developed before prototyping the software application with wireframes (mock-ups). With the user at the centre of the design and development, the Minimum Viable Product (MVP) was defined with the development of the main systems functionalities. Essential components were delivered, and more functionalities were progressively added. A SCRUM development method was applied, with twice-monthly Sprints during tool development.

5.4 Demonstration (Ongoing)

DSR mindset: This activity aims at demonstrating the use of the artefact to solve one or more instances of the problem. This can be done via the experimentation of the artefact's use.

Project Activity: After a first real-life experimentation which enabled us to provide a first level validation and refinements of the tool, an Alpha version of the tool is currently being experimented by early adopters. These volunteer partners were interested in demonstration so as to become more competitive on the market. The partners are committed to provide us feedback on the time saved during each phase of their assessment projects (scope definition, data collection, analysis and particularly reporting) as well as on their perceptions regarding user experience with the tool. A Beta version and more experimentations are expected by the end of this year 2017.

5.5 Evaluation (Planned)

DSR mindset: This activity aims at observing and measuring how well the artefact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artefact in the demonstration. It requires knowledge of relevant metrics and analysis techniques.

Project Activity: Using a survey approach, feedback will be sought from early adopters of the tool in terms of its effectiveness and usability. A systematic analysis of the collected feedback information will allow us to act accordingly in order to improve the SaaS Tool. Ongoing evaluation rounds are planned in order to collect feedback not only from the Alpha Version, but also from the next Beta one, and the definitive product once delivered to the market.

5.6 Communication

DSR mindset: This activity aims at communicating the problem and its importance, the artefact, its utility and novelty, the rigour of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate.

Project Activity: This paper is part of the communication activity to disseminate information about the tool. Social networks and TIPA training courses provisioning are our mainstream mediums to spread the news related to the new artefact supporting the TIPA method. Our TIPA website plays an important role in our communication plan. The ISO standardization community will also be part of the communication channel to demonstrate effective use and commercialisation of the process assessment standard.

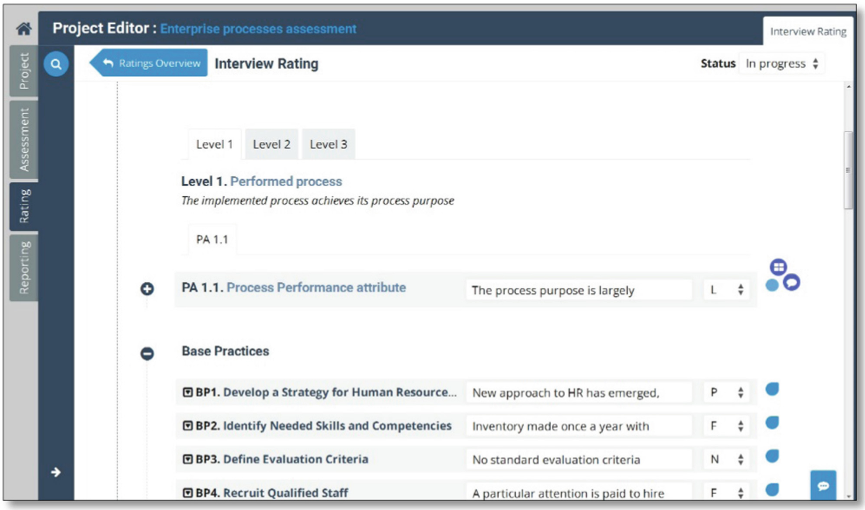

Fig. 3. Example of a screen of the TIPA Framework SaaS Tool

We believe that the SaaS tool plays an important part as part of the evolution of the TIPA Framework with the TIPA for ITIL application as our most important evaluation case so far. Figure 3 illustrates one of the most critical and useful screens for rating instances of an assessed process.

6 Conclusion

The TIPA Framework is flexible in the sense that it can support any business according to the selected process models. This TIPA Factory mindset stresses the generic mechanisms of process assessment with a measurement framework. With the TIPA for ITIL instantiation of the TIPA Framework, a continuous improvement loop is in place, with mechanisms for gathering feedback from the TIPA for ITIL community (more than 260 TIPA Assessors and Lead Assessors have been trained worldwide). Feedback on the adoption of the TIPA framework come from diverse sources, including the training courses and social media networks (LinkedIn, Facebook and the like). The added value of the toolbox has been emphasized with the development of SaaS tool. In order to better support TIPA adoption and to deploy the TIPA framework more broadly, we believe that the SaaS tool will play an indispensable role in the TIPA journey. Moreover, the SaaS tool is expected to simplify and optimize the assessor and lead assessor performance along with the storage of structured data on process assessments.

DSR has been known to generate field-tested and theoretically grounded design knowledge while developing artefacts. The DSR methodology proposes that the output of DSR activities should provide practical design knowledge. Therefore, the artefacts developed during our research work towards the TIPA evolution have adhered to the DSR cycles demonstrating the rigour, relevance and iterative design stages. As we understand while generating novel artefacts, evidence of utility of the artefact assures researchers that the contributions of the artefact are applicable. It is a well-accepted notion that an integrated approach that uses both design science and behavioral science paradigms can highlight more useful artefacts in the scientific community. Since our research only highlights design science paradigm to demonstrate our artefacts, we have planned and call upon future research for review and evaluation of our artefacts using behavioral science paradigm so that more robust contributions can be made to the scientific body of knowledge. We believe that reporting our TIPA evolution within the parameters of the DSR methodology has allowed us to explain how our TIPA artefacts represent valid contribution to the body of knowledge for further testing and analysis. We expect this will enable practitioners and other researchers to access trustworthy and authentic design knowledge in the discipline of process assessment.

References

1. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model (CMM) for Software, Version 1.1 Technical report. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA (1993)
2. ISO/IEC: JTC1/SC7 SC7N944R Report of the Study Group on the Needs and Requirements for a Standard for Software Process Assessment (1992)
3. SPICE Trials: Phase 2 SPICE Trials Final Report, vol. 1, ISO/IEC JTC1/SC7/WG10 (1991)
4. Jung, H.W., Hunter, R., Goldenson, D.R., El-Emam, K.: Findings from Phase 2 of the SPICE trials. Softw. Process. Improv. Pract. **6**(4), 205–242 (2001)
5. ISO/IEC: ISO/IEC 15504 Information technology — Process assessment, Parts 1-10 (2003, 2012)
6. ISO/IEC: ISO/IEC 33001 Information Technology — Process assessment — Concepts and terminology (2015)
7. Automotive Spice®. http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf. Last accessed 1 June 2017
8. Cass, A., Völcker, C., Ouared, R., Dorling, A., Winzer, L., Carranza, J.M.: SPICE for SPACE trials, risk analysis, and process improvement. Softw. Process. Improv. Pract. **9**(1), 13–21 (2004)
9. Lepmets, M., McCaffery, F., Clarke, P.: Development and benefits of MDevSPICE®, the medical device software process assessment framework. J. Softw. Evol. Process. **28**(9), 800–816 (2016)
10. Enterprise SPICE. http://enterprisespice.com/page/publication-1. Last accessed 1 June 2017
11. ISO/IEC: ISO/IEC 33071 Information Technology — Process assessment — An integrated process capability assessment model for Enterprise processes (2015)
12. innoSPICE. http://innospice.ning.com/. Last accessed 1 June 2017
13. Di Renzo, B., et al.: Operational risk management in financial institutions: process assessment in concordance with basel II. Softw. Process. Improv. Pract. **12**(4), 321–330 (2007)
14. TIPA® for ITIL® 2011 r2 v4.1. https://www.list.lu/fileadmin//files/projects/TIPA_T10_ITIL_PAM_r2_v4.1.pdf. Last accessed 1 June 2017
15. Renault, A., Barafort, B.: TIPA for ITIL – from genesis to maturity of SPICE applied to ITIL 2011. In: Proceedings of the International Conference EuroSPI 2014, Luxembourg (2014)
16. Barafort, B., et al.: ITSM Process Assessment Supporting ITIL: Using TIPA to Assess and Improve Your Processes with ISO 15504 and Prepare for ISO 20000 Certification, vol. 217. Van Haren, Zaltbommel (2009)
17. Barafort, B., Rousseau, A., Dubois, E.: How to design an innovative framework for process improvement? The TIPA for ITIL case. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 48–59. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43896-1_5
18. Hevner, A.R.: A three cycle view of design science research. Scand. J. Inf. Syst. **19**(2), 87–92 (2007)
19. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
20. Carlsson, S.A., Henningsson, S., Hrastinski, S., Keller, C.: Socio-technical IS design science research: developing design theory for IS integration management. ISeBM **9**(1), 109–131 (2011)

21. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing information system design theory in perspective: how useful was our 1992 initial rendition? J. Inf. Technol. Theor. Appl. **6**(2), 43 (2004)
22. Van Aken, J.E.: The Nature of Organizing Design: Both Like and Unlike Material Object Design. Eindhoven Center for Innovation Studies, 06-13 (2006)
23. Venable, J., Pries-Heje, J., Baskerville, R.: A comprehensive framework for evaluation in design science research. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 423–438. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29863-9_31
24. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. **24**(3), 45–77 (2007)
25. Rifaut, A., Dubois, E.: Using Goal-Oriented Requirements Engineering for Improving the Quality of ISO/IEC 15504 based Compliance Assessment Frameworks. In: 16th IEEE RE 2008, Barcelona, Catalunya, Spain, pp. 33–42 (2008)
26. Barafort, B., Renault, A., Picard, M., Cortina, S.: A transformation process for building PRMs and PAMs based on a collection of requirements – example with ISO/IEC 20000. In: Proceedings of the International Conference SPICE 2008, Nuremberg, Germany (2008)
27. Mueller, M., Hoermann, K., Dittmann, L., Zimmer, J.: Automotive SPICE in Practice: Surviving Implementation and Assessment. O'Reilly, New York (2008)
28. Fabbrini, F., Fusani, M., Lami, G., Sivera, E.: A SPICE-based software supplier qualification mechanism in automotive industry. Softw. Process. Improv. Pract. **12**(6), 523–528 (2007)
29. The SPICE User Group. http://spiceforum.ning.com/. Last accessed 1 June 2017
30. Alves, A.M., Salviano, C.F., Stefanuto, G.N., Maintinguer, S.T., Mattos, C.V., Zeitoum, C., Martinez, M., Reuss, G.: CERTICS assessment methodology for software technological development and innovation. In: 9th International Conference QUATIC 2014. IEEE (2014)
31. Silva, D.C., Raldi, A., Messias, T., Alves, A.M., Salviano, C.F.: A process driven software platform to full support process assessment method. In: 40th EuroMicro Conference on SEAA 2014. IEEE (2014)
32. Shrestha, A., Cater-Steel, A., Tan, W.-G., Toleman, M.: Software-mediated process assessment for IT service capability management. In: 22nd European Conference on Information Systems, ECIS 2014, Tel Aviv, Israel (2014)
33. ISO/IEC: JTC1/SC7 WG10 N1099 Transition from ISO/IEC 15504 to ISO/IEC 330xx (2017)
34. ISO/IEC: ISO/IEC 330xx Information Technology - Process Assessment (2013, 2017)

Development of an Assessment Model for Industry 4.0: Industry 4.0-MM

Ebru Gökalp(✉), Umut Şener, and P. Erhan Eren

Informatics Institute, Middle East Technical University, Ankara, Turkey
{egokalp, sumut, ereren}@metu.edu.tr

Abstract. The application of new technologies in the manufacturing environment is ushering a new era referred to as the 4th industrial revolution, and this digital transformation appeals to companies due to various competitive advantages it provides. Accordingly, there is a fundamental need for assisting companies in the transition to Industry 4.0 technologies/practices, and guiding them for improving their capabilities in a standardized, objective, and repeatable way. Maturity Models (MMs) aim to assist organizations by providing comprehensive guidance. Therefore, the literature is reviewed systematically with the aim of identifying existing studies related to MMs proposed in the context of Industry 4.0. Seven identified MMs are analyzed by comparing their characteristics of scope, purpose, completeness, clearness, and objectivity. It is concluded that none of them satisfies all expected criteria. In order to satisfy the need for a structured Industry 4.0 assessment/maturity model, SPICE-based Industry 4.0-MM is proposed in this study. Industry 4.0-MM has a holistic approach consisting of the assessment of process transformation, application management, data governance, asset management, and organizational alignment areas. The aim is to create a common base for performing an assessment of the establishment of Industry 4.0 technologies, and to guide companies towards achieving a higher maturity stage in order to maximize the economic benefits of Industry 4.0. Hence, Industry 4.0-MM provides standardization in continuous benchmarking and improvement of businesses in the manufacturing industry.

Keywords: Industry 4.0 · Industrial Internet of Things · Maturity Model · Assessment model

1 Introduction

While referring to Industry 4.0, German Federal Ministry of Education and Research mentions that "the flexibility that exists in value-creating networks is increased by the application of cyber-physical production systems (CPPS). This enables machines and plants to adapt their behavior to changing orders and operating conditions through self-optimization and reconfiguration" [1]. It is further defined as a successful transition from on-premise production systems and processes to "Smart Production", "Smart Manufacturing", "Integrated Industry", "Connected Industry" or "Industrial Internet", which covers distributed and interconnected manufacturing equipment, and requires intelligent systems, a proper engineering practice and related tools [1].

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 128–142, 2017.
DOI: 10.1007/978-3-319-67383-7_10

Industry 4.0 is expected to have a significant impact on supply chains, business models and processes [1]. As stated in [2], fundamental changes, which are likely to be enabled by the transformation to Industry 4.0, are as follows: Meeting personalized customer requests; flexibility; optimized decision-making; resource productivity and efficiency; creating value opportunities through new services; responding to demographic change in the workplace; improved work-life-balance.

It is stated that there is an expected increase of 23% (78.77 billion Euros) in Germany's Gross Domestic Product (GDP) from 2013 to 2025 based on the implementation of Industry 4.0 technologies [3]. Therefore, many companies face the challenge of assessing the diversity of developments and concepts summarized under the term Industry 4.0 and developing their own corporate strategies [4]. The companies with on-premise technologies try to re-shape their operations in line with emerging technologies in order to stay competitive and survive in the market place. Since Industry 4.0 is still in the initial stages of its development, it is essential to clearly define the structure and methodology of implementation guidelines for Industry 4.0 specifically. Therefore, there is a fundamental need to assist organizations which are transitioning to the Industry 4.0 environment and to guide them for improving their capabilities.

Structural approaches such as maturity models (MMs) or frameworks aim to assist organizations by providing comprehensive guidance and introducing a road map. The notion of maturity is used to define, assess and form a guideline and a basis for evaluating the progress in business (i.e., the maturity of process or a technology). The main idea for using MM is to describe the level of perfectness for an entity such as a new business model employed or a new software developed. The underlying supposition of using MMs is that, as the degree of maturity becomes higher, better progress is achieved in different aspects that contribute to the maturation of the entity. Therefore, the maturity model is considered as a baseline of this study.

The objectives of this study are to determine the sufficiency of the existing Industry 4.0 maturity models/frameworks for providing insights about the organization's maturity for adoption of Industry 4.0 as well as to identify the strengths and weaknesses of the existing models/frameworks. Accordingly, the literature is reviewed systematically to identify studies related to the maturity of Industry 4.0. Then, the review results are analyzed by comparing the characteristics of the models/frameworks according to a set of predefined criteria. After observing the strengths and weaknesses of these existing models/frameworks, a new SPICE based MM for Industry 4.0, named as *Industry 4.0-MM*, is proposed. The aim of the model is to provide a means for assessing a manufacturer's current Industry 4.0 maturity stage and for identifying concrete measures to help them reach a higher maturity stage in order to maximize the economic benefits of Industry 4.0.

The remainder of the paper is structured as follows: First, a brief explanation of the Industry 4.0 concept is given, followed by the results of the systematic literature review and the analysis of MMs in the context of the Industry 4.0. Then, the development of Industry 4.0-MM and the conclusion of the study are presented.

2 Background of the Study

2.1 Industry 4.0

As shown in Fig. 1, the process of industrialization began with the introduction of mechanical manufacturing equipment by the end of 18th century. The 1st Industrial Revolution is related to the transformation to mechanization by using hydro power and steam power. With this, the transformation from an agricultural to an industrial society started taking place. This revolution was followed by the 2nd Industrial Revolution around the turn of the 20th century, which involved automated mechanics in manufacturing that consumes electric power (i.e., mass-production). It was predominantly originated by organizational changes such as the implementation of Henry Ford's assembly line and the scientific management procedures highlighted by Frederic W. Taylor, better known as Taylorism. This was followed by the 3rd Industrial Revolution that started around 1969. This revolution is characterized by the implementation of information and communication technologies to achieve increased automation of manufacturing processes, as machines gradually take over and replace a large proportion of labor work. With the 3rd Industrial Revolution, the automation in the industry is improved by employing intelligent systems such as industrial robotics, and the domains of "intelligent mechatronics and robotics" [5] attract many practitioners due to various advantages. Consequently, a new concept referred to as Cyber-Physical Systems (CPS) that combine Internet of Things (IoT) technologies with the manufacturing ecosystem introduces a new era of the industrialization [6], which is seen as a significant paradigm shift in industrial manufacturing, named as Industry 4.0. Thereby, physical items are supplemented by their virtual representations in order to increase the automation, flexibility, and diversity of products by means of having better integrated manufacturing processes and systems [2]. Industry 4.0 is defined by Acatech [2] as *"the technical integration of CPS into manufacturing and logistics and the use of the Internet of Things and Services in industrial processes. This will have implications for value creation, business models, downstream services and work organization".*

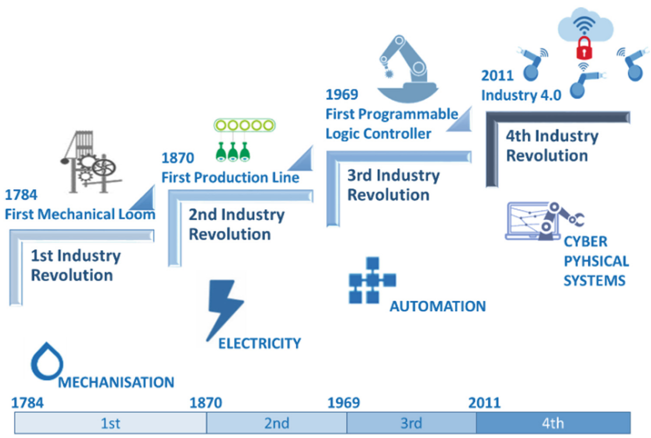

Fig. 1. History of industrial revolutions

The fundamentals of Industry 4.0 are highlighted by many researchers. Since Industry 4.0 covers many technologies, some of them overlap in terms of the functionalities and capabilities they offer. The essential technologies that are employed within the context of Industry 4.0 are as follows: Cloud Computing [7, 8]; Big Data [7, 8]; Internet of Things (IoT) [7]; Industrial Wireless Networks [9], Cyber-Physical Systems [10], Augmented Reality [11], Machine Learning [12], and Cyber Security [10].

Industry 4.0 targets the implementation of interconnecting, smart, and self-controlled structures of processes and systems [13]. Therefore, business processes based on the technologies underlying Industry 4.0 provide innovative value-added processes, providing more flexible, reliable, and efficient operations. In general, the latest developments in technology by itself offer new business opportunities and create new business models [14]; while Industry 4.0 is seen as a disruptive technological development that brings a new business model innovation in the manufacturing sector [14]. Because it is still in the initial stage of development, it is essential to define the structure and methodology of the implementation guidelines for Industry 4.0 for successful implementation in the industry. In order to investigate the success of manufacturing maturity in the context of the Industry 4.0, several maturity models have already been developed as discussed in the following section.

2.2 Industry 4.0 Maturity Models/Frameworks

A systematic literature review according to the guidelines proposed by Kitchenham [15] was applied in order to find existing maturity models(MMs) in the context of Industry 4.0 as follows: MMs for Industry 4.0 success/readiness were selected as the starting point of the search. Search terms of "Industry 4.0", "Industrial Internet of Things", "Industry Internet", "Industrial Internet", "Cloud-based Manufacturing", "Digitization", "Smart Manufacturing", "Cyber-physical systems", "Smart Factory", "Ubiquitous Manufacturing", and "Maturity Model" etc. were used. The databases of Scopus, Aisel, and Web of Science were scanned. 88 articles were identified with the search terms. Additionally, their references were reviewed. 36 papers are identified by reviewing references. In the results, journals covered by the SSCI and SCI indexes, and also conference proceedings (because there are only a few journals that investigate the MM for Industry 4.0 success) were included, while other publications (series, meetings, and reviews) were not. As part of the first elimination, the studies were evaluated in terms of their suitability by examining their keywords, titles, and abstracts, before reading the papers fully. The second elimination was performed by examining the relevance of the study. After applying these steps, 18 studies remained. As a result of a detailed analysis after reading studies fully, it is concluded that only 7 of 18 studies investigate the maturity of Industry 4.0. These 7 studies, given in Table 1, are considered as the base studies.

In order to analyze existing MMs objectively, a set of assessment criteria which have been employed in similar studies is identified based on the literature. Although there has not been such a study publishing the criteria for assessing the qualification of the Industry 4.0 MMs, we examined similar studies in the literature performed with SPICE and CMMI in order to identify the assessment criteria.

Table 1. Existing maturity models (MMs) in the context of Industry 4.0

Model	Model/Research name	Research context	Maturity levels	Dimensions
MM1 [16]	The connected enterprise maturity model	IT readiness	5 maturity stages (Assessment; Secure and upgraded network controls; Defined and organized working data capital (WDC); Analytics; Collaboration)	4 dimensions related to technological readiness. As stated in [17], no further information is provided related to aspect dimensions and the creation process of them
MM2 [18]	IMPULS – Industrie 4.0 readiness	Industry 4.0 readiness	6 maturity levels (Outsiders; Beginner; Intermediate; Experienced; Expert; Top performers)	6 dimensions (Strategy & Organization, Smart Factory, Smart Operations, Smart Products, Data-driven Services, and Employees)
MM3 [19]	Empowered and implementation strategy for Industry 4.0	Implementation strategies of Industry 4.0	No information provided regarding the MM	No information provided regarding the MM
MM4 [20]	Industry 4.0/digital operations self-assessment	Digital readiness for Industry 4.0	3 maturity levels (Vertical Integrator; Horizontal Collaborator; Digital Champion)	6 dimensions (Business Models; Product & Service; Portfolio Market & Customer Access; Value Chains & Processes; IT Architecture; Compliance, Legal, Risk, Security & Tax; Organization & Culture)
MM5 [17]	A maturity model for Industry 4.0 Readiness	Industry 4.0 maturity	Likert-scale maturity levels (from rating 1 = "not important"; to rating 4 = "very important")	8 dimensions (Strategy, Leadership, Customer, Products, Operations, Culture, People, Governance, Technology)
MM6 [21]	Towards a maturity model for Industrial Internet	Industrial Internet maturity	The research is not completed yet. No information regarding the MM	The research is not completed yet. No information regarding the MM
MM7 [22, 23]	SIMMI 4.0	Industry 4.0 maturity	5 maturity stages (Basic Digitization; Cross-Departmental Digitization; Horizontal and Vertical Digitization; Full Digitization; Optimized Full Digitization)	3 dimensions (Vertical Integration, Horizontal Integration, Cross-sectional Technology Criteria)

Rout et al. [24] criticize "*the purpose, the scope, the elements and the indicators*" of CMMI and mapping capability of CMMI with ISO 15504 and maturity results' verifiability based on *completeness-clearness-unambiguity* criteria. ISO/IEC 15504 Part7 [25] - "Assessment of Organizational Maturity" defines the purpose of this part as "*ensuring that the results are objective, impartial, consistent, repeatable, comparable and representative of the assessed organizational units*". Accordingly, we set the

Table 2. Assessment criteria for gap analysis

Criteria #	Criteria	Definitions
C1	Fitness for purpose	The level of fitness of the corresponding MM in terms of measuring maturity level in the context of Industry 4.0
C2	Completeness of aspects	The level of completeness of aspects in terms of addressing all or a subset of major aspects in the context of Industry 4.0
C3	Granularity of dimensions	The level of detail of explanations of the attributes in the corresponding dimensions
C4	Definition of measurement attributes	It questions whether the corresponding MM provides the description of the measurement attributes, or not
C5	Description of assessment method	It questions if the study provides a complete description of the assessment method
C6	Objectivity of the assessment method	The level of objectivity of maturity assessment method of the study. The definitions of the attributes, practices, and each level of the maturity should be described unambiguously. And the overall maturity level should correctly refect the number of questions positively answered

assessment criteria, described in Table 2, as compatible with these studies. Then, each study is evaluated according to those criteria. Consequently, the strengths and weaknesses of each MM are stated in a systematic way.

The qualifications of the criteria, such as Fitness for Purpose, Completeness of the Aspects, are expressed as a four-level scale: the rating that represents the extent of achievement of the criteria, as FA (Fully Achieved) meaning 86% to 100%, LA (Largely Achieved) meaning 51% to 85%, PA (Partially Achieved) meaning 16% to 50%, or NA (Not Achieved) meaning 1% to 15% of achievement.

The analysis was performed by three experts regarding the fields of MMs and Industry 4.0. First, they reviewed MMs independently based on the criteria given in Table 2. Then, they discussed them in a meeting in order to address conflicts and reached a consensus. Accordingly, the analysis results are briefly described in Table 3.

Table 3. Analysis of existing MMs in the context of Industry 4.0

Maturity model	C1	C2	C3	C4	C5	C6
MM1	NA	PA	NA	NA	NA	NA
MM2	PA	PA	PA	LA	FA	LA
MM3	NA	NA	NA	NA	NA	NA
MM4	PA	PA	PA	PA	NA	PA
MM5	PA	PA	PA	PA	PA	PA
MM6	PA	NA	NA	NA	NA	NA
MM7	PA	PA	PA	PA	LA	PA

As an example, MM1 is analyzed according to the criteria as follows. Since the proposed maturity model only investigates the technological readiness of enterprises, it does not specifically target the maturity of enterprises in the context of Industry 4.0. Therefore, C1 is stated as NA that means the fitness for the purpose is below 16% achievement. The completeness of aspect is partially achieved, because MM1 presents a set of dimensions that investigate the technological readiness. However, organizational and environmental aspects are not considered for evaluating the maturity of the enterprise. Therefore, C2 is stated as PA representing 51% to 85% achievement. Aspect dimensions are not provided, and there is no detailed information related to technological dimensions, therefore the levels of detail presented in the creation of dimensions appear to be very poor and C3 is represented by NA. Furthermore, there is no detailed information about definitions of measurement attribute; description of assessment method; and the information to be able to identify objectivity of assessment method are not provided in the study. Consequently, the overall experts' rate is below 15% achievement for C4, C5, and C6; therefore they are all represented with NA.
The findings of the analysis are as follows:

- **MM1** [16] proposes a MM that mainly investigates the aspect of technological readiness of enterprises for Industry 4.0. The assessment relies on 5 maturity stages, but the dimensions of the model are not provided and there is no detailed explanation regarding them and their items.
- **MM2** [18] proposes 6 dimensions that measure the Industry 4.0 readiness. After defining the maturity score as a percentage, they provide an action plan to boost the readiness in the context of technology, environment, and organization. However, the maturity level of the company is affected by the maturity level of competitor organizations. But by definition of the assessment method of this model, the competitor's maturity level is defined only if any other organization in the same market takes the survey, otherwise it is ignored.
- **MM3**- the maturity for Industry 4.0 is only a small part of the study, there is no information provided regarding the structure of MM. There is no detailed information related to dimensions and their items [17].
- **MM4**- It offers an online self-assessment tool for Industry 4.0 readiness, but the model only focuses on digital readiness for Industry 4.0 and consists of 6 dimensions. Items of each dimension and the creation of each item are neither explained, nor shared with the users.
- **MM5**- It proposes 8 dimensions for the assessment. The assessment method relies on the rating of each item by using the Likert-scale. The model is easy to apply for the assessment of maturity level. However, the model only produces an overall score indicating the maturity level. It does not provide an action plan to overcome weak sides of the enterprises being assessed.
- **MM6**- It is constructed based on Mettler's framework [26]. The study defines the design guideline for maturity modeling in the context of Industrial Internet. Since the research is not completed yet, there is no proposed MM.
- **MM7**- It only focuses on software/technological aspects of the maturity of Industry 4.0. The organizational (i.e., employees, company vision, etc.) and environmental

aspects (i.e., competitors, market structure, etc.) are not considered in the assessment of the maturity.

The systematic literature review can be summarized by concluding that there is a growing research stream in Industry 4.0 in recent years, however, there is a research gap due to limited research in the use of MMs for Industry 4.0. As given above, existing MMs in the literature were analyzed based on the criteria of scope, purpose, completeness, clearness (Definition of measurement attributes and methods), and objectivity. As seen in Table 3, it is concluded that none of them satisfies all criteria and they need to be improved. The most obvious deficiency of the models is that they don't support manufacturing enterprise architecture holistically. Additionally, none of them is developed based on a well-accepted framework for the assessment and improvement, and they do not have a well-defined structure with practices, inputs and outputs. Among these studies, there is not a well-accepted MM for Industry 4.0. The need for a structured Industry 4.0 assessment/maturity model remains valid. The aim of this study is to satisfy this need by providing Industry 4.0-MM. The development of Industry 4.0-MM is provided in the next section.

3 Development of Industry 4.0-MM

It is aimed to develop a new MM for Industry 4.0 to determine which capabilities a manufacturing organization needs to acquire in order to successfully introduce Indus-try 4.0 in a standardized way. Furthermore, it serves as a roadmap for transitioning to Industry 4.0. The model is aimed to provide a complete and comprehensive guideline for enabling organizations to observe their problematic areas and weaknesses as well as practices for applying the transformation to Industry 4.0 in a consistent way.

There are various well-accepted generic Software Process Capability/Maturity Models (SPCMMs), such as ISO/IEC 15504 [27] - also termed as Software Process Improvement and Capability dEtermination (SPICE) replaced with ISO 33000 series [28] and CMMI-DEV (Capability Maturity Model Integration for Development) model [29]. As a result of the observed benefits of these SPCMMs such as expense savings, increased involvement of employees, predictable and improved quality and productivity, customizing them to different domains other than software development is the subject of growing interest in the literature. Accordingly, many initiatives based on these SPCMMs have been proposed for various domain such as [30–33].

The aim of this study is to utilize the same approach for the Industry 4.0 domain by proposing a MM for Industry 4.0 based on SPICE. The reason for selecting SPICE is its well defined and commonly accepted structure for the assessment and improvements and its suitability for the development of maturity level assessment of organizations in the context of Industry 4.0.

3.1 Industry4.0-MM in Relation to SPICE

The structure of Industry4.0-MM is formed based on the ISO/IEC 15504 Part 2 [34] and ISO/IEC 15504 Part 5 [27] in order to obtain a common baseline for capability assessment, and to report the assessment outcome by employing a common measurement scale. Our purpose is to create a common basis for assessing establishment of Industry 4.0 technologies and to present the assessment results using a common rating scale. However, instead of process dimension of SPICE, we define Aspect dimension, which can be seen in Fig. 2.

It is difficult to identify boundaries of the process transformation for utilization of Industry 4.0. Besides process transformation, other dimensions such as infrastructure, information systems, data and organization as well as their integration are critical in the establishment of Industry 4.0. Hence, a new arrangement of Industry 4.0 processes and practices is performed to integrate them under meaningful and compatible abstract definitions, referred to as "Aspects".

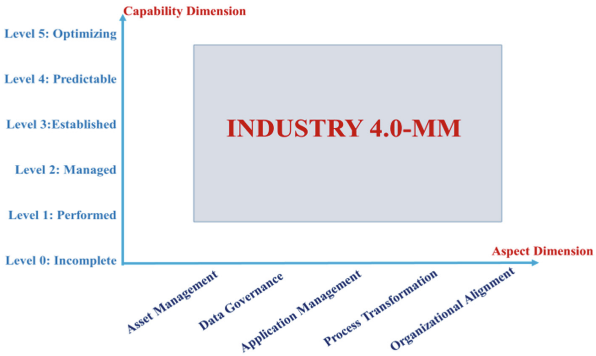

Fig. 2. The structure of the proposed Industry 4.0-MM

In the aspect dimension, aspects are defined and classified into such categories as Asset Management, Data Governance, Application Management, Process Transformation, and Organizational Alignment. The capability dimension is defined by capability levels and capability indicators. The capability dimension is adopted from SPICE, it has 6 levels, from "Level 0: Incomplete" to "Level 5: Optimizing", as seen in Figs. 3 and 4. The mappings for the constructs of the proposed Industry 4.0-MM and SPICE components are provided in Table 4.

3.2 Aspect Dimension

It has a holistic view consisting of Asset Management, Data Governance, Application Management, Process Transformation, and Organizational Alignment areas.

Asset Management: It covers IT systems of the organization, and technological readiness for Industry 4.0, usage of emerging business technologies (for instance, Cloud Computing based enterprise solutions), and security issues of smart technologies. This dimension measures the level of support that the organization can provide for cutting-edge technology domains which include Service-oriented Architecture; Cloud Computing; IT Security; Internet of Things (IoT); Industrial Wireless Networks.

Table 4. Mapping SPICE components to Industry 4.0-MM

SPICE	Industry 4.0-MM	Explanation
Processes	Aspect	Aspects inherit and include processes
Capability	Capability	Characterization of the ability of an aspect to meet current or projected business goals
Maturity	Maturity	Point on an ordinal scale that characterizes the maturity of the organization assessed in the scope of Industry 4.0-MM
Base practice	Base aspect practice	The actions or action groups that affect the achievement of the corresponding aspect purpose. These defined aspect practices provide a road map for an enterprise to transition into an Industry 4.0 environment
Process attribute	Aspect attribute	The indicator of the corresponding aspect performance is referred to as aspect attribute. It identifies the features of the aspect
Generic practice	Generic aspect practice	Activity that, when consistently performed, contributes to the achievement of a specified aspect attribute

On-premise infrastructure and IT resources (e.g. networking equipment, essential hardware, and other secondary applications) are considered as significant factors for implementing new technologies [35]. Furthermore, perceived security of the Industry 4.0 concept may stimulate enterprises or constitute a barrier while moving from a traditional business system to a smart environment. This is because enterprises are concerned about their operations' confidentiality and integrity, due to the fact that unauthorized operations or unexpected data loss can cause catastrophic outcomes for the business.

Data Governance: This aspect investigates the capability level of the following: data collection, usage, data analytics and big data tools, and data-driven services. Gathering and complete assessment of data from various sources, including manufacturing infrastructure and systems as well as information systems, enable organizations to make real-time decisions regarding current or future operations [36]. Therefore, an integrated and automatized data flow is critical both inside and outside the enterprise.

Application Management: With Industry 4.0, it is expected that revolutionary applications will come about principally as a result of combining applications with manufacturing and automation technologies [2]. It is aimed to ensure an optimal and secure design and construction of information systems that best work for its business and users. Application capabilities provide an abstract perspective on the functional behavior required to support the business. Interfaces and information flow of applications should be structured, connected, standardized, controlled and interoperable.

Organizational Alignment: It refers to the management of enterprises through Enterprise Architecture in terms of organizational structure, and strategy of the business. From the managerial point of view, the knowledge about the advantages of the smart manufacturing concept significantly affects the decision of IT investment and implementation. Since the IT personnel skill set and other essential human resource requirements for transformation are related to the organizational management of enterprises, this dimension is considered as a significant factor for assessing the capability.

Process Transformation: This aspect covers the transformation of the basic processes of each enterprise system which are Planning, Acquisition, Production, and Sale & Distribution. According to the business structure of the enterprise, once the transformation to Industry 4.0 begins, each process of the enterprise system should be mapped to the digital world. Furthermore, different value-added processes should be integrated across the enterprise architecture in a standardized manner.

3.3 Capability Dimension

It is important to recognize that successful transformations happen in stages. This model results in the formulation of a roadmap in all relevant areas with a step-by-step approach to achieve the benefits that reduce the investment and implementation risks for the organization. The model's approach is based on a succession of capability stages, from the basic requirements for Industry 4.0 to the full implementation. Each stage builds upon the previous one. As seen in Figs. 3 and 4, Industry 4.0-MM has six capability levels which are adopted from SPICE and the aspect attributes defined for each level are developed based on ISO 33003: Process Assessment- Requirements for Process Measurement Frameworks [28].

Level 0 Incomplete: Base aspect practices are partially achieved, or there is no implementation yet. The organization only focuses on the fundamental operations such as requirements analysis, acquisition, production, and sales.

Level 1 Performed: The corresponding aspect practices are achieved. Transformation has been started. Technological infrastructure for transitioning to Industry 4.0 is acquired, and the organization tends to employ smart technologies such as IoT. The vision of Industry 4.0 exists, and there is a roadmap for the transition strategy, yet it is not fully implemented. Aspect Attribute (AA) 1.1 Perform Aspect Practices are assessed at this level.

Level 2 Managed: Data set related to each operation is defined and started to be collected, but they are not integrated into the different functionalities of the operations. Physical items are starting to be represented by a virtual world. AA 2.1 Digitalization is assessed at this level.

Level 3 Established: Key activities of the business, value added operations are well-defined, and qualifications of processes and operations are consistent with corresponding standardization. The data set is clearly identified for each operation of the organization, and collected and systematically stored in a well-managed database. Vertical integration including factory-internal integration of sensors and actuators within machines up to Enterprise Resource Planning systems has been achieved. AA 3.1 Vertical Integration and AA 3.2 Standardization are assessed at this level.

Level 4 Predictable: Horizontal integration, including the integration of production networks at the business level is achieved by supply chain integration, but might

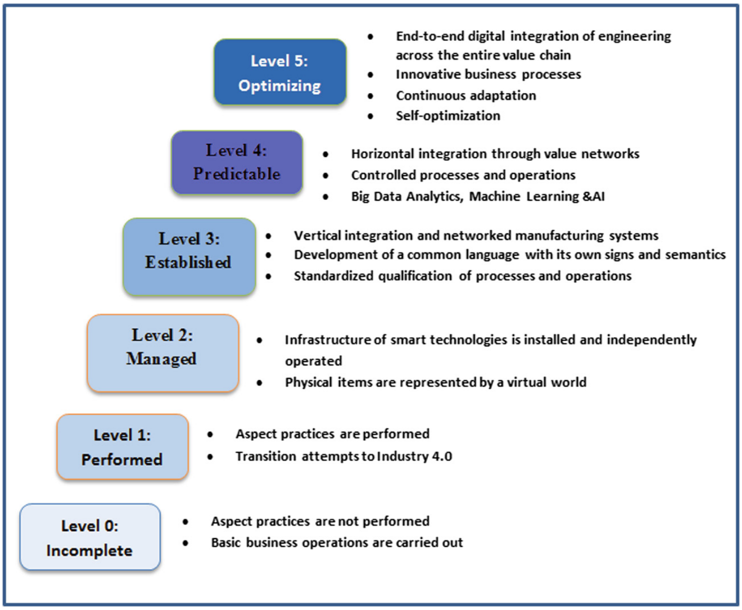

Fig. 3. Capability dimensions of Industry 4.0-MM

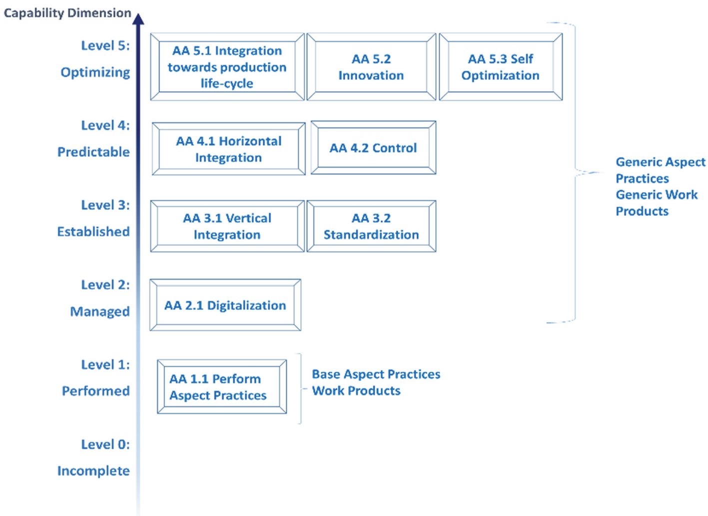

Fig. 4. Capability levels of Industry 4.0-MM

include more in the future, when close-to-real-time and product- or process-specific information is exchanged to increase the level of detail and quality in distributed manufacturing optimization.

Data analytics tools are employed to improve productivity of manufacturing organizations. The functionalities of whole enterprises are integrated in order to increase the efficiency of operations (i.e., the integration of SCM and CRM applications). Data are used to control the process and operations in real-time. AA 4.1 Horizontal Integration and AA 4.2 Control are assessed at this level.

Level 5 Optimizing: Integration towards engineering and product/production life to enable low-effort knowledge sharing and synchronization between product and service development and manufacturing environments has been achieved. The organization starts to learn from the collected data, and tries to improve its business continuously. The business model is evolving into an innovative structure. AA 5.1 Integration towards production life-cycle, AA 5.2 Innovation, and AA 5.3 Self Optimization are assessed at this level.

4 Conclusion

Since Industry 4.0 is still in the initial stages of its development, it is essential to clearly define the structure and methodology of implementation guidelines for Industry 4.0 specifically. Therefore, there is a fundamental need to assist companies in their transitions to utilization of Industry 4.0 technologies/practices, and to guide them for improving their capabilities in a standardized, objective, and repeatable way. Structural approaches such as MMs aim to assist organizations by providing comprehensive guidance and introducing a road map. Accordingly, the literature is reviewed systematically to find out about existing studies related to MMs of Industry 4.0. As a result of the review, 7 MMs have been identified, and they are analyzed by comparing the characteristics of the models/frameworks based on a set of predefined criteria including scope, purpose, completeness, clearness, and objectivity. It is concluded that none of them satisfies all expected criteria and they need to be improved. In order to satisfy the need for a structured Industry 4.0 assessment/maturity model, SPICE-based Industry 4.0-MM is proposed in this study. Industry 4.0-MM has a holistic approach consisting of the assessment of process transformation, application management, data governance, asset management, and organizational alignment areas. The capability dimension of the proposed model is adopted based on SPICE, and Process Attributes of SPICE are replaced by a total of nine Aspect Attributes.

Industry 4.0-MM is aimed to provide the following benefits: standardization in development, higher quality, more flexibility, continuous benchmarking and improvement, global competition among strong businesses, creation of appealing jobs at the intersection of mechanical engineering, automation, and IT, new services and business models.

As part of future work, it is planned to conduct an exploratory case study in order to validate the usefulness and applicability of the proposed maturity model.

References

1. The Economist Intelligence Unit: The Internet of Things Business Index: A Quiet Revolution Gathers Pace (2013)
2. Kagermann, H., Wahlster, W., Held, J.: Deutsche P.A. recommendations for implementing the strategic initiative Industrie 4.0. Final report of the INDUSTRIE 4.0 WG, 82 (2013)
3. Bauer, W., Horváth, P.: Industrie 4.0 - Volkswirtschaftliches Potenzial für Deutschland. Controlling, **27**, 515–517 (2015)
4. Beckert, S.: Empire of Cotton: A Global History. Knopf, New York (2014)
5. Kunii, T.L.: The 3‘d industrial revolution through integrated intelligent processing systems. In: I997 IEEE lnternational Conference on Intelligent Processing Systems, pp. 1–6 (1997)
6. Shrouf, F., Ordieres, J., Miragliotta, G.: Smart factories in Industry 4.0: a review of the concept and of energy management approached in production based on the Internet of Things paradigm. In: IEEE International Conference on Industrial Engineering and Engineering Management, 2015–January, pp. 697–701 (2014)
7. Xiong, G., Ji, T., Zhang, X., Zhu, F., Liu, W.: Cloud operating system for industrial application. In: 10th IEEE International Conference on Service Operation and Logistics and Informatics, SOLI 2015 - conjunction with ICT4ALL 2015, pp. 43–48 (2015)
8. Schmidt, R., Möhring, M., Härting, R.-C., Reichstein, C., Neumaier, P., Jozinović, P.: Industry 4.0-potentials for creating smart products: empirical research results. In: International Conference on Business Information Systems, pp. 16–27 (2015)
9. Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., Vasilakos, A.: Software-defined industrial internet of things in the context of Industry 4.0. IEEE Sens. J. **1**, 7373–7380 (2016)
10. Hermann, M., Pentek, T., Otto, B.: Design principles for industrie 4.0 scenarios. In: Proceedings of Annual Hawaii International Conference on System Sciences, pp. 3928–3937, March 2016
11. Paelke, V.: Augmented reality in the smart factory: supporting workers in an industry 4.0 environment. In: 19th IEEE International Conference on Emerging Technology and Factory Automation, ETFA 2014 (2014)
12. Shin, S.-J., Woo, J., Rachuri, S.: Predictive analytics model for power consumption in manufacturing. Procedia CIRP **15**, 153–158 (2014)
13. Reiner, A.: Industrie 4. 0 - Advanced Engineering of Smart Products and Smart Production. International Seminar on High Technology, pp. 1–14 (2014)
14. Kagermann, H., Wahlster, W., Held, J.: Bericht der Promotorengruppe Kommunikation. Im Fokus: Das Zukunftsprojekt Industrie 4.0. Handlungsempfehlungen zur Umsetzung Forschungsunion (2012)
15. Kitchenham, B.: Procedures for performing systematic reviews, vol. 33, p. 28. Keele University, Keele (2004)
16. Rockwellautomation: The Connected Enterprise Maturity Model. 12 (2014)
17. Schumacher, A., Erol, S., Sihn, W.: A maturity model for assessing and maturity of manufacturing enterprises. Procedia CIRP **52**, 161–166 (2016)
18. Lichtblau, K., Stich, V., Bertenrath, R., Blum, M., Bleider, M., Millack, A., Schmitt, K., Schmitz, E., Schröter, M.: IMPULS - Industrie 4.0- Readiness (2015)
19. Lanza, G., Nyhuis, P., Ansari, S.M., Kuprat, T., Liebrecht, C.: Befähigungs- und Einführungsstrategien für Industrie 4.0 (2016)
20. PricewaterhouseCoopers: The Industry 4.0/Digital Operations Self Assessment (2016)
21. Menon, K., Kärkkäinen, H., Lasrado, L.A.: Towards a maturity modeling approach for the implementation of industrial internet. In: Pacis 2016 Proceedings (2016)

22. Leyh, C., Schäffer, T., Bley, K., Forstenhäusler, S.: SIMMI 4.0 – A Maturity Model for Classifying the Enterprise-wide IT and Software Landscape Focusing on Industry 4.0, vol. 8, pp. 1297–1302 (2016)
23. Ziemba, E. (ed.): AITM/ISM -2016. LNBIP, vol. 277. Springer, Cham (2017). doi:10.1007/978-3-319-53076-5
24. Rout, T., Tuffley, A., Cahill, B.: CMMI evaluation: capability maturity model integration mapping to ISO/IEC 15504 2: 1998. Software Quality Institute, Griffith University (2001)
25. ISO/IEC: 15504, Information Technology – Process Assessment Part 7: Assessment of organizational maturity (2008)
26. Mettler, T.: A design science research perspective on maturity models in information systems. University of St. Gallen, St. Gallen, Switzerland, Technical report BE IWI/HNE/03, 41 (2009)
27. ISO/IEC: 15504, Information Technology — Process assessment Part 5 : An exemplar software life cycle process assessment model. 1–210 (2012)
28. ISO: ISO/IEC 33020: Information technology — Process assessment — Process measurement framework for assessment of process capability (2015)
29. CMMI Product Team: CMMI® for Development, Version 1.2. Framework. 573 (2006)
30. Automotive SIG.: The SPICE user group automotive special interest group, Automotive SPICE Process Reference Model (2010)
31. Medi SPICE: Portal do projeto Medi SPICE (2011)
32. Mesquida, A.L., Mas, A., Amengual, E., Calvo-Manzano, J.A.: IT service management process improvement based on ISO/IEC 15504: a systematic review. Inf. Softw. Technol. **54**, 239–247 (2012)
33. Gökalp, E., Demirörs, O.: Model based process assessment for public financial and physical resource management processes. Comput. Stand. Interfaces **54**, 186–193 (2016)
34. ISO/IEC: Software engineering - Process assessment - Part 2: Performing an assessment, 3 (2004)
35. Oliveira, T., Thomas, M., Espadanal, M.: Assessing the determinants of cloud computing adoption: an analysis of the manufacturing and services sectors. Inf. Manag. **51**, 497–510 (2014)
36. Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., Harnisch, M.: Industry 4.0. The Future of Productivity and Growth in Manufacturing. Bost. Consult. 1–5 (2015)

A SPICE-Based Maturity Model for the Governance and Management of Green IT

J. David Patón-Romero[1(✉)], Moisés Rodríguez[1,2], and Mario Piattini[1]

[1] Alarcos Research Group, Institute of Technologies and Information Systems, University of Castilla-la Mancha (UCLM), Paseo de la Universidad, 4, 13071 Ciudad Real, Spain
JoseDavid.Paton@gmail.com, mrodriguez@aqclab.es, Mario.Piattini@uclm.es

[2] AQCLab, Camino de Moledores, 13051 Ciudad Real, Spain

Abstract. Organizations around the world are increasingly concerned about the environment, adopting sustainable practices in their business processes. In the field of Information Technologies (IT) several Green IT practices have been proposed, but in isolation, so a framework is needed if the Green IT is to be implemented and improved in an efficient and integrated way. In this paper, we propose a maturity model (based on SPICE) to help organizations to implement the governance and management of Green IT gradually, as well as to improve their maturity level in this area. The validation of this proposal by experts and a case study seems to indicate that the proposal can be useful for implementing and improving the Green IT processes in organizations.

Keywords: SPICE · ISO/IEC 15504 · Maturity model · Green IT · Governance · Management

1 Introduction

In recent years, Information Technology (IT) has become one of the pillars of our society, changing not only the way we relate to each other and the way companies do business, but also how we interact with the planet. However, in this interaction with the planet we have lost our commitment to the environment, our commitment to life. Therefore, in our society a strong ecological awareness has emerged in order to address this problem, with the aim of obtaining a healthy planet and a sustainable ecosystem.

That is why in the area of IT has emerged the concept of Green IT, which seeks to bring the idea of environmental sustainability [1] closer to IT. Green IT can be defined as "*the study and practice of design, manufacture and use of hardware, software and communication systems with a positive impact on the environment*" (definition adapted from [2]). The importance of the idea of sustainability in our society and the growing demand of "green" products has made Green IT a determining area, gaining increasing importance within organizations, since it has become an important asset to add value to the business.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 143–155, 2017.
DOI: 10.1007/978-3-319-67383-7_11

However, while there is a growing number of research papers and isolated best practices of Green IT, there are still no specific standards to help organizations to establish the bases of these best practices (the governance and management of Green IT) and to verify that these Green IT implementations are sufficient, correct and work as expected [3].

That is why we have developed a first version of the “Governance and Management Framework for Green IT” [4], in which we have established the necessary characteristics to define, implement and audit the governance and management of Green IT in an organization. This developed framework however lacks a maturity model through which the characteristics of governance and management of Green IT established in this framework can be gradually evaluated and implemented. Thus, in this paper we propose a maturity model based on SPICE (a process reference model) for the governance and management of Green IT, i.e., a SPICE-based maturity model for the “Governance and Management Framework for Green IT”.

The rest of this paper is organized as follows: Sect. 2 explains the related work about the existing maturity models that are related to the subject of study (SPICE, IT, Green and Green IT); Sect. 3 presents the proposal of the SPICE-based maturity model for the “Governance and Management Framework for Green IT”; Sect. 4 shows the validations carried out for the process reference model proposed; and, finally, Sect. 5 presents the conclusions and future work to be done in this area. Also, Appendix A shows the definitions and purposes of the processes of the “Governance and Management Framework for Green IT” organized according to the proposed SPICE maturity levels.

2 Related Work

In the following sub-sections, the different maturity models based on SPICE and related to the area of IT, sustainability (Green) and Green IT are analyzed.

2.1 SPICE-Based Maturity Models

The ISO/IEC 15504 [5], also known as Software Process Improvement Capability Determination (SPICE), is a set of standards, developed by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), that propose models for the improving and assessing of processes related to information systems and software products.

SPICE has been applied in many fields such as aerospace [6], software engineering [7], government [8], risk management [9], automotive [10, 11], information security [12], health [13, 14], nuclear energy [15], among others. However, so far there is no application of this standard in the field of Green IT. It is also important to note that a few years ago (in 2012 at the ISO/IEC JTC 1/SC 7 plenary meeting in Jeju, South Korea) a proposal for defining an extension to the ISO/IEC 15504 to embrace sustainability was presented and accepted but failed to obtain enough resources to be carried on. It was a shame because we believe that it was a missed opportunity to advance in this area.

2.2 Other IT Maturity Models

In relation to the others maturity models related to the area of IT, the most ingrained and widely-used today by organizations are outlined below:

- ISO/IEC 33000 [16]: this new family of standards, developed by ISO and IEC, is replacing ISO/IEC 15504, reorganizing and extending the latter for the evaluation and improvement of the capacity and maturity of an organization's processes. Based on these family of standards, we developed a model for data quality processes implantation [17].
- Capability Maturity Model Integration (CMMI) [18]: this model, originally developed by Carnegie Mellon University (CMU) and administered nowadays by the CMMI Institute (acquired recently by ISACA - *Information Systems Audit and Control Association*), aims to evaluate and improve the processes of an organization for the development, maintenance and operation of information systems and software products.
- Maximizing the Combined Effects of COBIT 5 and CMMI [19]: this proposal is being developed by ISACA, in order to adapt the CMMI model to COBIT 5 (*Control Objectives for Information and related Technology*) [20], identifying at which maturity levels of the CMMI model must the different COBIT 5 processes of governance and management of IT be included.

2.3 Green and Green IT Maturity Models

In [21] a systematic mapping study is carried out in relation to the sustainability maturity models that currently exist, placing special emphasis on the area of Green IT. The study demonstrates the limited number of studies related to maturity models of sustainability (only 26 studies have been found) and, in particular, of Green IT (only 8 studies in this field). This systematic mapping study also shows the need to validate the maturity models proposed by the studies, since only 8 of the studies found validate their proposal. And, in particular, in relation to the proposed Green IT maturity models, only 2 are validated: study [22] carries out a validation through a case study, and study [23] through a survey.

It is important to highlight that, in addition to the Green IT maturity models found in this systematic mapping study, we have found as gray literature another study [24] that proposes a Green IT maturity model based on CMMI, validated through a case study. On the other hand, in the results of the systematic mapping study we can observe that there are no sustainability models or Green IT models that follow ISO/IEC 15504 (SPICE), which demonstrates the importance of exploiting this area of SPICE-based maturity models of sustainability. Therefore, the results of this systematic mapping study demonstrate the youth of this area of maturity models related to sustainability. Also, in relation to Green IT, it is not only important to develop common and updated frameworks, but also maturity models for these frameworks that allow the gradual implementation, evaluation and improvement of Green IT practices carried out by organizations.

3 SPICE-Based Maturity Model for the "Governance and Management Framework for Green IT"

The great growth of the idea of sustainability and, in particular, Green IT within organizations has led to the emergence of more and more research papers and isolated best practices in this respect. That is why, in the absence of a framework or standard to carry out these Green IT practices, we have developed a first proposal of the "Governance and Management Framework for Green IT" [4] (GMGIT, hereinafter), based on the structure of enablers of the COBIT 5 framework [20], which aims to optimize and standardize the adoption of Green IT in organizations.

However, this first version of the GMGIT lacks a maturity model that allows organizations to gradually implement, evaluate and improve their maturity level in the area of governance and management of Green IT. For this reason, in this paper we propose a maturity model (a process reference model) for the "Governance and Management Framework for Green IT", based on the default standard to evaluate and improve the maturity level in IT, ISO/IEC 15504 (SPICE). The application of the different characteristics of the SPICE standard to the "Governance and Management Framework for Green IT" is shown below. First, SPICE establishes 6 maturity levels, which we have adapted to the area of Green IT as follows:

- **Level 0 (Incomplete).** The organization does not take sustainability into account, and no Green IT practice is defined.
- **Level 1 (Performed).** The organization takes sustainability into account, and carries out Green IT practices in the most critical aspects related to sustainability.
- **Level 2 (Managed).** The Green IT practices are clearly defined, established and managed throughout the different business areas, contributing to sustainability in and by IT.
- **Level 3 (Established).** The organization follows the recognized standards and best practices of Green IT (Green IT is correctly managed and governed), as well as identifies in a continuous way and ensures the compliance with the external requirements.
- **Level 4 (Predictable).** The organization carries out the monitoring, evaluation and measurement of implemented Green IT practices, through a set of sustainability metrics established for that purpose.
- **Level 5 (Optimizing).** The organization is fully committed to sustainability and is oriented towards the continuous improvement of implemented Green IT practices, by means such as for example detailed performance reports, exhaustive use of sustainability metrics, and management of the innovation process in sustainability.

Second, in each of these maturity levels of Green IT, the different processes of the GMGIT have encompassed, as shown in the Table 1. It is important to note that the GMGIT does not include all the processes defined by COBIT 5, but of the 37 processes of COBIT 5 we select and adapt to Green IT 15 of them, which we consider most directly related to this area.

Table 1. SPICE maturity levels of the processes of the "Governance and Management Framework for Green IT".

Process	Level 1	Level 2	Level 3	Level 4	Level 5
EDM01: Ensure governance framework setting and maintenance			X		
EDM02: Ensure benefits delivery			X		
EDM03: Ensure risk optimization					X
EDM04: Ensure resource optimization					X
EDM05: Ensure stakeholder transparency			X		
APO01: Manage the IT management framework		X			
APO02: Manage strategy		X			
APO06: Manage budget and costs		X			
APO08: Manage relationships		X			
BAI02: Manage requirements definition		X			
BAI03: Manage solutions identification and build		X			
BAI09: Manage assets	X				
DSS01: Manage operations	X				
MEA01: Monitor, evaluate and assess performance and conformance				X	
MEA03: Monitor, evaluate and assess compliance with external requirements			X		

Finally, we have described each of the GMGIT processes according to the SPICE standard, i.e., identifying the attributes of each process, through which the compliance with said process can be analyzed. Table 2 shows by way of example the SPICE-based description of one of the GMGIT processes.

Table 2. SPICE-based description of the process "DSS01: Manage operations".

Attribute	Description
Process ID	DSS01
Process Name	Manage operations
Process Description	Co-ordinate and execute the activities and operational procedures required to deliver internal and outsourced IT services, including the execution of pre-defined standard operating procedures and the required monitoring activities
Process Purpose	Deliver IT operational service outcomes as planned
Process Outcomes	As a result of successful implementation of "Manage operations": 1.The operations of Green IT are carried out following the policies, principles, strategy and goals of Green IT 2.The standards, regulations and best practices of Green IT have been identified and implemented and are being complied with

(*continued*)

Table 2. (*continued*)

Attribute	Description	
Best Practices	**DSS01.BP1: Perform operational procedures.** Maintain and perform operational procedures and operational tasks of Green IT reliably and consistently. [Outcome: 1] **DSS01.BP2: Manage outsourced services.** Manage the operation of outsourced services so as to maintain their reliability and their consistency with the organization's Green IT. [Outcome: 1] **DSS01.BP3: Monitor IT infrastructure.** Monitor the IT infrastructure and events related to it, in an effort to ensure the alignment of all of them with the organization's Green IT. Store sufficient chronological information in operations logs to enable the reconstruction, review and examination of the time sequences of operations and the other activities surrounding or supporting those operations. [Outcome: 2] **DSS01.BP4: Manage the environment.** Maintain measures for protection against environmental factors. Install specialized equipment and devices to monitor and control the environment from the point of view of Green IT. [Outcome: 2] **DSS01.BP5: Manage facilities.** Manage facilities in line with laws, regulations, guidelines and other requirements related to Green IT. [Outcome: 2]	
Work Products	Inputs	Outputs
	Policies of Green IT. [Outcome: 1]	Operational procedures of Green IT. [Outcome: 1]
	Policies of management of the environment. [Outcome: 2]	Reports on the compliance of Green IT by third parties. [Outcome: 1]
	Policies of management of the facilities. [Outcome: 2]	Reports on the performance of the infrastructure of the IT, from the point of view of Green IT. [Outcome: 2]
		Alignment of Green IT with the management of the environment. [Outcome: 2]
		Alignment of Green IT with the management of the facilities. [Outcome: 2]

4 Validations

To verify the consistency and applicability of the SPICE-based maturity model for the GMGIT proposed in the previous section, we have carried out a couple of validations through a workshop with experts and through a case study in an IT service center.

4.1 Workshop

First of all, we decided to hold a workshop with experts in order to obtain a validation from a theoretical point of view, refining the proposed model before moving on to the practical level. These experts, five in all, belong to an IT department, have more than 10 years of experience in research and IT audits (with certification in CISA - *Certified*

Information Systems Auditor), and are currently working on issues related to Green IT, IT, auditing and maturity models.

During the workshop, the GMGIT and the SPICE maturity levels adapted to Green IT were presented and discussed first. Following this, each of the experts was asked for his proposal about at what maturity level should be found each of the processes defined in the GMGIT and each of these proposals was discussed.

After discussing the proposals of the experts and reaching a general proposal, we presented our proposal of the SPICE maturity levels of each of the GMGIT processes. Both proposals were discussed and the final proposal of the SPICE-based maturity model for the GMGIT was reached.

4.2 Application in a IT Services Center

In second place, we carried out a case study in a IT service center (for reasons of confidentiality identified hereinafter as SC), which is responsible for the management of IT services of a university of more than 30,000 students and is distributed in several campuses. Currently, the SC is beginning to implement sustainable measures in different areas of the business, including the following Green IT measures:

- Implementation of cloud computing services.
- Establishment of a corporate printing service, reducing the number of printing devices and raising awareness of the need to save ink and paper.
- Implementation of a service of withdrawal and recycling of electrical and electronic waste.
- Acquisition of IT equipment according to internationally recognized sustainability standards such as UE Energy Star v5, ISO 14001 o ISO 779/9296.
- Redesign of the data center, to improve energy efficiency and cooling.

Thanks to these Green IT measures, the SC has achieved good results in favor of environmental sustainability:

- Reduction of 20% of the energy destined for the cooling of the data center (obtaining a PUE - *Power Usage Effectiveness* of 1.4).
- Reduction of 52% of CO_2 emissions from university IT.
- Withdrawal of more than 48 tons of obsolete computer equipment for recycling.

From these results, it is estimated that the university has avoided the generation of 7,261 kg of CO_2 and has produced a saving of 2,631 m^3 of water. However, these Green IT practices have been carried out in an isolated manner and without following a specific framework or standard. For this reason, the SC decided to carry out an audit following the GMGIT, in order to know its current state of Green IT and adopt the framework to implement, evaluate and improve the Green IT.

In this audit, the high involvement of the SC with sustainability was observed, but many shortcomings were identified, especially in the definition and formalization of the Green IT practices. Analyzing these results and applying them to the developed SPICE-based maturity model, we have concluded that the SC is partially at Level 1, as can be seen in Table 3.

Table 3. Fulfillment of the processes and their best practices of Level 1 in the SC.

Processes and their Best Practices of Level 1	Yes	Partially	No
BAI09: Manage assets			**X**
BAI09.BP1: Identify and record current assets			X
BAI09.BP2: Manage critical assets			X
BAI09.BP3: Manage the asset life cycle			X
BAI09.BP4: Optimize asset costs			X
BAI09.BP5: Manage licenses			X
DSS01: Manage operations		**X**	
DSS01.BP1: Perform operational procedures		X	
DSS01.BP2: Manage outsourced services	X		
DSS01.BP3: Monitor IT infrastructure	X		
DSS01.BP4: Manage the environment			X
DSS01.BP5: Manage facilities	X		

We are currently working with the SC to overcome the deficiencies found, in order to reach the Level 1 of maturity of Green IT and start to work on the following levels, gradually implementing the Green IT and improving its maturity level in this area.

5 Conclusions and Future Work

Organizations, in their quest to improve and gain more and more value, have realized the enormous potential and impact of the idea of sustainability within their models and areas of the business. That is why the organizations are increasingly rethinking their way of interacting with the environment and have begun to act in this regard in the area of IT, implementing Green IT initiatives in their processes and daily operations. However, in this area of the Green IT organizations do not have any specific standards or frameworks to help them to implement, evaluate and improve the Green IT practices that they carry out. In order to overcome this obstacle, we have developed the "Governance and Management Framework for Green IT" and, in the present paper, we propose a SPICE-based maturity model for this framework, thanks to which it is intended to help to gradually implement new practices of Green IT in an organization, as well as to evaluate and improve the maturity level of Green IT of an organization.

In the first validations of the proposed maturity model carried out, we have managed to consolidate at theoretical and practical level the utility of this model for organizations in this area of Green IT. However, this is only a starting point and we will continue working in this area of Green IT, developing and improving through more validations both the "Governance and Management Framework for Green IT" and the maturity model proposed in this paper, making them into standards-compatible models. On the other hand, we also intend to bring the ISO 14000 family of standards [25] closer to Green IT, in order to identify those characteristics that can be integrated into the "Governance and Management Framework for Green IT", serving as a guide for those organizations that seek a certification in this standard.

Sustainability is a reality in all areas of knowledge and a fundamental aspect for life, so it is our duty to defend this idea, to protect the environment, and work towards a better and more sustainable future.

Acknowledgements. This work is part of the project GINSENG (TIN2015-70259-C2-1-R) funded by the Spanish Ministerio de Economía y Competitividad and the FEDER Fund (Fondo Europeo de Desarrollo Regional); and GLOBALIA (PEII-2014-038-P), Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha.

Appendix A: Processes of the "Governance and Management Framework for Green IT" Organized by the SPICE Maturity Levels

A.1 Level 1

BAI09: Manage assets

- **Description of the process:** Manage IT assets through their life cycle to make sure that their use delivers value at optimal cost, they remain operational (fit for purpose), they are accounted for and physically protected, and those assets that are critical to support service capability are reliable and available. Manage software licenses to ensure that the optimal number are acquired, retained and deployed in relation to required business usage, and the software installed is in compliance with license agreements.
- **Statement of the purpose of the process:** Account for all IT assets and optimize the value provided by these assets.

DSS01: Manage operations

- **Description of the process:** Co-ordinate and execute the activities and operational procedures required to deliver internal and outsourced IT services, including the execution of pre-defined standard operating procedures and the required monitoring activities.
- **Statement of the purpose of the process:** Deliver IT operational service outcomes as planned.

A.2 Level 2

APO01: Manage the IT management framework

- **Description of the process:** Clarify and maintain the governance of organization IT mission and vision. Implement and maintain mechanisms and authorities to manage information and the use of the organization IT in support of governance objectives in line with guiding principles and policies.
- **Statement of the purpose of the process:** Provide a consistent management approach to enable the organization governance requirements to be met, covering management processes, organizational structures, roles and responsibilities, reliable and repeatable activities, and skills and competencies.

APO02: Manage strategy

- **Description of the process:** Provide a holistic view of the current business and IT context, the future direction, and the initiatives required to migrate to the desired future context. Leverage organization architecture building blocks and components, including externally provided services and related capabilities to enable nimble, reliable and efficient response to strategic objectives.
- **Statement of the purpose of the process:** Align strategic IT plans with business objectives. Clearly communicate the objectives and associated accountabilities so they are understood by all, with the IT strategic options identified, structured and integrated with the business plans.

APO06: Manage budget and costs

- **Description of the process:** Manage the IT-related financial activities in both the business and IT functions, covering budget, cost and benefit management, and prioritization of spending through the use of formal budgeting practices and a fair and equitable system of allocating costs to the organization. Consult stakeholders to identify and control the total costs and benefits within the context of the IT strategic and tactical plans, and initiate corrective action where needed.
- **Statement of the purpose of the process:** Foster partnership between IT and organization stakeholders to enable the effective and efficient use of IT-related resources and provide transparency and accountability of the cost and business value of solutions and services. Enable the organization to make informed decisions regarding the use of IT solutions and services.

APO08: Manage relationships

- **Description of the process:** Manage the relationship between the business and IT in a formalized and transparent way that ensures a focus on achieving a common and shared goal of successful organization outcomes in support of strategic goals and within the constraint of budgets and risk tolerance. Base the relationship on mutual trust, using open and understandable terms and common language and a willingness to take ownership and accountability for key decisions.
- **Statement of the purpose of the process:** Create improved outcomes, increased confidence, trust in IT and effective use of resources.

BAI02: Manage requirements definition

- **Description of the process:** Identify solutions and analyze requirements before acquisition or creation to ensure that they are in line with organization strategic requirements covering business processes, applications, information/data, infrastructure and services. Co-ordinate with affected stakeholders the review of feasible options including relative costs and benefits, risk analysis, and approval of requirements and proposed solutions.
- **Statement of the purpose of the process:** Create feasible optimal solutions that meet organization needs while minimizing risk.

BAI03: Manage solutions identification and build

- **Description of the process:** Establish and maintain identified solutions in line with organization requirements covering design, development, procurement/sourcing and partnering with suppliers/vendors. Manage configuration, test preparation, testing, requirements management and maintenance of business processes, applications, information/data, infrastructure and services.
- **Statement of the purpose of the process:** Establish timely and cost-effective solutions capable of supporting organization strategic and operational objectives.

A.3 Level 3

EDM01: Ensure governance framework setting and maintenance

- **Description of the process:** Analyze and articulate the requirements for the IT governance of the organization, and put in place and maintain effective enabling structures, principles, processes and practices, with clarity of responsibilities and authority to achieve the organization's mission, goals and objectives.
- **Statement of the purpose of the process:** Provide a consistent approach integrated and aligned with the organization governance approach. To ensure that IT-related decisions are made in line with the organization's strategies and objectives, ensure that IT-related processes are overseen effectively and transparently, compliance with legal and regulatory requirements is confirmed, and the governance requirements for board members are met.

EDM02: Ensure benefits delivery

- **Description of the process:** Optimize the value contribution to the business from the business processes, IT services and IT assets resulting from investments made by IT at acceptable costs.
- **Statement of the purpose of the process:** Secure optimal value from IT-enabled initiatives, services and assets; cost-efficient delivery of solutions and services; and a reliable and accurate picture of costs and likely benefits so that business needs are supported effectively and efficiently.

EDM05: Ensure stakeholder transparency

- **Description of the process:** Ensure that organization IT performance and conformance measurement and reporting are transparent, with stakeholders approving the goals and metrics and the necessary remedial actions.
- **Statement of the purpose of the process:** Make sure that the communication to stakeholders is effective and timely and the basis for reporting is established to increase performance, identify areas for improvement, and confirm that IT-related objectives and strategies are in line with the strategy of the organization.

MEA03: Monitor, evaluate and assess compliance with external requirements

- **Description of the process:** Evaluate that IT processes and IT-supported business processes are compliant with laws, regulations and contractual requirements. Obtain assurance that the requirements have been identified and complied with, and integrate IT compliance with overall organization compliance.

- **Statement of the purpose of the process:** Ensure that the organization is compliant with all applicable external requirements.

A.4 Level 4
MEA01: Monitor, evaluate and assess performance and conformance

- **Description of the process:** Collect, validate and evaluate business, IT and process goals and metrics. Monitor that processes are performing against agreed-on performance and conformance goals and metrics and provide reporting that is systematic and timely.
- **Statement of the purpose of the process:** Provide transparency of performance and conformance and drive achievement of goals.

A.5 Level 5
EDM03: Ensure risk optimization

- **Description of the process:** Ensure that the organization's risk appetite and tolerance are understood, articulated and communicated, and that risk to organization value related to the use of IT is identified and managed.
- **Statement of the purpose of the process:** Ensure that IT-related organization risk does not exceed risk appetite and risk tolerance, the impact of IT risk to organization value is identified and managed, and the potential for compliance failures is minimized.

EDM04: Ensure resource optimization

- **Description of the process:** Ensure that adequate and sufficient IT-related capabilities (people, process and technology) are available to support organization objectives effectively at optimal cost.
- **Statement of the purpose of the process:** Ensure that the resource needs of the organization are met in the optimal manner, IT costs are optimized, and there is an increased likelihood of benefit realization and readiness for future change.

References

1. Brundtland, G., Khalid, M., Agnelli, S., Al-Athel, S., Chidzero, B., Fadika, L., Okita, S.: Our Common Future ("Brundtland Report"). Oxford University Press (1987)
2. Calero, C., Piattini, M. (eds.): Green in Software Engineering. Springer, Cham (2015)
3. Patón-Romero, J.D., Piattini, M.: Indicators for Green in IT audits: a systematic mapping study. In: 3rd International Workshop on Measurement and Metrics for Green and Sustainable Software Systems (MeGSuS 216), pp. 4–12 (2016)
4. Patón-Romero, J.D., Piattini, M., García Rodríguez de Guzmán, I.: A Governance and Management Framework for Green IT. Sustainability (Submitted)
5. ISO: ISO/IEC 15504 (Information technology — Process assessment). International Organization for Standardization, Geneva, Switzerland (2003)
6. Cass, A., Volcker, C., Winzer, L., Carranza, J.M., Dorling, A.: SPiCE for SPACE: a process assessment and improvement method for space software development. ESA Bull. **107**, 112–119 (2001)

7. Garzás, J., Pino, F.J., Piattini, M., Fernández, C.M.: A maturity model for the Spanish software industry based on ISO standards. Comput. Stand. Interfaces **35**(6), 616–628 (2013)
8. Gökalp, E., Demirörs, O.: Towards a process capability assessment model for government domain. In: Clarke, Paul M., O'Connor, Rory V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 210–224. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_16
9. Ivanyos, J., Sándor-Kriszt, E.: ECQA Governance SPICE assessor skills for evaluating integrated risk management scenarios. J. Softw. Evol. Process **27**(8), 545–554 (2015)
10. Automotive SIG: Automotive SPICE Process Assessment/ Reference Model. Version 3.0. VDA Quality Management Center, Berlin, Germany (2015)
11. Lami, G., Biscoglio, I., Falcini, F.: An empirical study on software testing practices in automotive. In: Clarke, Paul M., O'Connor, Rory V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 301–315. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_22
12. Mesquida, A.L., Mas, A.: Implementing information security best practices on software lifecycle processes: the ISO/IEC 15504 security extension. Comput. Secur. **48**, 19–34 (2015)
13. McCaffery, F., Dorling, A.: Medi SPICE development. J. Softw. Evol. Process **22**(4), 255–268 (2010)
14. Söylemez, M., Tarhan, A.: The use of maturity/capability frameworks for healthcare process assessment and improvement. In: Clarke, Paul M., O'Connor, Rory V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 31–42. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_3
15. Varkoi, T., Nevalainen, R., Mäkinen, T.: Toward nuclear SPICE - Integrating IEC 61508, IEC 60880 and SPICE. J. Softw. Evol. Process **26**(3), 357–365 (2014)
16. ISO: ISO/IEC 33000 (Information technology — Process assessment). International Organization for Standardization, Geneva, Switzerland (2015)
17. Carretero, A.G., Gualo, F., Caballero, I., Piattini, M.: MAMD 2.0: Environment for data quality processes implantation based on ISO 8000-6X and ISO/IEC 33000. Comput. Stand. Interfaces (2016)
18. Carnegie Mellon University: CMMI for Development (Version 1.3). Carnegie Mellon University, Pittsburgh, PA, USA (2010)
19. ISACA: Maximizing the Combined Effects of COBIT 5 and CMMI: A Guide to Using the Practices Pathways Tool. ISACA, Rolling Meadows, IL, USA (2017)
20. ISACA: COBIT 5: A Business Framework for the Governance and Management of Enterprise IT. ISACA, Rolling Meadows, IL, USA (2012)
21. Patón-Romero, J.D., Piattini, M.: Green IT maturity models: a systematic mapping study. In: 12th Iberian Conference on Information Systems and Technologies (CISTI 2017), pp. 2110–2115 (2017)
22. Buchalcevova, A.: Green ICT maturity model for Czech SMEs. J. Syst. Integr. **6**(1), 24–36 (2015)
23. Hankel, A., Oud, L., Saan, M., Lago, P.: A Maturity Model for Green ICT: The case of the SURF Green ICT Maturity Model. In: 28th EnviroInfo 2014 Conference (2014)
24. Curley, M., Kenneally, J., Carcary, M.: IT Capability Maturity Framework (IT-CMF). In: The Body of Knowledge Guide, 2nd edn. Van Haren Publishing, Zaltbommel, pp. 103–118 (2016)
25. ISO: ISO 14000 (Environmental management systems). International Organization for Standardization, Geneva, Switzerland (2015)

A Multi-layer Representation Model for the ISO/IEC 33000 Assessment Framework: Analysing Composition and Behaviour

Alvaro Fernández Del Carpio(✉)

Software Engineering Department, Universidad La Salle,
Alfonso Ugarte Avenue 517, Arequipa, Peru
alfernandez@ulasalle.edu.pe

Abstract. Software Process Improvement (SPI) models are a very important topic for SPI workers as software engineering students that require a good comprehension of the process assessment models. This paper proposes a visual multi-layered representational model describing in a highly practical way the ISO/IEC 33000 Assessment Framework through of structural and behavioural views. This kind of representation comprises several semantic layers based on the dimensions of the key elements of the model with an additional dimension for specifying measurements. The structural elements of the model are assigned on distinct layers and connected through dependence and co-occurrence connector in applying the Assessment Model on software processes. The representation model was tested with software developers, academic experts and students.

Keywords: ISO/IEC 33000 · ISO/IEC 15504 · SPICE · Software process improvement · SPI · Knowledge representation

1 Introduction

Software Process Improvement (SPI) is considered an important goal for software firms to produce high quality software products [1]. The software process model selected for this research was the ISO/IEC 33000 as an upgrade of the well-known ISO/IEC 15504 standard used by many software organizations around the world. However, the development of software in some regions is not yet carried out under SPI umbrella. Concretely, in the south region of Perú, just a few companies have knowledge and apply in their projects the ISO/IEC 15504 or 33000 series. The main interest in this research is promoting and strengthening the incorporation and use of this standard within software organizations.

A well-formed knowledge representation and an effective transference of knowledge is critical for software teams [2, 3], becoming a strategic area of knowledge management for practitioners, researchers [4], and organizations [2, 5]. An effective sharing of know-hows and feedbacks regarding processes influence the characteristics of the software product [6].

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 156–169, 2017.
DOI: 10.1007/978-3-319-67383-7_12

Although some software tools provide mechanisms to visualize software processes and their measurements according to maturity levels, a multilayer-based representation bringing out an intrinsic graph of both Process Reference and Assessment Models could be more appropriated to achieve a better insight about their composition, the way as instances of processes are carried out by SPI practitioners, and the dynamic of measurements over processes. As graphs are structures widely used to represent the knowledge [7], a kind of representation based on layered graphs for a software process model would provide more insight in performing, monitoring and analysing the model. Likewise, the kind of representation proposed in this research would enable the user to get useful information from ISO/IEC 33000 models by facilitating rapid reflections and continuous introspections into them (e.g. when exploring aspects of composition and behaviour of processes and measurements). This approach would close the gap of weak knowledge about the standard that some SPI practitioners could have on it.

The paper is organized as follows. First, a background presents the main concepts and research works underlying the proposed model. Next, the proposal is described and explained. Then, the results obtained for validating the proposal in organizations are presented. The next section describes some threats to validity of the work. Finally, conclusions and future work are presented.

2 Background

The Software Process Improvement Capability dEetermination (SPICE) includes the well-known ISO/IEC 15504 as a model applied in many organizations for improving and evaluating processes, system information and software products maintenance [8, 9] and the recent ISO/IEC 33000 series replacing the previous series 15504. SPICE has been customized for different application domains such as Automotive SPICE, InnoSPICE, SPICE4SPACE, Enterprise SPICE, etc. [10]. SPICE assessment is usually performed as a part of a process improvement initiative and/or a capability determination approach [11].

The ISO/IEC 33000 family emerges to assess process quality characteristics. This standard contains as key elements of the model: (1) The Process Reference Model (PRM) which is composed of a set of inter-related processes, (2) The Process Assessment Model (PAM) for assessing quality characteristics of processes, and (3) The Measurement Framework (MF) for measuring the process quality characteristics of capability [12]. The achievement of the capability level of a process instance is expressed in terms of a software process attributes (SPA) [13]. Since SPA involves a subjective measurement procedure, the reliability of this procedure is vital in order to have confidence in the assessment results. A construct is considered as a conceptual object that is neither directly measurable nor observable. Individual ratings of measurable items are summed up and an overall score is obtained. Actual variation across the organizations in capability and the error component are factors that act upon the variability in a set of item scores [14].

The representation of the outcomes achievements achieved by each practice can fall in assessing multiple process instances. Hence, [15] represented it as two path diagrams of measurement models, where constructs are visualized as ovals, observed measures as rectangles, causal paths as single-headed arrows, and correlations as double-headed arrows. Some principles of composite measures and the definition of types of aggregation methods to develop composite measures were specified in order to get a better comprehension of process measurement frameworks of process quality characteristics [16]. Scale types were reviewed and used the reflective and formative measurements models to establish relationships between process attributes and practices, proposing composite measure development methods.

On the other hand, forms of representing knowledge are addressed to a hierarchy of data, rule-based representations and logic-based representations [17]. A graph-based representation expresses knowledge on a logical basis and in a structured way, showing how the knowledge is built, and allowing the control over the formation process [18], facilitating an easy understanding by users. Graphs-based knowledge representation enables knowledge-based reasoning, allowing handling uncertainty through fuzzy technology [19], such as fuzzy Petri nets [20, 21] and building nodes of knowledge for displaying types of human knowledge and detecting new knowledge with grouping terms into complex expressions [22]. Likewise, a RDF-based representation method is used to model the knowledge when developing innovative design [23]. Graphs considering properties about the spatial and temporal representations are termed spatio-temporal graphs. They include nodes as spatio-temporal objects and links defining the spatio-temporal relationships such as: spatio, temporal and spatio-temporal. A spatio-temporal object can be represented as tuple (I, G, T, A), where I describes the id of the object, G is the localization, T represents the temporal characteristics, and A is a set of attributes describing others properties [24].

Some research works have been focused on clarifying the structuration and outcome measurements through models depicting constructs, process instances and relationships between all them. However, representations of the model as structured layers made up of groups of functional aspects and as a mean for extending the representation of the ISO/IEC 33000 models in order to get different kind of views has not been found in the literature.

3 The Representation Model

Software Process Improvement (SPI) has become an important referent in many software organizations for keeping up a continuous assessment and improvement of their processes and practices, increasing the efficiency and effectiveness of a software development in order to fulfill the customers and stakeholders requirements of the organization [25, 26]. Considering that SPI is addressed to develop products with quality, this research proposes a representation scheme of the ISO/IEC 33000 Assessment Framework as a multidimensional layer-based model in order to facilitate the understanding and to improve skills about structuration and applicability of key elements of the model as a relevant concern for novel users of the standard. The reference model architecture consists of the process dimension and the process capability dimension. With a multi-layered representation model, coupling the dimensions' components and binding to some others structural elements, both static and dynamic views of the model can be obtained. A set of viewpoints provides the views of the model, which fall into the following types: composition (how the model is structured), communication (behaviour and interconnection among the elements), and assessment of processes.

3.1 Static View

This view describes the structure of both Process Reference Model and Process Assessment Model on a multi-layer architecture (see Fig. 1). Due to space restrictions in the document, a simple example taking a small part of the ISO/IEC 12207:2016 PRM is represented with the static view specification.

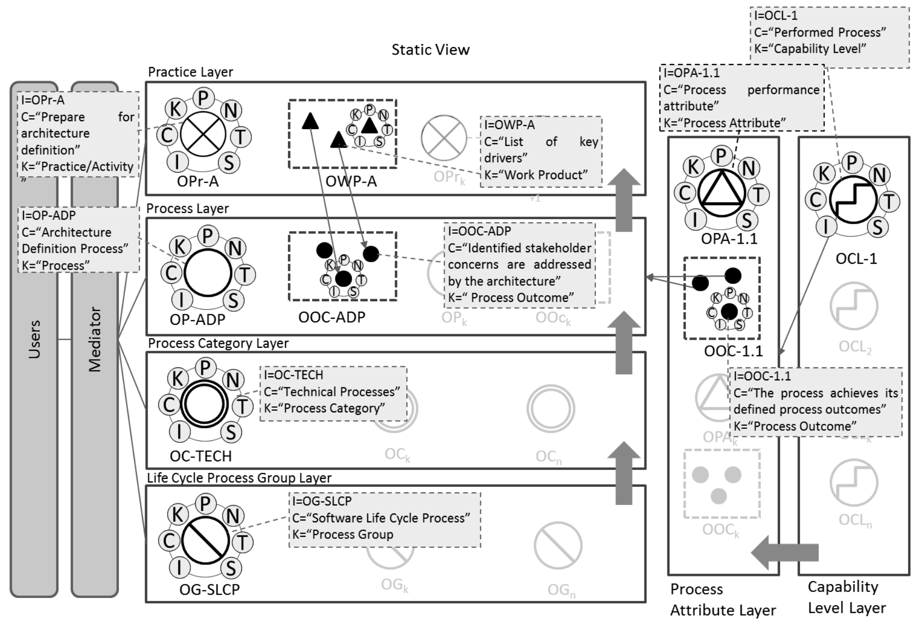

Fig. 1. A static view of the model represented as a multidimensional layer-based architecture

The two dimensions of the process assessment model, process and capability dimensions, are broken down and deployed on several layers trying to maintain the semantic integrity. Each layer lays down a specific role and represents an abstraction of the model elements or components, describing a type-based organization and associations among them. Therefore, a descriptive visualization of the model composition can be obtained.

The layers concerning the Process Reference Model are organized as a bottom-up association, and layers regarding the Assessment Model have a transversal orientation in order to relate process and practice outcomes with process attributes. Elements of the view fall into the following dimensions:

a. **Process dimension.** This dimension encompasses the following layers:

- Life cycle process group layer: it contains components related to the groups of life cycle processes.
- Process category layer: it has components related to the process categories.
- Process layer: it contains components related to processes of the domain.
- Practice layer: it contains components related to practices performed to produce work products.

b. **Capability dimension**

- Capability level layer: it contains components related to capability levels.
- Process attributes layer: it contains components related to process attributes.

c. **Composition dimension**

- User layer: it contains users on charge of some parts of the model.
- Mediator layer: it contains the mechanisms for connecting the different processes and practices and gets some level of orchestration between them.

Components of each layer are represented as spatio-temporal objects (Table 1) which can be described as tuple (*I, K, T, A*), where *I* is the Identifier, *K* is the type (e.g. group, category, process, practice, process attribute, capability level, and measurement), *T* is the temporal characteristic established as a discrete time, and *A* represented a set of others properties (Caption(C), Purpose (P), Notes (N), and Status of the component (S: halt, execution, finish)). Properties are annexed to components as options to be configured by user, such as is shown in Fig. 1 and where a part of these properties is shown with the example.

Table 1. Types of nodes

Symbol	Description
◎	Life Cycle Process Group (OG)
⃠	Process Category (OC)
○	Process (OP)
⊗	Practice (OPr)
	Capability Level (OCL)
▲	Work products (OWP)
●	Outcome of process (OOC)

Table 2. Types of connectors

Symbol	Description
⟶	Dependency between components
- - - ->	Co-occurrence between components

Associations between components are defined by two types of connectors (see Table 2): (a) Dependency between the components, and (b) Co-occurrence of components (e.g. parallel execution of practices).

The inter-connection between node objects generates several sets of subjacent graphs which encompass the Multi-dimensional Graph (MG), defined as MG = {*subOGG*(), *subOCG*(), *subPG*(), *subOCLG*(), *subOPAG*()} (i, j, k ϵ N), where:

- *subOGG* = ({OC_i}, {E_k}), where {OC_i} is a set of Process Category nodes linked to a Process Group node through a set of connectors {E_k}.
- *subOCG* = {(OP_i, {OOC_j}), {E_k}}, a set of tuples comprised of a Process node OP_i related to a set of Process Outcome nodes {OOC_j}. Each tuple is linked to a Process Category node through a set of connectors {E_k}.
- *subPG* = {(OPr_i, {OWP_j}), {E_k}}, a set of tuples comprised of a Practice node OPr related to a set of Work Product nodes {OWP_j}. Each tuple is linked to a Process node through a set of connectors {E_k}.
- *subOCLG* = {(OPA_i, {OOC_j}), {E_k}}, a set of tuples comprised of a Process Attribute node OPA_i related to a set of Outcome nodes {OOC_j}. Each tuple is linked to a Capability Level node through a set of connectors {E_k}.
- *subOPAG* = {(OP_i, {OOC_j}), {E_k}}, a set of tuples composed by Process node OP_i related to a set of Outcome nodes {OOC_j}. Each tuple is linked to a Process Attribute node through a set of connectors {E_k}.

3.2 Dynamic View

It describes the dynamic aspects by performing processes and how the outcomes are assessed through process attributes and assessment indicators. This view is intended to show the progression of executing process instances diagrammed as nodes interconnected. Assessment elements for confirming the realization of practices based on evidence coming from work products of the assessed process are established into a new measurement layer (see Fig. 2).

(a) **Measurement Layer**

This layer contains the elements for measuring the process results. Instances of measurements are created as node objects (see Table 3) and associated with a set of results that should be accomplished according to a Process Attribute instance. Each node contains a measurement value (χ). This layer supports both the reflective and formative measurements models [27] at specifying the links between the construct and the measurements.

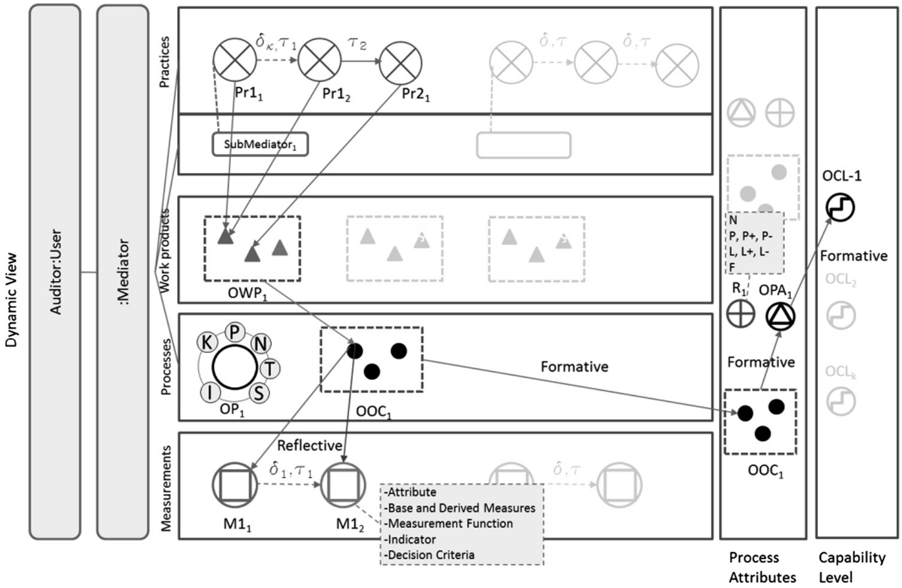

Fig. 2. A representation of the dynamic behaviour of the model

Table 3. Type of node for measurements

Symbol	Description
	Measurement

(b) **Mechanisms of interaction**

When a Capability Level is specified for assessing a set of processes, the corresponding Process Attributes together with a set of outcomes that should be achieved are instanced into the Process Attribute Layer. Performing practice instances produce Work Products which contribute to the achievement of the Process Outcomes. The progression development mainly in processes, practices and measurements are depicted through dependency and co-occurrence connectors. The difference of development between instances is marked with value δ, and time variations are indicated with the value τ. All the measurement instances applied on Process Outcomes are specified into the Measurement Layer. According to assessments, results are depicted either in red (incomplete) or in green (complete). Status of nodes is updated as the development is carrying on, and development time is registered into the time information element. Thus, aspects of temporality can be defined into the model. Results of evaluating an Attribute Process is specified by the Rating node (see Table 4). This kind of node depicts the rating scale achieved by a Process Attribute (N, P, L, F plus the refined measures P+, P−, L+, and L−).

Table 4. Type of node for ratings

Symbol	Description
⊕	Rating

The development of processes and practices are orchestrated through a mediator component according to the invocation sequence. The mediator component also establishes links between users and processes associated with practices in order to get traceability through all the components connected.

Each node of the model includes a set of information elements to be reviewed by the user.

(c) **Types of viewpoints**
A set of viewpoints can be developed from views in order to provide the conventions of representing forms of elaborating them. The most highlighted viewpoints are:

- Composition: depicts the structure of the model based on structural elements grouped by layers and connectors.
- Communication: depicts the execution of the process and practices with the generation of work products and outcomes. Information visualized is characterized by user-processes flow, the development progression of processes and practices, the organization of developing processes and practices by the mediator at operation time, and traceability between the different kinds of instances.
- Assessment: depicts the relation between outcomes, measurements and process attributes occurred over time. Information visualized is the progression of measurements on process outcomes and achievement variations between expected and observed outcomes measured.

4 Case Study

This research was evaluated through surveys by distributing questionnaires to relevant participants. This study was conducted on academic and software industry environments, represented by software engineering students belonging to Universidad Católica de Santa María (UCSM) and Universidad La Salle (ULS), and software staff working at Tata Consulting and Microdata S.R.L., considered as outstanding companies dedicated to build software solutions. The representation model was applied for performing and assessing software processes and getting different views of analysis about the structural elements composition and monitoring of its behaviour in basis on several components instances.

The items in the questionnaire measure the perceptions about the structural and dynamic representation for analysing the composition and behaviour of the structural elements of the model based on a five point Likert type scale where 1 = Strongly Disagree and 5 = Strongly Agree. A total of 57 members from different universities and software industry sector participated in this process. Software projects were previously selected to serve as sources for the purposes of validating the model, determining the group of processes, practices and work documents for development. Participants working in software companies were selected explicitly for having experience and knowledge on developing projects focused on SPI and using ISO/IEC 15504 or ISO/IEC 33000 standards. Several meetings held with all participants were addressed to explain the representational model, pointing out the relationships between dimensions of the standard and elements of the model. Indications were addressed about how the assessment process is represented and how the model should be used. On the other hand, software engineering students were asked to do some exercises on assessment activities according to the standard and following the proposed model.

The questionnaire consisted of two sections for evaluating the static and dynamic views of the representational model. Questions of the first section were related to: Q1-Suitable separation of the structural elements of the Reference and Evaluation Model, Q2-Dimensions of the Representational Model are clearly identified, Q3-The structural elements of the model are sufficiently representative, Q4-Connections among structural elements form a hierarchical representation and build an intrinsic level graph expressing precisely the semantics of the model, Q5-Notations of node types are clearly understood, Q6-Information attributes of nodes are enough representatives, and Q7-Dependency and co-occurrence connections express the relations between the structural elements. On the other hand, questions of the second section were related to: Q8-Instances of processes and practices represent the development progression over time through dependence and co-occurrence connections, Q9-The model shows how practices create dynamically work products, Q10-Work products are associated with process results, specifying the realization time and the integration between them, Q11-Measurements are performed on process results through measurement instances, Q12-Behaviour is visualized through measurement instances with values and variations between them, Q13-Suitable use of colour codes for representing the achievement status of structural elements (processes, practices, measurements), Q14-Linking of processes and practices with users in charge of them, Q15-Sequence of processes and practices by the mediator component according to the dependence and co-occurrence that must exist between them, and Q16-Identification of traceability between the different structural elements. Moreover, some additional questions regarding complementary information of the model were also formulated.

Data obtained was analysed using the descriptive statistical analysis. Reliability analysis was employed to verify the reliability of the measurement construction.

Results and Analysis in the Software Industry Sector

15 responses were received from SPI workers belonging to software industry companies. The participants were identifies by their managers as relevant SPI practitioners. Tables 5 and 6 depict the reliability of the measures construct as well as a descriptive analysis of the answers with respect to static view.

The reliability analysis of Cronbach's alpha carried out on the perceived values scale comprising 7 items revealed an adequate reliability of the questionnaire, $\alpha = 0.719$.

Table 5. Reliability statistics for static view in software industry sector

Cronbach's alpha	Cronbach's alpha based on standardized items	N. of items
,719	,732	7

Table 6. Descriptive statistics for static view in software industry sector

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Mean	4,067	3,733	4,133	4,000	4,067	4,000	4,000
Median	4,000	4,000	4,000	4,000	4,000	4,000	4,000
Mode	4,00	4,00	4,00	4,00	4,00	4,00	4,00
Std. deviation	,704	,704	,640	,929	,799	,655	,756
Minimum	3,00	3,00	3,00	2,00	3,00	3,00	3,00
Maximum	5,00	5,00	5,00	5,00	5,00	5,00	5,00

Most items appeared to be worthy of retention, resulting in a decrease in the alpha if deleted, but in the case of item 7, the alpha would increase to $\alpha = 0.750$.

Tables 7 and 8 depict the reliability of the measures construct as well as a descriptive analysis of the answers with respect to dynamic view.

The reliability analysis comprising 9 items showed the questionnaire indicates an acceptable reliability, $\alpha = 0.711$. Most items appeared to be worthy of retention, resulting in a decrease in the alpha if deleted, but in item 14 the alpha would increase to $\alpha = 0.720$.

Table 7. Reliability statistics for dynamic view in software industry sector

Cronbach's alpha	Cronbach's alpha based on standardized items	N. of items
,711	,718	9

Table 8. Descriptive statistics for dynamic view in software industry sector

	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Mean	3,867	4,067	4,000	3,800	3,800	4,000	3,600	3,800	4,000
Median	4,000	4,000	4,000	4,000	4,000	4,000	3,000	4,000	4,000
Mode	4,00	4,00	4,00	4,00	4,00	4,00	3,00	4,00	4,00
Std. deviation	,516	,704	,756	,777	,775	,926	,737	,676	,756
Minimum	3,00	3,00	3,00	2,00	2,00	2,00	3,00	2,00	3,00
Maximum	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00

Results and Analysis in the Academic Sector

42 responses were received from software students with knowledge in SPI models. Many of them are also working in software development projects. Tables 9 and 10 depict the reliability of the measures construct as well as a descriptive analysis of the answers with respect to static view.

Table 9. Reliability Statistics for Static view in software academic sector

Cronbach's alpha	Cronbach's alpha based on standardized items	N. of items
,703	,700	7

Table 10. Descriptive statistics for static view in software academic sector

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Mean	3,9286	3,6905	3,9524	3,8333	3,6905	3,8571	3,9524
Median	4,0000	4,0000	4,0000	4,0000	4,0000	4,0000	4,0000
Mode	4,00	3,00	4,00	4,00	4,00	4,00	4,00
Std. deviation	,71202	,74860	,73093	,65951	,74860	,87154	,73093
Minimum	3,00	3,00	2,00	2,00	2,00	2,00	3,00
Maximum	5,00	5,00	5,00	5,00	5,00	5,00	5,00

The reliability analysis of Cronbach's alpha carried out on the perceived task values scale comprising 7 items showed the questionnaire indicates adequate reliability, $\alpha = 0.703$. Most items appeared to be worthy of retention, resulting in a decrease in the alpha if deleted, except in item 7, which would increase the alpha to $\alpha = 0.725$.

Tables 11 and 12 depict the reliability of the measures construct as well as a descriptive analysis of the answers with respect to dynamic view.

The reliability analysis comprising 9 items showed the questionnaire indicates adequate reliability, $\alpha = 0.725$. Most items appeared to be worthy of retention, resulting in a decrease in the alpha if deleted, but in item 12 the alpha would increase to $\alpha = 0.759$.

Table 11. Reliability Statistics for dynamic view in software academic sector

Cronbach's alpha	Cronbach's alpha based on standardized items	N. of items
,725	,724	9

Table 12. Descriptive statistics for dynamic view in software academic sector

	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Mean	3,762	3,929	3,881	3,809	3,714	4,024	3,881	3,762	3,857
Median	4,000	4,000	4,000	4,000	4,000	4,500	4,000	4,000	4,000
Mode	4,00	4,00	4,00	4,00	3,00[a]	5,00	4,00	4,00	5,00
Std. Deviation	,759	,997	,633	,6713	,774	1,137	,772	,850	1,049
Minimum	2,00	1,00	3,00	2,00	2,00	1,00	2,00	2,00	1,00
Maximum	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00

[a]Multiple modes exist. The smallest value is shown

5 Threats to Validity

The internal validity of this study was limited by the fact that some participants had different previous knowledge about the ISO/IEC 33000 standard due to experiences in analysing and/or developing projects oriented to SPI. Some kind of lack of knowledge about the ISO/IEC 33000 was compensated for becoming more familiar with the ISO/IEC 15504 standard. Some participants showed different expectations for attending in the research validation process regarding their personal interest in knowing the benefits of the proposed model for their work and self-learning about the standard.

On the other hand, concerning the external validity, the selection process of software development companies applying the SPICE standard was a little complicated, just a few of them are using the standard for assessing their software processes in Arequipa city. However, the chosen companies were the most representatives in the city where this research was carried out. To mitigate these threats and as future works, the representation model of the ISO/IEC 33000 should be evaluated in other regions (e.g. Lima city) where is feasible to find more software companies using this SPI standard.

6 Conclusions

This research provides a specific kind of representation for the ISO/IEC 33000 models based on layered structure and graphs in order to achieve a more flexible and adaptable representation model. It would allow both practitioners and learners of this standard to get a better comprehension of its structure, making a tracking of processes and measurements over time, and getting insight on aspects related to its applicability in assessing processes. It is intended that using this form of representation the knowledge on conceptual and dynamic aspects would be easily identified when SPI practitioners apply the standard.

As some of the benefits expected to get an enhancement on maintainability of structural elements through additions, changes or updates to layers, and serve as a mean of communication between SPI workers themselves. However, the independence aspect featuring a layered model is constrained in this case since that is not complete. Changes at bottom layers of the model would affect the subsequent ones but not necessarily on the contrary. The same criteria can be applied for transversal layers. Namely, some changes into a layer can cascade with other layers.

Some recommendations gathered from the review process mention that it is necessary to clarify a bit more the graphical notations in the model since some parts look a bit confusing. There were certain difficulties in carrying out measurements on processes by software workers since this task required a continuous register of data. It is required more provision of guiding examples showing different cases of applying the model. Finally, the characteristics of the dimensions should be a bit more pinned down.

For most participants, the representational model helped them to get a better understanding of the structure and behaviour of the process and assessment models of the ISO/IEC 33000 (with 55,3% agree and 27,7% strongly agree). The 95,7% of participants would use the representational model for analyzing the compositions and behaviours of the standard at assessing software processes.

As future works, improvements on the issues identified will be incorporated into the model as well as the inclusion of spatio-temporal characteristics such as the interrelation of containment or overlapping between elements of the model.

References

1. Almomani, M.A.T., Basri, S., Mahamad, S., Bajeh, A.O.: Software process improvement initiatives in small and medium firms: a systematic review. In: 3rd International Conference on Advanced Computer Science Applications and Technologies (ACSAT), pp. 162–167. IEEE (2014)
2. Xiao-hong, W., Bao-sheng, Z., Wen-jing, W.: Research on stability of knowledge transfer in virtual technology innovation team. In: International Conference on Management Science and Engineering (ICMSE), pp. 969–975. IEEE (2010)
3. Xiao-na, B., Gang, Q., Guo-liang, Z.: An empirical study of the relationship between team social capital and knowledge transfer: mediating role of transactive memory system. In: International Conference on Management Science and Engineering (ICMSE), pp. 1370–1378. IEEE (2013)
4. Hongli, L., Yao, F., Zhigao, C.: Effects of social network on knowledge transfer within R&D team. In: International Conference on Information Management, Innovation Management and Industrial Engineering, vol. 3, pp. 158–162. IEEE (2009)
5. Chen, T., Fu, H.: The subject knowledge representation and utilizations in e-learning. In: 2nd International Symposium on Information Engineering and Electronic Commerce (IEEC), pp. 1–4. IEEE (2010)
6. Ghobadi, S.: What drives knowledge sharing in software development teams: a literature review and classification framework. Inf. Manag. **52**(1), 82–97 (2015)
7. Fionda, V., Pirro, G.: Querying graphs with preferences. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 929–938 (2013)
8. Gökalp, E., Demirörs, O.: Developing process definition for financial and physical resource management process in government domain. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 169–180. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_13
9. ISO: ISO/IEC 15504-1:2004, Information technology — Process assessment — Part 1: Concepts and vocabulary, ISO/IEC JTC1 (2004)
10. Lami, G., Fabbrini, F., Buglione, L.: An ISO/IEC 33000-compliant measurement framework for software process sustainability assessment. In: Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 50–59. IEEE (2014)
11. ISO: ISO/IEC 15504-4, Information Technology — Process Assessment — Part 4: Guidance on Use for Process Improvement and Process Capability Determination, ISO (2004)
12. ISO: ISO/IEC 33000, Information technology: process assessment, ISO (2015)
13. ISO: ISO/IEC 33020, Information technology: process assessment: process measurement framework for assessment of process capability. ISO (2015)
14. Jung, H.W., Hunter, R.: Evaluating the SPICE rating scale with regard to the internal consistency of capability measures. Softw. Process: Improv. Pract. **8**(3), 169–178 (2003)
15. Jung, H.W.: Investigating measurement scales and aggregation methods in SPICE assessment method. Inf. Softw. Technol. **55**(8), 1450–1461 (2013)

16. Jung, H.-W., Varkoi, T., McBride, T.: Constructing process measurement scales using the ISO/IEC 330xx family of standards. In: Mitasiunas, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2014. CCIS, vol. 477, pp. 1–11. Springer, Cham (2014). doi:10.1007/978-3-319-13036-1_1
17. Mohamed, R., Watada, J.: Evidence theory based knowledge representation. In: Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services, pp. 74–81. ACM (2011)
18. Portmanna, E., Kaltenriedera, P., Pedryczb, W.: Knowledge representation through graphs. Procedia Comput. Sci. **62**, 245–248 (2015)
19. Shetty, R.T., Riccio, P.M., Quinqueton, J.: Hybrid model for knowledge representation. In: International Conference on Hybrid Information Technology, ICHIT 2006, vol. 1, pp. 355–361. IEEE (2006)
20. Ribarić, S., Zadrija, V.: An object-oriented implementation of a knowledge representation scheme based on fuzzy Petri nets. In: Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), vol. 2, pp. 987–993. IEEE (2010)
21. Suraj, Z.: Knowledge representation and reasoning based on generalised fuzzy Petri nets. In: 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 101–106. IEEE (2012)
22. Jakupovic, A., Pavlic, M., Mestrovic, A., Jovanovic, V.: Comparison of the nodes of knowledge method with other graphical methods for knowledge representation. In: 36th International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), pp. 1004–1008. IEEE (2013)
23. Zhen, L., Jiang, Z., Su, H., Liang, J.: RDF-based innovative design knowledge represent. In: First International Conference on Semantics, Knowledge and Grid, SKG 2005, pp. 77–77. IEEE (2005)
24. Besembel, I., Montilva, J.: Modelling spatio-temporal relationships in object oriented applications. Sci. Eng. **21**(2), 19–23 (2000)
25. Unterkalmsteiner, M., Gorschek, T., Islam, A.K., Cheng, C.K., Permadi, R.B., Feldt, R.: Evaluation and measurement of software process improvement – a systematic literature review. IEEE Trans. Softw. Eng. **38**, 398–424 (2012)
26. SEI: CMMI for Development, Version 1.3, Bedford, Hanscom AFB (2010)
27. ISO: ISO/IEC 33003, Information Technology—Process Assessment—Requirements for Process Measurement Frameworks. ISO (2015)

SPI and Models

Applying Agent-Based Simulation to the Improvement of Agile Software Management

Nuria Hurtado(✉), Mercedes Ruiz, Cristina Capitas, and Elena Orta

Department of Computer Science and Engineering, University of Cadiz (Spain), C/Chile Nº1, 11003 Cadiz, Spain
{nuria.hurtado,mercedes.ruiz,elena.orta}@uca.es, cristinacapitassanchez@mail.uca.es

Abstract. Among agile methodologies, eXtreme Programming (XP) is one of the best known and better defined. However, one factor that hinders its application is the lack of native XP support for project management. One of the techniques that could help in the improvement of XP projects management is the simulation modeling. In this paper, we examine, through a literature review, the evidences of the application of modeling and simulation techniques to support the management in XP projects. From this review we conclude that there is still work to be done in this area, and more specifically in the teamwork management, having in mind that agile team management is the most influential factor in achieving agile team productivity. As a proof of concept, we present Sim-Xperience: a simulation model to assist the XP team in the management of their projects; this model, unlike those found in the literature, has been developed following the agent-based paradigm, especially suited to simulate social behaviors. Through the model input parameters you can configure the specific features of the project you want to simulate and of the development team. Thus, the model allows you to analyze the effect of different decisions on team management process, observing the evolution of the project development as well as the deviations in comparison with initial estimations. To illustrate the model simulation we have conducted a case study, where we have seen the results of the simulation model under two different allocation tasks strategies, concluding that using a strategy where the team member experience is not the priority criterion is better for the increase of team experience in the long term.

Keywords: Agent-based simulation · Agile development · eXtreme programming · Teamwork management

1 Introduction

Agile software development methodologies are gaining interest among academic researchers and practitioners. These methods arise as a community reaction to the traditional, plan-based approaches to software engineering that have dominated the world of software engineering for years [1]

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 173–186, 2017.
DOI: 10.1007/978-3-319-67383-7_13

There are numerous references in the literature about the advantages of agile methods, with its emphasis on individuals and interactions over processes and tools, on working software over comprehensive documentation, on customer collaboration over contracts and formal negotiations, and on responding to changes instead of sticking to rigid planning [2, 3]. One of the best known and most applied agile methods is eXtreme Programming (XP). XP is driven by a set of shared values that include simplicity, communication, feedback, and courage [4–6].

As we have mentioned above, numerous authors study the effectiveness of XP practices. But most research in this line is based on empirical studies, which consume a lot of resources. There is little empirical evidence on a scientific basis to support the implementation of XP and little research captures the interlocking relationships among XP practices [7, 8]. At the same time, XP provides very little project management support that helps the development team to make decisions [9] and according to Melo et al., agile team management is the most influential factor in achieving agile team productivity [10].

In these changing environments, where XP is particularly suitable, techniques for modeling and simulation are especially useful; because, through them, it will be possible to provide the development team with a way of experiencing the different configurations and understanding the effects of different policies without assuming the risks and costs that a real experiment entails. At the same time, these tools are helpful for early estimations of the projects.

In this paper, we examine, through a literature review, the evidences of the application of modeling and simulation techniques to support the management in XP projects. From this review, we conclude that there is still work to be done in this area, and more specifically in the teamwork management. As a proof of concept, we present Sim-Xperience, a simulation model that, in contrast with the models we can find in the literature, has been implemented under the paradigm of agent-based simulation. It is intended to serve as an aid to decision making in the field of software project management under the XP methodology. Sim-Xperience pays special attention to the development team and their evolution during the project, covering one of the principles of the XP methodology: the focus on people.

The model presented in this paper simulates an iteration of the development process allowing us to configure a huge set of input parameters that allows us to adapt the model to the specific characteristics of the project that is being simulated. The results of the iteration can be used, if desired, to feed the input of the next iteration of the model and thus, simulate either a part or a whole project.

During the simulation, Sim-XPerience uses an animation to dynamically display the behavior of the team pairs as well as the evolution of the team's experience and the performance of the tasks over time. The simulation model provides information about the Key Performance Indicators (KPIs) of the project under different task allocation strategies.

In relation with this context, the present work considers the following general Research Questions:

RQ1. Is there evidence in the literature of the application of modeling and simulation techniques in the field of XP project management?

RQ2. Would it be possible to simulate the behavior of the development team and their evolution during in a XP project?

The structure of the paper is as follows: Sect. 2 discusses related work on the application of modeling and simulation techniques to the XP project management; Sect. 3 explains the methodology followed to develop Sim-XPerience; Sect. 4 details model building following the phases of the methodology explained in Sect. 3; the case of use chosen to display the results of the simulation model will be explained in the simulation phase. Finally, in Sect. 5 we present the conclusions and the future work to be developed in this area.

2 Related Work

Modeling and simulation techniques have been applied since the early 90 s to address different issues related to the software process, being considered valuable tools in the tasks of management and decision making [11].

The discipline of modelling the software process with System Dynamics started with Abdel-Hamid in 1984 [12]. Since then, several simulation models have been developed to respond to different questions related to the software development process. A comprehensive view can be obtained by combining two systematic literature reviews of software process simulation modeling, the initial stage [13] with the extended stage [14]. Also it is possible to find an important contribution to the system dynamic modeling body of knowledge in [15].

There are a variety of simulation approaches, but there are three major approaches used to build simulation models: System Dynamics (SD), Process-centric ("Discrete Event", DE) modeling, and Agent Based modeling (AB).

In this section we present the results of the literature review performed to find previous work in the application of simulation modeling in the field of software development with XP, paying special attention to the work related to aiding decision-making in managing XP software projects. To carry out the review we have searched in the following electronic databases: Scopus, IEEE Xplore, ACM Digital Library, SpringerLink and Web of Science. This search was conducted in April 2016, using the search string: "extreme programming" AND "Simulation". Below, we present the results obtained.

Kuppuswami [16] proposes a dynamic simulation model of the XP development process with the aim of determining the effects of XP on the cost of change curve. The paper describes the steps to build the curve using the model. In another work [17] the author develops a dynamic simulation model in order to analyze the effect of the XP practices on the development effort. The results showed that an increase in usage level of individual XP practices reduces development cost.

Vojislav [18] proposes the use of system dynamics to model, simulate and analyze the software development process utilizing the XP approach. The model focused specifically on the quality assurance aspects of this methodology. Specifically, the model simulates the effects of pair programming, refactoring, test-driven development, and small developmental iterations focused on a task or unit basis.

Marco Melis et al. [8] develop a model of hybrid simulation, discrete event and System Dynamics, focused on two XP practices: pair programming and tests-first programming. Using contrasted data from real projects, the model simulates the evolution of an XP software project changing the usage levels of the two practices and concludes on one hand that the simulation leads to deeper understanding of the XP process dynamics and on the other hand that the increase in the usage of such practices significantly diminishes product defectiveness. We found a complementary study performed by Turnu et al. [19] that study the effects of the adoption of XP practices on open source development using a simulation model. In particular, it evaluates the effects of TDD (Test Driven Development) comparing the results with and without using the TDD practice and concluding that the incorporation of agile practice yields better results in terms of code quality.

Navarro E.O. [20] uses the Simulation as an educational tool to teach about Software processes. Amongst her models she presents a specific tool to teach XP processes.

The work conducted by Yong and Zhou [7] describes a system dynamics model of the software development process with XP that can be used to quantitatively evaluate the software process and to study the effects of the XP practices through simulation. The model can be a useful tool for project managers to decide the adoption of part of the XP practices. Table 1 summarizes the results of the review.

Table 1. Summary of review results.

Reference	Approach	Issues addressed
[16]	SD	to find the effects of XP on cost of change curve
[17]	SD	to analyze the effect of the XP practices on the development effort
[8]	SD&DE	evolution of an XP software project changing the usage levels of two XP practices (Pair Programming and Test Driven Development)
[20]	Own	educational tool to teach about XP processes
[19]	SD	study the effects of the adoption of the Test Driven Development practice on open source development
[18]	SD	study the effects of pair programming, refactoring, test-driven development, and small development iterations focused on a task or unit basis
[7]	SD	to quantitatively evaluate the software process and to study the effects of the XP practices through simulation

From the review performed we can observe that there are evidences in the literature of the effectiveness of the modeling and simulation techniques in the scope of XP project management. Thus, our first RQ is answered, but we conclude that there is still work to be done in this area, especially as far as the teamwork management is concerned because very few works deal with teamwork issues. Sim-Xperience, aims to serve as a proof of concept in the application of modeling and simulation techniques to the XP teamwork.

All simulation models discussed above follow the simulation approach based mainly on systems dynamic in which the abstraction process is performed on the system as a whole, focusing on the causal relationships linking the observable variables. Sim-Xperience, however, has been made using agent-based paradigm focusing on the basic components of the system, their individual properties and the interaction between them; leaving the overall system performance to emerge as a result of the conditions imposed on their components. Since the main objective of Sim-Xperience is to observe the evolution of the team pairs, agent-based approach is particularly appropriate.

3 Development Methodology

There are different methodologies for the development of simulation models [21–24]. This work has used as the main reference the consolidated process SPSM (Software Process Simulation Modelling) [24].

The process used consists of the following steps:

1. Defining the problem
2. Choosing the simulation approach
3. Choosing the simulation tools and techniques
4. Designing the model
   (A) Defining the model scope
   (B) Specifying the purpose
   (C) Identifying the output variables
   (D) Identifying the input parameters
   (E) Model Conceptualization
5. Implementation of the model
6. Model verification, calibration and validation
7. Simulation and analysis of results
8. Documentation

4 Model Building

To explain the model building we follow the scheme proposed in Sect. 3.

To develop the simulation model we have chosen the agent-based paradigm because one of the main objectives is to support the development team management, having in mind factors of experience and motivation with the aim, among others, of observing the behavior and evolution of each pair of the team during the successive project iterations. The agent-based paradigm is fit for this purpose. The model has been implemented through the simulation tool AnylogicTM [25].

The scope of Sim-Xperience is a project iteration. The model focuses on the simulation of independent iterations of the project, assuming development teams from 2 to 10 members. However, the model allows us to store the output from an iteration in order to serve as input for the next and so, be able to simulate a whole project. To develop the model we assume that the different user stories of the iteration are broken down into smaller units, called ‘tasks’ [26].

Table 2. Model output variables.

Output variable	Description
Task Time	Time to complete each task (days)
Iteration Time	Time to complete all the tasks of the iteration
IDE	Increase of the Team member Experience in each development phase
ITE	Increase of the Team Experience
Cost	Cost of iteration (euros/month)

The simulation aims to show the evolution of the development team as well as some of the KPI of any software development project, such as cost and time. The Table 2 shows the main output variables.

To implement other XP practices, we have considered that team members do not work more than 40 h per week (8 h/working day). It has also been considered that the tasks belonging to a user story are divided into four main development phases: analysis, design, coding and testing.

The input parameters of the model can be classified into three categories: team parameters, iteration parameters and task parameters. Table 3 shows the main model input parameters.

4.1 Model Conceptualization and Implementation

To perform the model conceptualization, we have considered two agents: team member and task. Both of them will pass through different states reflected in Table 4. The model follows the team member's state diagram shown in Fig. 1. The simulation ends when all the tasks are completed. The team member state will pass from Available to Standby when a task is assigned to him. The team member state will change from Standby to Working when he has a pair assigned to the same task and will pass from Working to Available when the task is completed. The order of task performance is based on the previously associated priority, considering the possible dependence that may exist on other tasks

The task will pass from *No Allocated* to *Allocated* when two team members are assigned to it. The task will pass from *Allocated* to *Completed* when its time ends.

When a task is *Allocated* and is being performed, the evolution of the experience factor takes place. The experience of a team member can be improved in two aspects: on the one hand, due to the fact of working in pairs (synergy), and on the other hand due to the time spent in the phase to which the task belongs.

The time necessary to complete a task is affected by two main reasons: the experience of the pair of team members and the task difficulty, but it could be affected also by the motivation factor and other external delays that might happen.

Once all the tasks have been completed the model will show the results regarding time, cost and experience improvement. It would be possible to see the improvement of each team member experience in each development phase and the evolution of the average experience of the team. Also it is possible to compare the initial values for the estimation of time and cost with the final values after simulation.

Table 3. Model input parameters.

Category	Input parameter	Description
Team parameters	Team member name	
	Number of team members	
	Levels of expertise	The choice of the number of levels set the range of expertise, it is possible to choose between 1 and 5 (5 levels).
	Team member expertise	Initial level of expertise of each team member, in each development phase
	Team member salary	Junior: team members with less than two years of experience. Semi-Senior: team members with less than four years of experience. Senior: team members with more than four years of experience.
	Team input file	Possibility to choose saving the team members' input file after the simulation in order to use these results as an input for the next iteration
	Team experience	Possibility to decide whether the team members experience will or will not influence time of task completion
	Team motivation	Possibility to decide whether the team motivation will or will not influence time of task completion
Iteration parameters	Number of tasks	
	Estimated duration	The iteration will have a set delivery date, which will have a total duration estimated in accordance with the recommendations of XP
	External delays probability	It indicates the estimated probability for external delays Between 0 and 1
	External delays frequency	It indicates the frequency estimated for external delays. It can be every day, every week, every 3 days etc.
Task parameters	Task ID	It is an integer that identifies the task
	User story	User story to which the task belongs
	Development phase	Analysis, design, development or testing
	Estimated time	Estimated time to perform the task
	Task priority	Indicates the priority of a user story, (all the tasks belonging to the same user story have the same priority)
	Difficulty	Indicates the task difficulty (between 0 and 1)
	Dependence	Task that must be completed before
	Description	Task description

Table 4. Agents' states.

Agent	State	Description
Team member	Available	The Team member is available. Waiting for a task to be assigned
	Standby	The Team member is waiting for a partner to share the task performance
	Working	Working in the phase where the task belongs
Task	No allocated	Initial state for all the tasks
	Allocated	The task has a pair of team members assigned and working.
	Completed	The task is finished.

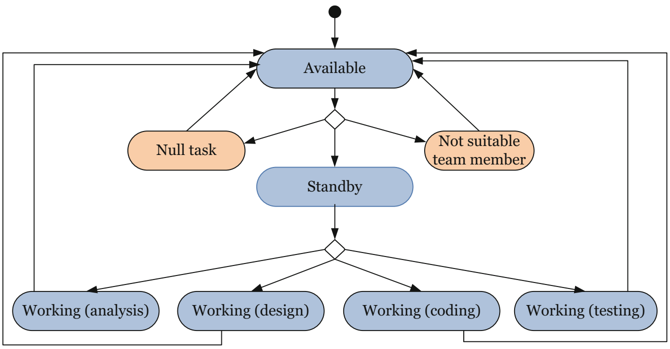

Fig. 1. Team member state diagram.

4.2 Model Validation

Validation of simulation models is an issue that de-serves attention. Validation of the model structure and simulation results is to ensure that the model adequately represents the system under study and its expected behavior [27, 28].

To perform the validation of Sim-Xperience we have used the following validation techniques [29]:

– Animation: the model dynamically and graphically shows the behavior of the process as the simulation progresses. This enables the team to observe the progress of the process and to stop to modify the parameters if it were necessary. It is possible to observe the distribution and behavior of pairs as well as the evolution of KPIs through some charts. Thus, we can see the time and cost deviations with regard to the estimations, as well as the evolution of the learning curve of the team, the task allocation between the pairs and the evolution of the task performed by each pair of the team in each phase.
– Face Validity: The model has been tested by users, from both academic and industrial scopes, familiar with the agile software development and XP practices and they agree that the model behavior is reasonable comparing with the real-world system.

– Extreme Condition Tests: The model structure and outputs are plausible for any extreme and unlikely combination of the model input parameters.
– Internal Validity: Multiple simulations have been performed to validate the consistency of the model.
– Parameter Variability - Sensitivity Analysis: these tests consist of determining those parameters to which the model is highly sensitive, and asking if the real system would exhibit similar behavior to the corresponding parameters. This technique has been used in combination with Face validity technique to validate the behavior from a qualitative point of view.
– Traces: The model has been traced to determine that the logic is correct.

It is difficult to fully validate a simulation model due to the lack of historical data of projects and the diversity of organizations and projects [30]. The values for the Sim-Xperience input parameters used in the simulation experiment were extracted from the limited project data provided from the literature.

To use Sim-Xperience as a predictive tool in an industrial context, it is necessary to calibrate it to the specific features of the organization. It is important to remember that the model is a customizable tool, providing a wide set of configuration parameters that, once calibrated, describes the project or organization in particular. This is a common feature in most of the simulations models found in the literature [30–32]. However, we are working to extend the model validation using data from real projects.

4.3 Simulation and Analysis of Results

To illustrate the results we propose the following case study: we will study the effect of different task allocation strategies in the project behavior. We will compare the evolution of the output variables between two different strategies. Thus, each strategy will be simulated for the development of ten similar projects in order to show a long term evolution. We assume that when a project ends, the same team joins the next project, having their experience increased.

Even though the XP teams are self-regulated, some kind of strategy is used to allocate the work [28]. In the work allocation process two main factors are considered: the number of tasks performed by the team member and the experience in the task phase. Depending on the strategy chosen by the user before starting the simulation, the model will give a higher priority to one of these factors.
We have considered two main strategies:

- Strategy A: the priority criterion is team member experience with a focus on workload allocation for team productivity. With the aim of improving the team productivity the task will be assigned to the pair of team members available who have more experience in the phase to which the task belongs.
- Strategy B: the priority criterion is task scheduling with a focus on workload allocation for uniformity. All workers must have a similar workload. Then, a task will be assigned to the available team member with fewer tasks performed in the phase. This strategy is more likely to mix team members with different levels of experience in a pair.

Table 5 shows the values for the Team input parameters for each of the ten projects.

Table 6 shows the values for the Iterations input parameters. We assume 5 iterations in each project. All the projects will have the same parameters in both strategies.

Table 5. Values for the team input parameters.

Team Parameter	Value
Levels of expertise	5
Number of team members	8
Team member expertise	In the range of [1–5]
Team member salary	Junior: 900 eur/month Semi-Junior: 1500 eur/month Senior: 2300 eur/month
Team input file	Yes
Team experience	Yes
Team motivation	Yes

Table 6. Values for the iteration input parameters.

Iteration Parameter	It1	It2	It3	It4	It5
Number of tasks	52	28	24	12	23
Estimated duration	70	43	37	48	50
External delays probability	0.3	0.4	0.5	0.4	0.3
External Delays Frequency	1	2	3	5	7

To give the values to the tasks input parameters, we have randomly selected the values of each parameter within the ranges previously established.

Figure 2 shows the final results of the simulation for the proposed experiment for the main outcome variables: time (left) and cost (right). The red bars show the results following the strategy A and the blue bars correspond to strategy B.

Strategy A provides significantly better results for indicators of time and cost during the first four projects than Strategy B. However, from the fifth project, the results begin to be more balanced in both strategies. On the other hand Fig. 3 represents

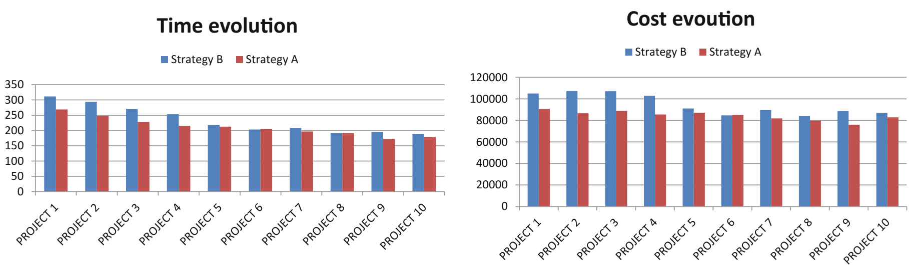

Fig. 2. Results for time (left) and cost (right) output variables.

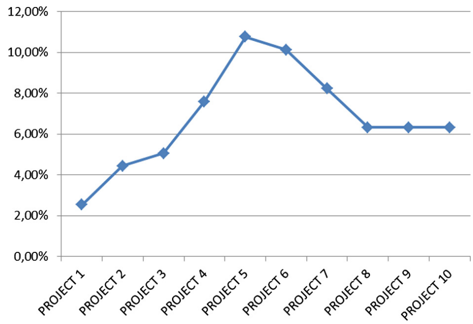

Fig. 3. Rate of improvement in the average experience of the team. Strategy B vs. Strategy A.

the rate of improvement in the average experience of the team that strategy B presents vs. Strategy A. We can see that Strategy B achieved a greater increase in experience. For this indicator Strategy B offers Project 5 an increase of 11% over Strategy A.

In conclusion we can say that, in the short term, Strategy A achieves better results for the main indicators of the project. But, after a certain time, for similar results in time and cost, we get more trained teams using the Strategy B. In addition, Strategy B is more in line with the spirit of the XP practices.

It is important to note that the model dynamically shows the behavior of the process as the simulation progresses through different graphics. Figure 4 shows, as an example, a screen shot of one of the screens that the model shows during simulation.

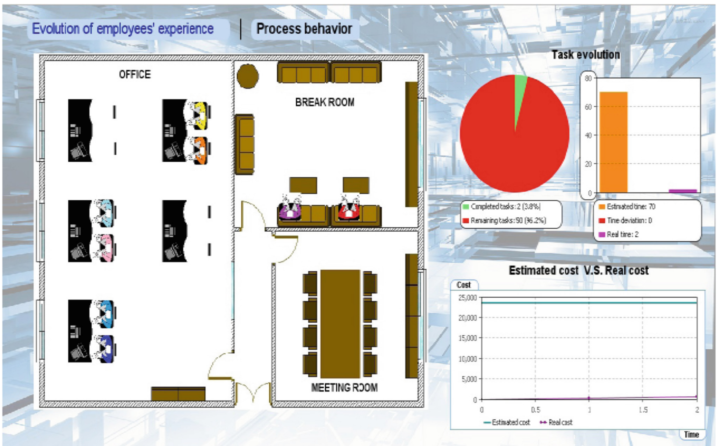

Fig. 4. Screen shot during the simulation of Sim-Xperience.

5 Conclusions and Future Work

In this paper we have firstly, explored the evidences of previous work in the application of modeling and simulation techniques in the field of XP project management. Using the results of the literature review carried out, we have concluded that although there are evidences of the effectiveness of these techniques in the field of XP project management, there is still work to do, particularly about the teamwork management issues. With the second RQ set at the beginning of the paper we wondered if it would be possible to simulate the behavior of the development team and their evolution during in a XP project. To answer the second RQ we have developed, as a proof of concept, an agent-based simulation model called Sim-Xperience. We have presented the model building and its simulation. The model allows for testing of different decisions in managing the team in software development projects following the principles of XP methodology.

We have shown that Sim-Xperience could help decision-making in teamwork management in XP projects. The model can dynamically show the evolution of an XP project and of its development team. It allows us to vary a large number of input parameters and thus adapt it to the specific needs of the development team and project. To illustrate the model simulation we have conducted a case study, where we have seen the results of the simulation model under two different allocation tasks strategies, concluding that using a strategy where the team member experience is not the priority criterion is better for the increase of team experience in the long term.

As future work to develop, we consider extending the model with other parameters and KPIs as well as completing the model validation using data from real projects. Also, we intend to extend the model with other parameters relevant to process management.

Acknowledgements. This work has been funded by the Spanish National Research Agency (AEI) with ERDF funds under grants TIN2013-46928-C3-2-R and TIN2016-76956-C3-3-R, and the Andalusian Plan for Research, Development and Innovation (grant TIC-195).

References

1. Dyba, T., Dingsøyr, T.: Empirical studies of agile software development: a systematic review. Inf. Softw. Technol. **50**, 833–859 (2008)
2. Dyba, T., Dingsøyr, T.: What Do We Know about Agile S. D.? Soft. IEEE **26**, 6–9 (2009)
3. Serrador, P., Pinto, J.K.: Does Agile work? — A quantitative analysis of agile project success. Int. J. Project Manage. **33**, 1040–1051 (2015)
4. Awad, M.A.: A comparison between agile and traditional software development methodologies, M.S. Thesis, University of Western Australia (2005)
5. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston (2000)
6. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley, Boston (2004)

7. Yong, Y., Zhou, B.: Evaluating extreme programming effect through system dynamics modeling. In: International Conference on Computational Intelligence and Software Engineering, CiSE (2009)
8. Melis, M., Turnu, I., Cau, A., Concas, G.: Evaluating the impact of test-first programming and pair programming through software process simulation. Softw. Process Improv. Pract. **11**, 345–360 (2006)
9. Valkenhoef, G.V., Tervonen, T., Brock, B., Postmus, D.: Quantitative release planning in extreme programming. Inf. Softw. Technol. **53**, 1227–1235 (2011)
10. Melo, C.O., Cruzes, D.S., Kon, F., Conradi, R.: Interpretative case studies on agile team productivity and management. Inf. Softw. Technol. **55**, 412–427 (2013)
11. Kellner, M.I., Madachy, R.J., Raffo, D.M.: Software Process Simulation Modeling: Why? What? How? J. Syst. Softw. **46**(2), 91–105 (1999)
12. Abdel-Hamid, T.: The dynamics of software project staffing: An Integrative System Dynamics Perspective, Ph.D. dissertation, Massachusetts Institute of Technology, (1984)
13. Zhang, H., Kitchenham, B., Pfahl, D.: Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review. In: Wang, Q., Pfahl, D., Raffo, David M. (eds.) ICSP 2008. LNCS, vol. 5007, pp. 345–356. Springer, Heidelberg (2008). doi:10.1007/978-3-540-79588-9_30
14. Zhang, H., Kitchenham, B., Pfahl, D.: Software Process Simulation Modeling: An Extended Systematic Review. In: Münch, J., Yang, Y., Schäfer, W. (eds.) ICSP 2010. LNCS, vol. 6195, pp. 309–320. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14347-2_27
15. Madachy, R.J.: Software Process Dynamics. Wiley-IEEE Press, Chichester (2008)
16. Kuppuswami, S., Vivekanandan, K., Rodrigues, P.: A system dynamics simulation model to find the effects of xp on cost of change curve. In: XP2003 Conference Proceedings, pp. 54–62 (2003)
17. Kuppuswami, S., Vivekanandan, K., Ramaswamy, P., Rodrigues, P.: The effects of individual XP practices on software development effort. SIGSOFT Softw. Eng. Notes **28**(6), 6–7 (2003)
18. Misic, V.B., Gevaert, H., Rennie, M.: Extreme dynamics: Modeling the extreme programming software development process. In: Proceedings of ProSim04 workshop on Software Process Simulation and Modeling, pp. 237–242 (2004)
19. Turnu, I., Melis, M., Cau, A., Setzu, A., Concas, G., Mannaro, K.: Modeling and simulation of open source development using an agile practice. J. Syst. Archit. **52**(11), 610–618 (2006)
20. Navarro, E.O.: SimSE: A Software Engineering Simulation Environment for Software Process Education. University of California, Irvine (2006)
21. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill, Boston (2000)
22. Martinez, I.J., Richardson, G.P.: Best practices in system dynamics modeling. In: Proceedings of the 29th International Conference of the System Dynamics Society, Hines, .H., Diker, V.G. (Eds.), Atlanta, GA, USA, pp. 1–22, plenary paper (2001)
23. Law, A.M.: How to build valid and credible simulation models. In: Proceedings of the 2009 Winter Simulation Conference, pp. 24–33 (2009)
24. Ali, N.B., Petersen, K.A.: A consolidated process for software process simulation: state of the art and industry experience. In: Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2012, pp. 327–336, art. no. 6328171 (2012)
25. Anylogic (AnylogicTM) http://www.anylogic.com. Accessed: June 2017]
26. Sharp, H., Robinson, H.: Collaboration and coordination in mature eXtreme programming teams. Int. J. Hum Comput Stud. **66**, 505–518 (2008)

27. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2011 Winter Simulation Conference, pp. 183–198 (2011)
28. Sokolowski, J.A., Banks, C.M.: Principles of Modeling and Simulation: A Multidisciplinary Approach. Wiley, Hoboken (2009)
29. Sargent, R.G.: Verification and validation of simulation models. In: Proceedings of the 2011Winter Simulation Conference 2011, pp. 183–198 (2011)
30. Choi, K., Bae, D.: Dynamic project performance estimation by combining static estimation models with system dynamics. Inf. Softw. Technol. **51**, 162–172 (2009)
31. Garousi, V., Khosrovian, K., Pfahl, D.: A customizable pattern-based software process simulation model: design, calibration and application. Softw. Process Improv. Pract. **14**(3), 165–180 (2009)
32. Kouskouras, K.G., Georgiou, A.C.: A discrete event simulation model in the case of managing a software project. Eur. J. Oper. Res. **181**(1), 374–389 (2007)

An Exploratory Study on Usage of Process Mining in Agile Software Development

Sezen Erdem[1(✉)] and Onur Demirörs[2,3]

[1] ASELSAN INC., Ankara, Turkey
erdem@aselsan.com.tr
[2] Computer Engineering Department,
Izmir Institute of Technology, Izmir, Turkey
demirors@metu.edu.tr
[3] School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia

Abstract. Agile software development methods have become popular in the software development field during the last decade. Majority of software organizations develop or claim to develop software based on agile methods. Process mining is a process management technique that allows for the analysis of business processes based on the event logs. The aim of process mining is to discover, monitor and improve real processes, but not assumed processes, by extracting knowledge from event logs readily available in information systems. Process mining can be used to discover agile processes followed in organizations/projects to determine the actual processes followed. Process mining can also establish the necessary evidences for assessing or measuring the agility of organizations. This study explores the usability of process mining methods in agile software development context. The results of an exploratory case study on using process mining techniques in a software project managed by Scrum are depicted. We also discuss the benefits of the process mining techniques used and compare different tools utilized.

Keywords: Process mining · Agile software development · Process discovery · Process conformance checking

1 Introduction

Process is defined as sequence of interdependent and linked activities to convert inputs into outputs. Manufacturing industries have focused on improving processes as it enables improvement of cost, cycle time and reliability at the same time. Software development society has also been working on developing methodologies to improve processes. Although the frameworks such as SPICE and CMM have significant success they also pose significant challenges [1–4]. Over the last two decades, agile software development methods have become very popular in software development area. Agile methods which are based on iterative development foundations brings more light weight and people centric view point when compared with the traditional approaches such as waterfall. Manifesto for Agile Software Development advices the agile teams to

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 187–196, 2017.
DOI: 10.1007/978-3-319-67383-7_14

value individuals and interactions over processes and tools [5]. These principles frequently lead to development processes which are not formalized. Some agile methods as Scrum and XP prescribe a set of practices. Frequently these practices are not applied as they should or interpreted by agile teams in such a way that the outcomes are unpredictable [6, 7]. Determination of the sequence of events and the techniques used by each personnel independently may result in inconsistency, instability, and unpredictability. Moreover, different development teams in an organization may interpret the agile method rules differently which leads to interoperability problems between the projects of an organization. So it is vital to draw some borders. However having defined processes also does not guarantee that the personnel follows the processes with high fidelity. A method to extract the actual processes followed by agile teams can be valuable and will help to visualize consistency, stability, interoperability and repeatability problems. Extracting actual process might also help organizations to assess their agility.

Process mining is a process management technique that allows for the analysis of business processes based on the event logs. The aim of process mining is to discover, monitor and improve enacting processes, by extracting knowledge from event logs readily available in today's information systems [8]. Process mining can provide the right tools to discover agile processes followed by agile teams to understand the reality of the organization. Process mining can also be a base for constructing an assessment framework to measure the agility of the organizations.

In this study, we performed an exploratory case study to evaluate the usage of process mining in agile software development context. A scrum project in a defense industry company is selected as target and traces of product backlog items and bug records are analyzed to mine their actual state flows. Results are compared with the predefined flows. This study is important to show the applicability of process mining techniques in agile context. Agile processes leave less traces when compared with the traditional software development approaches like waterfall. We have observed that, observing the process fidelity is difficult in agile contexts and working on process mining in agile context also requires a data collection methodology. However, we have also observed that process mining can be used to extract the actual processes, and organizations can be aware of their agile maturity using the results.

The remainder of this paper is organized as follows. In Sect. 2, a brief information about process mining is given and its usage areas in agile software development is discussed. In Sects. 3 and 4, an exploratory case study as an introduction to mining the actual process in a scrum project and its results are shared with the readers. And in Sect. 5, concluding remarks and future works are discussed.

2 Process Mining

Process mining is a relatively young research discipline. The techniques related with process discovery has its roots in various disciplines such as data mining, computational intelligence and machine learning. The earliest study on process discovery is attributed to Cook et al. [9–11]. Agrawal et al. are also early pioneers of process mining [12]. A number of research studies on process mining have been conducted during the last decade and the trend is spectacular. Aalst provides a comprehensive overview of the state-of-the-art in process mining in his books [13, 14]. There are various methods and algorithms proposed for different purposes. Cook and Wolf examined the use of statistical analysis methods (Rnet, Ktail and Markov) for use in mining tasks [15]. Also heuristic approaches [16], genetic algorithms [17], fuzzy mining techniques [18] and cluster analysis [19] are used in process mining. Akman and Demirors [20] studied on applicability of the process discovery algorithms for software organizations and their findings provide insight on how process discovery and mining algorithms could be effectively used.

As the capabilities of information systems and features of CASE tools are improved, it become possible to record and analyze flow of software development process. Each action generates some event log data, each shareholder in development process leave some footprints that can be traced to extract information. However, the challenge is to exploit data in a meaningful way. As a research discipline which sits between computational intelligence and data mining on the one hand, and process modeling and analysis on the other hand, the aim of process mining is to extract such meaningful information. Starting point for process mining is an event log. All process mining techniques assume that it is possible to sequentially record events such that each event refers to an activity and is related to a particular case [8]. Process mining studies can be categorized with respect to their purpose: discovery, conformance checking and enhancement. Process mining also covers different perspectives, control-flow, organizational, case and time perspectives.

Agile software projects are generally developed by small teams and in short iterations. Agile methods are more light weight, more people centric and leave less traces when compared with the traditional approaches such as waterfall. Process mining techniques can be used to analyze application of agile methods. Scrum is the most popular agile methodology around the world and prescribe a set of practices for the teams as team formation, roles, meeting schedules. Application life cycle management tools have built-in agile templates to help teams to follow their jobs. Activities of agile teams can be mined to discover what is going on and how is going on. Process discovery will be beneficial to extract the steps followed by agile teams, required inputs to progress, intermediate outputs generated inside the iterations and roles that has taken places through development. It also becomes possible to compare application of agile methods by different agile teams with process discovery. Caldeira [21] made a research to increase the awareness of software developers about their development process and reveal improvement opportunities by mining event logs of development environments. In this study, they plan to discover the process of an agile team by mining the data generated by their development tools. Rubin et al. [22] also has a work related with process mining and agile development. The work describes a bottom-up approach, which takes event logs (e.g., trace data) of a software system for the analysis of the user and system runtime behavior and for improving the software. It does not deal with the development processes but the improvement of software functionality by using process mining techniques in an agile manner.

3 Case Study

3.1 Case Study Design

In order to test the applicability of process mining in agile software development, we have planned an exploratory case study in an organization who utilizes an agile methodology.

The research questions are determined to be:

(1) Is it possible to trace event logs generated through a project developed using agile methodologies?
(2) Is it possible to extract the flow of actions in terms of agile events?
(3) Is it possible to provide useful feedback to the software development team about their processes?

The case selection criteria determined to be:

- Software development project managed by Scrum. Scrum is the most popular agile methodology and it has predefined rules that can be used for conformance checking. Also sprints generate time periods to analyze and improve the process.
- At least 5 employees will be in Scrum Team and at least 6 sprints should happen. Otherwise meaningful data for analysis cannot be obtained.

We have planned the following activities:

- Collect data from the tools used by the organization.
- Analyze data and extract state flow information for Product Backlog Items and Bugs
- Transform state flow information into a format readable by process mining tools
- Run process mining analysis with Disco and ProM tools (as they are the most frequently used tools)
- Evaluate the results

3.2 Application of the Plan

Case study is performed in an organization which is a leader system developer in the defense industry sector and have been performing development projects for over 40 years. Case study is conducted on a software development division of the company which develops C4I software. A software project which is developed by following SCRUM methodology is selected to conduct case study. The details of the project is given in the following table (Table 1).

In the case study, the data from Microsoft TFS is analyzed to discover flow of Product Backlog Items (PBI) and Bugs from creation to closing. An application is developed in C# with MS Visual Studio to analyze and transform info into XES. Following steps are executed:

- Microsoft TFS has a development API and supports developing application with C#. Developers can create queries to extract data from TFS database. We developed an application to access the Scrum project data. We queried history of product

Table 1. Project information

Agile method	SCRUM
Scrum development team size	6 person
Number of sprints	9
Sprint length	4 weeks
Agile management tool	Microsoft TFS

backlog items and bugs. History data contains all of the changes made on PBIs and bugs. History data is processed to extract state transitions and data is filtered and only parts of the data required for process mining analysis are extracted.

- Processed data is not in the format that process mining tools can understand. Our application also transforms the data into XES format to feed the process mining tools.
- Disco and ProM tools are used to analyze the data. Input is feed as XES files.
- Results are evaluated by discussion.

4 Case Study Results

In the Sprint 1 planning meeting, the team defined a flow to execute PBIs (Fig. 1) and Bugs (Fig. 3). The results of process mining analysis based on the data collected through 9 sprints are compared with the defined flows (Fig. 2).

When the actual flow and defined flow for PBI and Bug states are compared, the patterns are pretty much similar to each other. However there are some patterns which are not expected to occur in normal execution (Fig. 4):

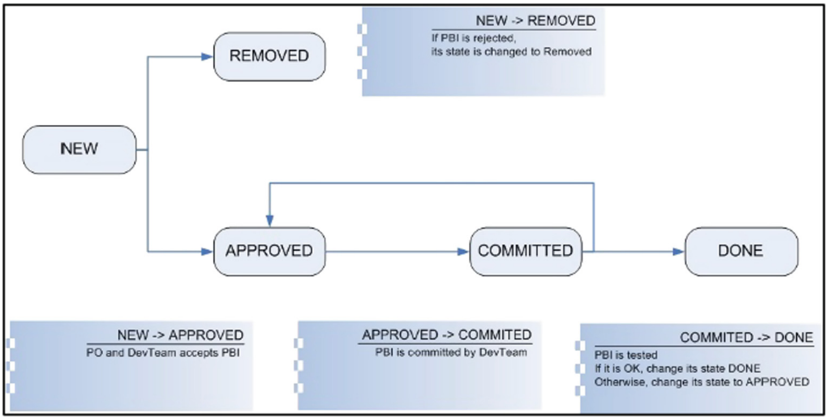

Fig. 1. Defined flow for PBI execution

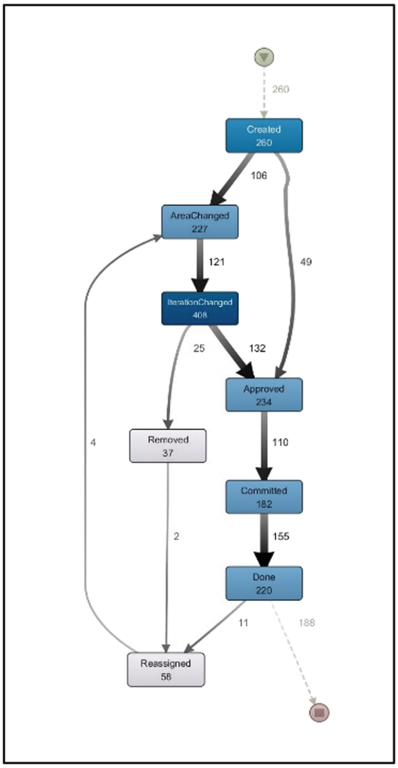

Fig. 2. Actual flow for PBI execution (Disco)

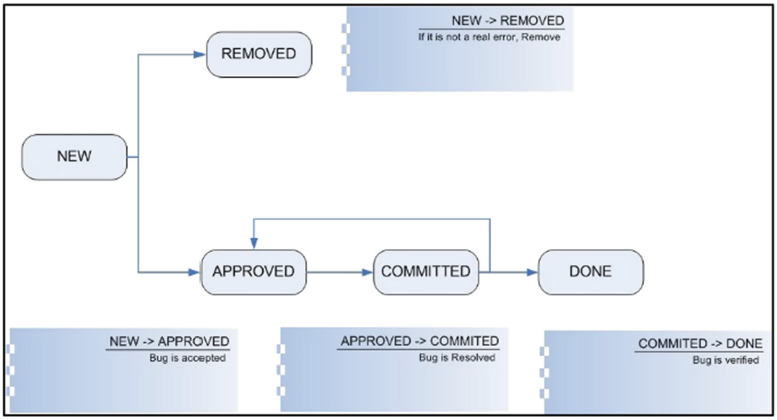

Fig. 3. Defined flow for Bug execution

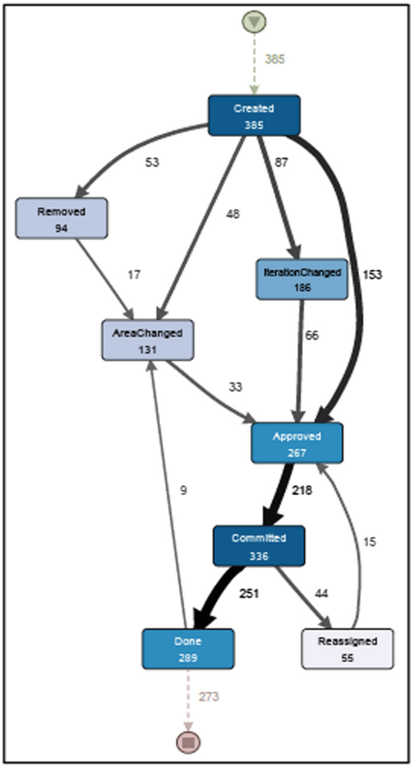

Fig. 4. Actual flow for Bug execution (Disco)

– PBI state change from Done to Reassigned
– PBI state change from Removed to Reassigned
– PBI area change at Done state
– Bug area change at Done state

These patterns are analyzed by the team and some are signed as noise (made by mistake). There are also some patterns which are real execution models. These patterns signals that the team made some mistakes in the execution. The team changed the state of some PBI to Done. But at later iterations, it is realized that some of these PBIs are not really done. They have misinterpreted the Scrum method. All these misuse or misinterpretation patterns are opportunities for the team to improve their process and also opportunity to detect the point where they leave Scrum rules.

During the case study, we faced with some difficulties especially in extracting data and transforming it into a format that can be processed by process mining tools. Also in analysis phase, there are many algorithms that can be run on data. The decision of which algorithm best fits with the data to generate successful results is a non-trivial issue.

There exist many tools with process mining capability as ProM, Disco (Fluxicon) ARIS Process Performance Manager (Software AG), Comprehend (Open Connect), Discovery Analyst (StereoLOGIC), Flow (Fourspark), Futura Reflect (Futura Process Intelligence), Interstage Automated Process Discovery (Fujitsu), OKT Process Mining Suite (Exeura), Process Discovery Focus (Iontas/Verint), ProcessAnalyzer (QPR), Rbminer/Dbminer (UPC) and Reflect one (Pallas Athena). Among several available tools, ProM and Disco are selected for the use in analysis since they are popular and also available for academic evaluation purpose. Both ProM and Disco have powerful analysis capabilities. Disco provides a simpler user interface and a filtering mechanism which has clear representation compared to ProM. ProM has many plug-ins and serves a high number of alternative to run analysis which makes the tool very powerful. But this can also make the tool complex. Since our work is not comprehensive, we cannot make a detailed comparison between the tools. The result generated by both tools are nearly the same. But as our work progress through analysis of agile processes, we will experience detailed features of the tools and have chance to compare them.

5 Conclusion and Future Works

Process mining traces the footprints and outputs generated through the application of development process to extract knowledge about the processes. Agile methods have become a de facto standard in software development over the last decade. However, due to the developer centric nature of methods, process fidelity issues may not be given required importance which might lead to critical problems. Process mining can be used as a beneficial tool for analyzing the application of agile methods in organizations. Case study mentioned in this paper has shown that agile methods leaves many evidences to be tracked to extract knowledge about real process. By analyzing the data, beneficial results to make agile team aware of their process can be obtained and also opportunities to diagnose the failing parts of the applied process.

However, reliability of the data is the most important criteria in the correctness of the analysis. Data collection is in the hearth of the process mining. The quality of process mining results heavily depend on the input. Therefore, event logs should be treated as first-class citizens in the information systems supporting the processes to be analyzed. In order to benefit from process mining, organizations should aim at event logs at the highest possible quality level. As a future work we plan to develop a data collection model for process mining in agile methods (especially Scrum) to make efficient process mining analysis.

Although generally used for discovery, process mining is not limited with discovery. Discovery is just one of the three basic form of process mining. Conformance checking and enhancement can be used to extract knowledge to discover failure points in the real processes and generates improvement opportunities. Conformance checking can also be used for measuring the agility rates of organizations. As a future work, with the completion data collection model for process mining in agile methods, an assessment framework for measuring the agility of organizations based on process mining can be constructed.

This is an exploratory work, our future plans are constructing a framework for collecting data in agile software context for process mining and generating conformance checking methodology for assessing organizations/projects agility. Although this study is a preliminary work on using process mining in agile software development, our observations provided us motivation to continue the research to generate a data collection model for process mining in agile methods and a mining based assessment model to measure agility.

References

1. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI for Development: Guidelines for Process Integration and Product Improvement. Pearson Education, Upper Saddle River (2011)
2. Emam, K.E., Melo, W., Drouin, J.N.: SPICE: the theory and practice of software process improvement and capability determination. IEEE Computer Society Press, Los Alamitos (1997)
3. Uskarcı, A., Demirörs, O.: Do staged maturity models result in organization-wide continuous process improvement? insight from employees. Comput. Stand. Interfaces **52**, 25–40 (2017)
4. Tarhan, A., Demirors, O.: Apply quantitative management now. IEEE Softw. **29**(3), 77–85 (2012)
5. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Kern, J.: Manifesto for agile software development (2001)
6. Top, Ö.Ö., Demirörs, O.: Assessing software agility: an exploratory case study. In: Mitasiunas, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2014. CCIS, vol. 477, pp. 202–213. Springer, Cham (2014). doi:10.1007/978-3-319-13036-1_18
7. Ozcan-Top, O., Demirörs, O.: A reference model for software agility assessment: AgilityMod. In: Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2015. CCIS, vol. 526, pp. 145–158. Springer, Cham (2015). doi:10.1007/978-3-319-19860-6_12
8. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28108-2_19
9. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM TOSEM **7**(3), 215–249 (1998)
10. Cook, J.E., Wolf, A.L.: Automating process discovery through event-data analysis. In: Proceedings of 17th International Conference on Software Engineering, pp. 73–82 (1995)
11. Cook, J.E.: Process discovery and validation through event-data analysis, Ph.D. thesis, University of Colorado, Boulder, Department of Computer Science (1996)
12. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: Schek, H.-J., Alonso, G., Saltor, F., Ramos, I. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 467–483. Springer, Heidelberg (1998). doi:10.1007/BFb0101003
13. Aalst, W.M.: Process Mining: Data Science in Action. Springer, Cham (2016)
14. Aalst, W.M.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, New York (2011)
15. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Trans. Softw. Eng. Methodol. (TOSEM) **7**(3), 215–249 (1998)
16. Weijters, A.J., Van der Aalst, W.M.: Rediscovering workflow models from event-based data using little thumb. Integr. Comput.-Aided Eng. **10**(2), 151–162 (2003)

17. de Medeiros, A.K., Weijters, A.J., van der Aalst, W.M.: Genetic process mining: an experimental evaluation. Data Min. Knowl. Disc. **14**(2), 245–304 (2007)
18. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). doi:10.1007/978-3-540-75183-0_24
19. Schimm, G.: Mining exact models of concurrent workflows. Comput. Ind. **53**(3), 265–281 (2004)
20. Akman, B., Demirörs, O.: Applicability of process discovery algorithms for software organizations. In: 35th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2009, pp. 195–202. IEEE (2009)
21. Caldeira, J., e Abreu, F.B.: Software development process mining: discovery, conformance checking and enhancement. In: 2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC), pp. 254–259. IEEE (2016)
22. Rubin, V., Lomazova I., Aalst, W.M.: Agile Development with software process mining. In: Proceedings of the 2014 International Conference on Software and System Process, pp. 70–74. ACM (2014)

A Formalization of the ISO/IEC 15504: Enabling Automatic Inference of Capability Levels

Diogo Proença[1,2(✉)] and José Borbinha[1,2]

[1] INESC-ID, Lisbon, Portugal
{diogo.proenca,jlb}@tecnico.ulisboa.pt
[2] Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

Abstract. This paper presents a formalization that captures definitions of a number of concepts of ISO/IEC 15504 and relations among the concepts. The formalization is expressed in a formal language, OWL. The two main objectives for this formalization was to be consistent with the ISO/IEC 15504-5 process assessment model and to be effective, i.e., to allow for an automatic determination of a process capability level based upon data about the process attributes and ratings. The formalization is presented in a number of levels, from more general concepts to more specific. To assess the validity of the formalization, a number of test cases for the scenario of automatic determination of the capability levels were developed. A set of OWL reasoners were then used to derive the capability levels for the acquisition process group. While the test results were all positive, the real value of this formalization comes from the fact that it faithfully captured the main aspects of ISO/IEC 15504, a well established and accepted model for the assessment of processes capability levels, and that an inference engine was able to support the assessment of processes capability levels of an organization.

Keywords: ISO/IEC 15504 · Process capability · Process assessment · OWL

1 Introduction

The ISO/IEC 15504 was born as the "Software Process Improvement and Capability Evaluation" (SPICE) [1] in 1993. The acronym is still used today by the user groups of the standard and in the title of the annual conference. ISO/IEC 15504 is a reference for maturity models which consists of capability levels which in turn consist of the process attributes and further consist of practices, work products and resources. It helps assessors to give an overall determination of organizations' capabilities for delivering products, be it systems or services.

In order to assert a specific capability level, an organization must follow an assessment process which needs to show that the organization satisfies various requirements specified in ISO/IEC 15504-5 process assessment model. This model is rather complex since it includes many concepts that are interrelated in quite complicated ways. In order to assess the capability level of a certain process, an organization may use competent assessors who are familiar with the ISO/IEC 15504-5 process

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 197–210, 2017.
DOI: 10.1007/978-3-319-67383-7_15

assessment model; they can verify through indicators what process outcomes are in place, rate process attributes and determine the achieved capability level. Additionally, they must also collect all the "objective evidence used to support the assessors' judgement in rating process attributes" [2].

Checking all the objective evidence is a very tedious process. This issue is complicated even more when ISO/IEC 15504 evolves and some elements in the model change. For instance, a new type of practice is accepted by the industry or a new outcome is identified as necessary to satisfy a specific process. And one more issue is that people are prone to errors. In other words, the tedious process of verifying the objective evidence, rating the process attributes and determine the capability levels in an organization may be unintentionally erroneous. A computer based support tool would be able to alleviate some of the problems mentioned above. A computer tool would not be prone to errors. It could be faster. It would be cheaper to use a tool than people to check the evidence. Moreover, if designed properly, any modification in the ISO/IEC 15504 process assessment model (such as the new ISO 3300xx family of standards) can be relatively easily implemented, and a new version of the tool can be made available to the users in a relatively short time. The above discussion provides a motivation for the work presented in this paper. In order to address the issues mentioned above, a computer-interpretable version of the ISO/IEC 15504 model would have to be developed. By "computer-interpretable", we mean a version of the model that could be used by a computer (an inference engine) to actually infer whether a given organization has achieved a specific capability level, or infer the highest level that it can be classified at. For this task, the computer would have to be provided with appropriately structured input about the processes in the given organization. In order to facilitate such an inference task, a representation of the ISO15404-5 process assessment model would need to have computer executable semantics. Towards these goals, in this paper we provide a representation of the ISO15404-5 process assessment model in the Web Ontology Language (OWL) [4], a language with formal, computer-executable semantics. This means that a generic OWL inference engine can correctly derive the capability level of an organization's processes, provided the engine is supplied with the data about the organization as described in this paper. A computer-interpretable version of the ISO/IEC 15504 model could also play the role of an enabler of the interoperability among process management systems. When two process management systems share a formalization of the same model, they could exchange information about particular processes that are in use. The important aspect of this scenario is that the two systems would be able to "understand" the meaning of the exchanged information, in the sense that they would be able to (automatically) draw conclusions of the implications of a specific process. This paper is organized as follows. The next section gives a brief description of the background behind this work. In Sect. 3 we discuss the main usage scenario that we considered as a potential application of our ISO/IEC 15504 model formalization, i.e., the automatic inference of the capability levels. Section 4 provides a description of the ISO/IEC 15504 model formalization. The formalization is introduced in levels, in three levels, from the most general classes and properties to the lower-level subclasses. Section 5 presents a description of the approach to the validation of the formalization. In particular, it describes the test data and the tools used for inferring capability levels. And finally, Sect. 6 presents our conclusions and suggestions for future research.

2 Background

To ensure a common understanding, we explain in this section the key terms and concepts that might not be in the scope of the intended audience of this paper, such as, the "Web Ontology Language" and the "Semantic Web Rule Language".

2.1 Web Ontology Language (OWL)

The Web Ontology Language (OWL) is described in [4] as a "semantic web language designed to represent rich and complex knowledge about things, groups of things, and relations between things". OWL has three sub-languages which vary in complexity and can be used by specific communities of implementers and users. Each of the sub-languages can be described as follows [5]:

- **OWL Lite:** Allows the creation of hierarchies and the addition of simple constraints, such as cardinality constraints with the values 0 or 1. It has a lower formal complexity than the other two languages;
- **OWL-DL:** "Provides maximum expressiveness while retaining computational completeness" [5]. It can use all of OWLs' available constraints, however with restrictions to make it decidable;
- **OWL Full:** Provides maximum expressiveness, however not guaranteeing computability. One of the examples given is for differentiating it from OWL-DL is that in OWL Full it is possible that a class can also be an individual or a collection of individuals.

The terminology used in OWL is different from that used in Description Logics (DL): a DL concept corresponds to an OWL class, a DL role corresponds to an OWL property, and a DL individual corresponds to an OWL object. An OWL-specified ontology is interpreted as a set of "objects" and a set of "properties" which relate objects with each other. Ontologies expressed in OWL consist of axioms that constrain the classes and their relationships. Axioms allow making explicit information that otherwise is implicit through the use of logical inference. Classes are considered to be "the main building blocks of an OWL ontology" [6].

2.2 Semantic Web Rule Language (SWRL)

The Semantic Web Rule Language (SWRL) is an expressive rules language combining Horn clauses with concepts defined in OWL and can be used to increase ability of the inference individuals in a knowledge base in OWL. [7] SWRL rules are composed of two parts: the antecedent (body) and consequent (head). Each rule is an implication between the antecedent and the consequent, which can be understood as: when the preceding conditions are true, then the consequent conditions are also true. Both parts consist of a combination of zero or more atoms, not allowing disjunctions or negation.

Atoms in turn, are formed by a predicate and one or more arguments (the number and type of these arguments are determined by the type of atom which in turn is defined by the predicate used).

Although SWRL rules may be represented in more than one format, the human readable syntax is used in this section. In this format, the arrow (→) is used to separate antecedent and consequent, the caret (^) is the junction between the atoms and the question mark (?) distinguishes the variables of the individuals names. Figure 1 presents a SWRL rule, represented in human readable syntax, highlighting its parts and characteristics.

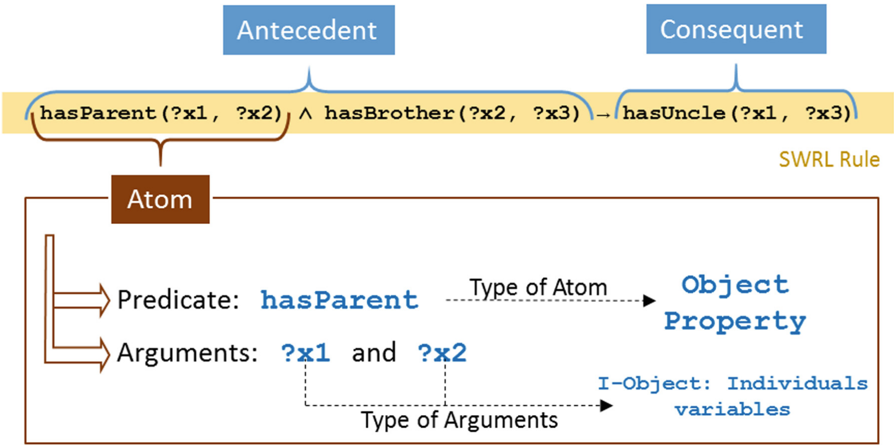

Fig. 1. SWRL rule in human readable syntax and its parts.

Semantically, the above example illustrates that if the parent of an individual has a brother, then this brother is uncle of the individual. The rule has three atoms (all the same type) two being in the antecedent and one in the consequent (which predicates are `hasParent`, `hasBrother` and `hasUncle` respectively) and uses three variables for individuals: `?x1`, `?x2` and `?x3`. Note that the rules are stored in the ontology and the use of this type of annotation allows the inference of new knowledge about individuals, since the rules correspond to conditional statements which, when met, add new information to the knowledge base.

In conclusion, the use of SWRL rules to make semantic annotations as assertions allows to create conditional statements that allows the inference of new knowledge about individuals in ontologies.

3 Usage Scenarios of the ISO/IEC 15504 Formalization

In the intended usage scenario, an organization selects the processes in the scope of the assessment and collects information about its base and generic practices, expresses this information in terms of the ISO/IEC 15504 formalization and then invokes a model interpreter tool to check the consistency of the representation and to derive the capability levels of the organization's processes. While it is possible that some of the practices in an organization have different names than the practices listed in the ISO/IEC 15504 process assessment model, it would be the responsibility of the organization to associate its local practices with the practices recognized in the model. If the model is logically inconsistent, some remedial action would have to be taken to eliminate the source(s) of inconsistency. As was mentioned earlier, in this paper we describe a formalization of the ISO/IEC 15504 model in the Web Ontology Language (OWL) [5], a primary language for the Semantic Web [8]. According to the approach practiced in the Semantic Web, the modeling consists of two phases: (1) The representation of the generic concepts of a domain as an ontology that includes classes, properties (relations) and constraints; and (2) The capturing of the instances of the classes and the properties that are specific to a case being modeled by the ontology. Since we used OWL as the language to formalize the ISO/IEC 15504-5 process assessment model, we followed the same approach as in the Semantic Web. First, we formalized ISO/IEC 15504 as an ontology. This ontology captures the main concepts of this model. Note that the use of the term "ontology" makes use of the interpretation of this notion that is used in knowledge representation [9] and not as it is used in philosophy [10]. We call this formalization the ISO/IEC 15504 Ontology. This ontology is then used to annotate specific and generic practices of a specific organization. In the next step, a generic OWL reasoner, e.g., HermIT [11], FaCT++ [12] or Pellet [13], is used to check the consistency of the representation and then to derive the classification of the capability level of the organization's selected processes.

4 Structure of the ISO/IEC 15504 Ontology

The ISO/IEC 15504 process assessment model is a very complex structure. Its description is provided in natural language text plus some graphics. In some cases, it is supported by figures and tables. The representations, along with textual descriptions, detailed in [3] were helpful in selecting concepts to be captured in the ISO/IEC 15504 formalization. The ontology includes hundreds of classes and nine kinds of properties. Some of the classes are primitive (or declared), i.e., they have some necessary restrictions that need to be satisfied by an individual to be an instance of one of those classes. The defined classes are those that have both necessary and sufficient conditions for an instance to be member of such a class. For these classes, not only an individual needs to satisfy the restrictions of the class definition to be an instance of a given class, but also an OWL reasoner can infer whether a given individual is a member of such a class. This fact is important for the automatic inference of the membership of an organization process in particular capability levels for the processes of the ISO/IEC 15504 process assessment model.

4.1 Top Level Classes and Properties

The goal of the work we describe in this paper was to capture the main parts of the ISO/IEC 15504 model, *i.e.*, identify various concepts and relationships to be represented in the ontology. The top level of the ontology is shown in Fig. 2. Concepts are represented as OWL classes and relationships as OWL properties. Our first decision was to proceed in an incremental top-down approach. In the first step (increment) we have identified six top-level classes: Process_Category, Process_Group, Process, Capability_Level, Process_Attribute, and Rating. All these classes are shown in Fig. 2 as yellow rectangles. The decision to consider these six classes in the ontology was based on the statements in the ISO/IEC 15504 documentation [3]. Process categories are used in the ISO/IEC 15504 as containers for process groups. Process Groups consist of the processes that "contribute to a complementary area" [3]. A process is a "set of interrelated or interacting activities which transforms inputs into outputs" [2]. A process achieves a process capability level which is a "a point on the six-point ordinal scale (of process capability) that represents the capability of the process" [2] A capability level is satisfied by a set of process attributes which represent measurable characteristics of process capability applicable to any process [2]. Each Process_Attribute has a Rating which is "a judgement of the degree of achievement of the process attribute for the assessed process" [2] expressed through a four-point scale. Relations among specific classes are shown in Fig. 2 as arrows with associated labels representing relation names. The top level of the ontology includes six relations, or properties in the terminology of OWL: aggregates, consistsOf, achievesA, characterizedBy, satisfiedBy and hasA. We tried to choose relations names so that they reflect their use within the ISO/IEC 15504 model. The relationships among the six top-level

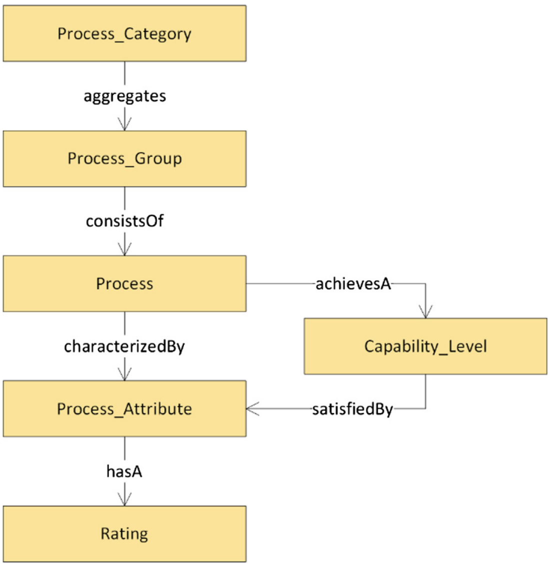

Fig. 2. Top level of the ISO/IEC 15504 ontology

classes are patterned upon the hierarchy detailed in ISO/IEC 15504 which suggests this kind of relationships. The ISO/IEC 15504 model describes relationships between practices, work products, generic resources and outcomes. This aspect was not modeled in our ontology, primarily because our focus at this time was on the ability to infer the capability levels of an organization's processes. The ontology would need to be expanded to capture the outcomes relationships, which would be a trivial task.

4.2 Second Level of the ISO/IEC 15504 Ontology

The second increment of the ontology introduces the classes at one level deeper in the ontology (Fig. 3). Seven new classes, Practice (and two subclasses of Practice), Work Product (and two subclasses of Work Product) and Generic Resource. The subClassOf relation between two classes is represented in Fig. 3 by an arrow with a hollow arrow end pointing to the superclass. We introduced Practice as a superclass of Base_Practice and Generic_Practice. A base practice is "an activity that, when consistently performed, contributes to achieving a specific process purpose" [2], is specific to a certain process and is a process performance indicator that contributes to the achievement of process attribute PA1.1. On the other hand, a generic practice "contributes to the achievement

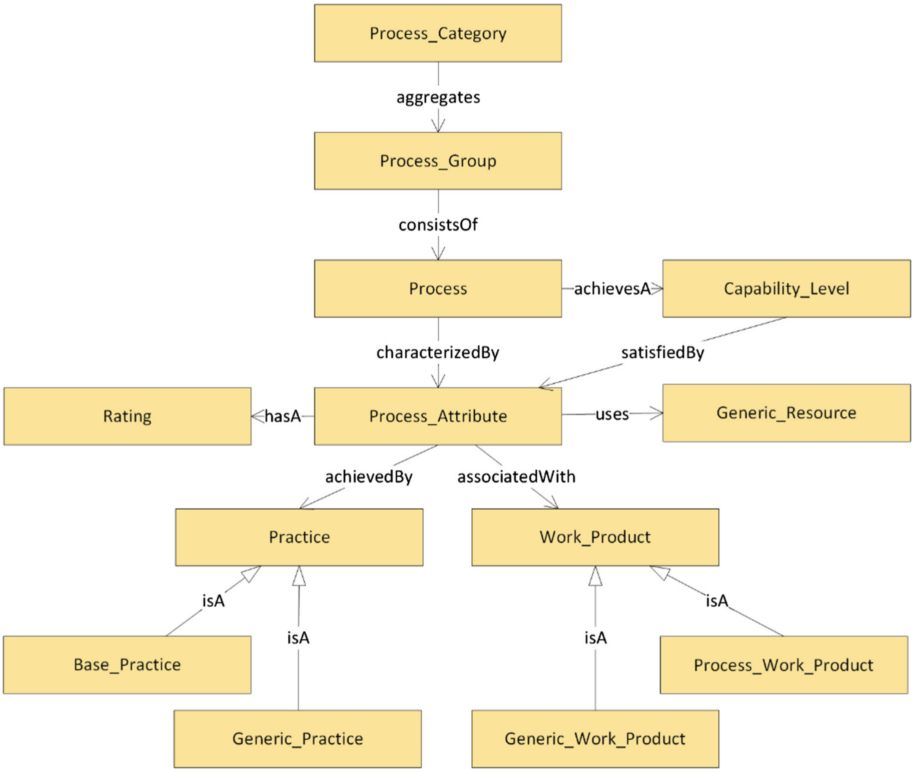

Fig. 3. Second level of the ISO/IEC 15504 ontology.

of a specific process attribute" [2], is generic for any process and is a process capability indicator that contributes to the achievement of the process attributes for capability levels 2 to 5. We also introduced Work_Product as a superclass of Process_Work_Product and Generic_Work_Product. A work product is "an artifact associated with the execution of a process" [2]. A Process_Work_Product is specific to a process, it is a process performance indicator and is detailed in each process description. A generic work product is generic for all processes and is a process capability indicator. Finally, generic resources are "associated resources that may be used when performing the process in order to achieve the attribute" [2], are process capability indicators and are generic to any process. The primary reason for the introduction of these superclasses is to show the commonalities between generic and base practices, and between generic and process Work Products. We also introduced three additional properties between process attribute and practice (achievedBy), between Process_Attribute and Work_Product (associatedWith), and between Process_Attribute and Generic_Resource (uses).

One of the primary considerations behind this work was the automatic inference of the capability levels from the data about process attributes and ratings of a process. As with any formalization, the choice of the formalization language imposes some constraints on what can be represented in the ontology, as well as how it can be done. Since we chose OWL as the formalization language for our ontology we had to construct the ontology in such a way that the automatic inference of capability levels from information about process attributes and ratings is possible. OWL facilitates various kinds of inferences, such as, subsumption [14], satisfiability [15], instance retrieval [16] and type inference [17]. Subsumption reasoning allows the inference that one class is a subclass of another. This inference is based upon the intentional definitions of the classes using primarily property restrictions—defining a class as those individuals that on a given property have values from another class. Satisfiability reasoning allows one to infer whether a proposed type of individual (class) is "satisfiable", i.e., whether it can be instantiated concretely. Instance retrieval allows one to infer which of the individuals are instances of a particular class. Type inference derives the classes that a given individual is an instance of One of the first decisions that we had to make was how the concept of capability level should be represented.

We introduced the Capability_Level class, as shown in Figs. 2 and 3. An OWL reasoner can be used to infer whether the process satisfies which capability levels. Thus the kind of inference used for this purpose is type inference, as described above. For such an inference to be possible, the classes representing particular capability levels must be defined classes. Towards this aim, we defined appropriate restrictions for each Capability_Level subclass. In Fig. 4 we show how restrictions are represented in OWL through Protégé[1].

Due to the size of the ontology, only a part of the restriction on the Capability_Level_4 class is shown. Since the Capability_Level_4 class has a number of restrictions, they are captured as an intersection of particular restrictions. In this case we show that each instance of Capability_Level_4 must have at least one association

1 http://protege.stanford.edu/.

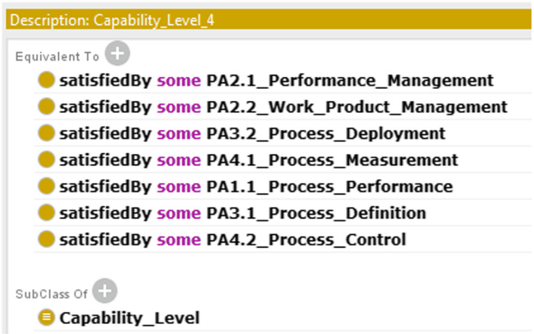

Fig. 4. Restrictions for the Capability_Level_4 class

with PA1.1 to PA4.2 through the satisfiedBy property. In plain English terms, this means that a process to be classified as an instance of Capability_Level_4 must include in its Process Attributes instances of PA1.1-PA4.2 through the characterizedBy property. Additionally to this fact we also designed a set of rules in SWRL so that an OWL Reasoner can infer the capability level. For example, to achieve Capability Level 1 there should be an instance of a process that is characterizedBy PA1.1. This PA1.1 instance must have (hasA) a Rating of L or F, which in SWRL syntax is:

```
Process(?p), PA1.1_Process_Performance(?pa), characterizedBy(?p,
 ?pa), (F-Fully_Achieved or L-Largelly_Achieved)(?r), hasA(?pa,
          ?r) -> achievesA(?p, Capability_Level_1)
```

4.3 Third Level of the ISO/IEC 15504 Ontology

In this increment, the ontology of the ISO/IEC 15504 Process Assessment Model is expanded by providing definitions for the subclasses of Process_Category, Process_Group, Process, Process_Attribute, Generic_Resource, Base_Practice, Generic_Practice, Generic_Work_Product and Process_Work_Product. Due to the relatively large size of the ontology, it is difficult to show it in a graphical form. For this reason, in Table 1 we show the definition of one process subclass, the ACQ.1 (Acquisition Preparation). A full version of this table would show all the subclasses of Process_Category in the first column, then the Process_Group in the second column and finally the Process in the third column. We chose the names of all the subclasses in way that are the identical as defined in the ISO/IEC 15504-5 process assessment model. ACQ.1 shown in Table 1 belongs to the Acquisition Process Group which in turn belongs to the Primary lifecycle processes category. The second column contains the Process_Attribute subclasses that characterize the ACQ.1 Process. The third column contains the Generic Practices and Base Practices subclasses. Following the convention used in the description of the ISO/IEC 15504-5 process assessment model [3], the name of each subclass is prefixed by GP for Generic Practices and BP for Base Practices. Following the prefix of GP, there is a number, which acts as an identifier for the subclasses of Generic_Practice as used in the ISO/IEC 15504-5. So GP1.1.1_Achieve_ the_process_outcomes in Table 1 is the class for the generic practices numbered 1 for the Process Attribute 1.1. The convention for the BP prefix accepted in [3] is that there is an identifier for the process before the BP prefix and an enumerator after the BP prefix. As a result, ACQ.1.BP1_Establish_the_need is the first base practice for the ACQ.1 process.

The "(…)" in Table 1 means that there are more process attributes in the ontology not shown here due to space constraints. The subclasses at a row are related to each other through properties and restrictions. The subclasses in column one are restricted to such instances that on property characterizedBy have at least one (existential restriction) value from a specific subclass of Process_Attribute. In other words, the subclasses of Process are defined by the characterizedBy property and a subclass of Process_Attribute. The Process_Attribute class is defined by the allValuesFrom restriction (necessary and sufficient) on properties achievedBy with values in the class Practice. However, a relatively rich subclassification of Process_Attribute provides more information than this restriction. Each subclass of Process_Attribute is defined as an existential restriction to particular subclasses of Practice on property achievedBy.

So for instance, PA1.1_ACQ.1_Process_Performance must have at least one of the six practices as value of achievedBy. In total, there are 57 Process_Attribute subclasses, 9 subclasses for each of the Process Attributes and then 48 subclasses for the PA1.1 process attributes, one for each process detailed in ISO/IEC 15504-5, this is because PA1.1 process performance indicators are based on the base practices and work products of each of the processes, this results in a subclass of PA1.1 for each process. Not shown in Table 1, beyond the Practices, there are also work products and generic resources associated with all the process attributes although these are not shown here due to space constraints. The work products follow the same convention as in ISO/IEC 15504-5, an example of a subclass of Generic_Work_Product is "05-00_Goals", and an example of a subclass of Generic_Resource is "GR3.1.4_Process_infrastructure" which is the fourth generic resource for PA3.1.

5 Validation of the Formalization

In the previous sections we showed how the ISO/IEC 15504 ontology presented in this paper was constructed. The main purpose of this discussion was to show the relation between the ISO/IEC 15504 process assessment model described in [3] and the ontology, and show that the ontology is a relatively faithful formalization of the model. Our main goal was to convince the reader that this is actually the truth. Obviously, this is a subjective judgment. Since, as stated in Sect. 1, the main usage scenario for this ontology that guided its development was the automatic inference of the capability levels of an organization's processes, we also tested this formalization on a number of cases. For this purpose, a set of test cases were developed and then OWL inference engines were used for the automatic inference of facts entailed by the ontology. In particular, the derivation of the capability level of a set of processes was demonstrated. In this section we describe some of our experiments.

Table 1. Example relationships among subclasses of Process, Process_Attribute and Practice.

Process	Process_Attribute	Practice
ACQ.1_ Acquisition_ Preparation	PA1.1_ACQ.1_ Process_Performance	ACQ.1.BP1_Establish_the_Need
		ACQ.1.BP2_Define_the_Requirements
		ACQ.1.BP3_Review_Requirements
		ACQ.1.BP4_Develop_Acquisition_Strategy
		ACQ.1.BP5_Define_Selection_Criteria
		ACQ.1.BP6_Communicate_the_Need
	PA2.1_Performance_ Management	GP2.1.1_Indentify_the_Objectives
		GP2.1.2_Plan_and_Monitor_the_Performance
		GP2.1.3_Adjust
		GP2.1.4_Define_Responsabilities_and_Authorities
		GP2.1.5_Identify_and_Make_Available_Resources
		GP2.1.6_Manage_the_Interfaces
	PA2.2_Work_ Product_Management	GP2.2.1_Define_the_Requirements_for_the_Work_Products
		GP2.2.2_Define_thee_Requirements_for_Documentation_ and_Control
		GP2.2.3_Identify,_Document_and_Control
		GP2.2.4_Review_and_Adjust_Work_Products
	PA3.1_Process_ Definition	GP3.1.1_Define_the_Standard_Process
		GP3.1.2_Determine_the_Sequence_and_Iteraction
		GP3.1.3_Identify_the_Roles_and_Competencies
		GP3.1.4_Identify_the_Required_Infrastructure_and_ Work_Environment
		GP3.1.5_Determine_Suitable_Methods
	(…)	
	PA4.1_Process_ Measurement	GP4.1.1_Identify_Process_Information_Needs
		GP4.1.2_Derive_Process_Measurement_Objectives
		GP4.1.3_Establish_Quantitative_Objectives
		GP4.1.4_Identify_Product_and_Process_Measures
		GP4.1.5_Collect_Product_and_Process_Measurement_Results
		GP4.1.6_Use_the_Results_of_the_Defined_Measurement
	(…)	
	PA5.1_Process_ Innovation	GP5.1.1_Define_the_Process_Improvement_Objectives
		GP5.1.2_Analyse_Measurement_Data
		GP5.1.3_Identify_Improvement_Opportunities
		GP5.1.4_Derive_Improvement_Opportunities
		GP5.1.5_Define_an_Implementation_Strategy
	(…)	

5.1 Test Data

In order to attain a capability level for a certain process, an organization should have the appropriate Ratings for the Process Attributes that Capability Level consists of as depicted in Figs. 2 and 3. For our experiments, five organizations were created including the assessment results. The assessment results for the five organizations and the ratings for the process attributes for the processes contained in the acquisition process group (ACQ) are shown in Table 2. The organizations in Table 2 are labeled O1 through O5. There is a row for each organization, then each of these rows are decomposed into the processes assessed in the scope of the ACQ process group. Then for each process there is a column for each of the process attributes, in the intersection between a process and process attribute will be the rating for that specific process attribute, and in the last column is the achieved capability level (CL) for that specific process and organization. All the data from Table 2 was annotated in terms of the ISO/IEC 15504 ontology so that it could be processed by an OWL reasoner. Moreover, instances of all the classes from the ISO/IEC 15504 ontology (Base_Practice, Generic_Practice, Process_Work_Product, Generic_Work_Product and Generic_Resource) had to be created. Those instances need to be present for a particular organization in order to satisfy the restrictions of the process attribute that the organization has been assigned.

Table 2. Test cases.

Org.	Process	PA1.1	PA2.1	PA2.2	PA3.1	PA3.2	PA4.1	PA4.2	PA5.1	PA5.2	CL
O1	ACQ.1	F	F	F	F	F	F	F	P	L	4
	ACQ.2	F	F	F	F	F	F	F	F	F	5
	ACQ.3	F	F	F	F	F	L	L			4
	ACQ.4	F	L	L							2
	ACQ.5	F	L	L							2
O2	ACQ.1	F	F	F	L	L					3
	ACQ.2	F	F	F	L	L					3
	ACQ.3	**Out of Scope**									
	ACQ.4	L									1
	ACQ.5	L									1
O3	ACQ.1	F	F	F	F	F	F	F	L	P	4
	ACQ.2	F	F	F	F	F	L	L			4
	ACQ.3	F	F	F	F	F	N	L			3
	ACQ.4	F	F	F	L	L					3
	ACQ.5	**Out of Scope**									
O4	ACQ.1	F	F	F	F	F	L	L			4
	ACQ.2	F	F	F	F	F	P	L			3
	ACQ.3	F	F	F	P	L					2
	ACQ.4	L									1
	ACQ.5	L									1
O5	ACQ.1	F	F	F	F	F	L	L			4
	ACQ.2	F	F	F	L	L					3
	ACQ.3	**Out of Scope**									
	ACQ.4	F	L	P							1
	ACQ.5	L									1

5.2 Testing

As the first step, both the ISO/IEC 15504 ontology and the OWL files that contained the test cases were checked for consistency using an ontology consistency checker HermIT [11]. The result of the tests on the final version of these files was that all the ontologies were consistent. In the next step, three OWL reasoners were used to derive selected processes capability levels of the five organizations. The reasoners were HermIT [11], FaCT++ [12] and Pellet [13]. All three inference engines are available for free for research purposes. All of them were able to process the reasoning tasks for all five cases within seconds. All the test cases resulted in correct (expected) inference. In other words, for all the five test cases listed in Table 2, HermIT, FaCT++ and Pellet derived that the organizations satisfied the levels specified in the table, as well as all the levels below the highest level.

6 Conclusions

The main purpose of the work described in this paper was to demonstrate the capability of automatic classification of capability levels based upon some characteristics of the processes used by an organization. Towards this aim, a comprehensive formalization of the ISO/IEC 15504 model as an ontology was implemented in OWL-DL. The ontology includes hundreds classes and nine properties. The ontology has been validated five test cases. In order to annotate these organizations, a large number of instances had to be added to the base ISO/IEC 15504 ontology. The number of instances depend on the processes in the scope of the assessment. The HermIT, FaCT++ and Pellet inference engines were used successfully to derive the capability levels of the organizations based on the supplied data, and for all of the test cases the inference engines derived the same conclusions as defined when the test cases were defined. Since the ISO/IEC 15504 Ontology has computer-executable semantics, it can be used for automatic reasoning about the capability levels of organizations' processes, based upon some data provided by the organization. An OWL reasoner can be used for this purpose. The ontology could also be used in other scenarios, including process improvement and process optimization. And finally, the ontology can be used for passing information about particular aspects of processes. Although the validation results indicate that the ontology faithfully captures the concepts and constraints of the ISO/IEC 15504 model, the ultimate value of the ontology can only be appreciated if the community accepts it and decides to use it for both capturing process information and for the interchanging of information among various tools and various users. This ontology can be seen as a "core ontology" that can be extended to more fully capture the concepts that are needed by the potential users.

Acknowledgements. This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

References

1. SPICE Project Organization, Software Process Assessment (SPICE). http://www.sqi.gu.edu.au/SPICE/
2. ISO/IEC 15504-1:2004, Information technology - Process assessment – Concepts and Vocabulary, International Organization for Standardization and International Electrotechnical Commission Std. (2004)
3. ISO/IEC 15504-5:2012, Information technology - Process assessment – An exemplar Process Assessment Model, International Organization for Standardization and International Electrotechnical Commission Std. (2012)
4. W3C, OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition), World Wide Web Consortium Recommendation (2012)
5. W3C, OWL Web Ontology Language Semantics and Abstract Syntax, World Wide Web Consortium Recommendation (2004)
6. Horridge, M.: A practical guide to building OWL ontologies using protégé 4 and CODE tools, technical report, The University Of Manchester (2011)
7. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C. http://www.w3.org/Submission/SWRL/
8. W3C, Semantic Web Activity, World Wide Web Consortium Recommendation (2006)
9. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquisition **5**(2), 199–220 (1993). doi:10.1006/knac.1993.1008
10. Bunge, M.: Treatise on Basic Philosophy: Ontology I: The Furniture of the World. Springer, Dodrecht (1977)
11. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: an OWL 2 reasoner. J. Autom. Reasoning **53**, 245–269 (2014)
12. Tsarkov, D., Horrocks, I.: FaCT++ Description logic reasoner: system description. In: Proceedings of the Third International Joint Conference, USA, pp. 292–297 (2006)
13. Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., Katz, Y.: Pellet: a practical owl-dl reasoner. In: Web Semantics: Science, Services and Agents on the WWWs, vol. 5, pp. 51–53 (2007)
14. Vaculin, R.: Process Mediation Framework for Semantic Web Services. Ph.D. thesis, Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics, Charles University (2009)
15. Lemaignan, S.: Grounding the Interaction: Knowledge Management for Interactive Robots. Ph.D. thesis, Universite de Toulouse (2012)
16. Areces, C.: Logic Engineering. The Case of Description and Hybrid Logics. Ph.D. thesis, Institute for Logic, Language and Computation, University of Amsterdam (2000)
17. Baader, F., Horrocks, I., Sattler, U.: Description logics. Found. Artif. Intell. **3**, 135–179 (2008)

A Model-Driven Proposal to Execute and Orchestrate Processes: PLM_4BS

Julián Alberto Garcia-Garcia(✉), Ayman Meidan,
Antonio Vázquez Carreño, and Manuel Mejias Risoto

Web Engineering and Early Testing (IWT2) Group,
Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla,
Avda Reina Mercedes s/n., 41012 Seville, Spain
{julian.garcia,antonio.vazquez}@iwt2.org,
ayman.meidan@gmail.com, risoto@us.es

Abstract. Business Processes Management (BPM) is a widely consolidated business strategy to improve and optimize the internal operation of any company. However, BPM is not usually simple to apply in software organizations because Software Processes (SPs) involve high degree of creativity, abstraction and rework, among other aspects. This situation provokes that these companies usually focus on modeling their processes but later, the orchestration and execution are manually and/or unilaterally performed by each involved role. This situation makes each SP difficult to maintain, monitor, evolve and measure. At present, there are model-based proposals to model SPs, but most of them fail to define the execution context of the process. This paper presents PLM_4BS, a model-driven framework to support modeling, execution and orchestration of SPs. It has been successfully validated in different real environments, what has returned us valuable feedback to improve PLM_4BS in the near future.

Keywords: Business Processes Management · Model-Driven Engineering · Execution and orchestration of processes

1 Introduction

It is a worldwide accepted knowledge that in the last years, Business Process Management (BPM) [1] has become a suitable strategy to increase excellence and productivity in any kind of organization. BPM tries to strategically assess processes and improve their effectiveness and efficiency within the organization with the aim to reduce costs and improve quality, productivity and competitiveness in relation to other organizations of the same business area.

Model Driven Engineering (MDE), aims to raise the level of abstraction in program specification and increase automation in program development. The idea promoted by MDE is to use models at different levels of abstraction for developing systems, thereby raising the level of abstraction in program specification.

However, although BPM has been successfully applied to many kinds of organizations, there are difficulties in software companies because of some special features of the software process. In [3], the authors identify and describe properties that characterize

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 211–225, 2017.
DOI: 10.1007/978-3-319-67383-7_16

software processes in comparison with other processes (e.g., industrial processes) such as: (*i*) they are constantly evolving, as they usually incorporate new lifecycles and technologies and they frequently comprise several iterations that produce different software products versions; (*ii*) they are complex because they are strongly influenced by many unpredictable circumstances and many work teams; and (*iii*) they often rely on communication, coordination and cooperation of different frameworks and development technologies as well as on the different roles they play.

These features frequently provoke that BPM is not properly applied to software organizations that usually and justly focus on defining their processes, forgetting the process execution because most of the activities cannot be easily and effectively automated [4]. Once the process is defined, each involved role performs the process execution [5] and orchestration [6] manually and/or unilaterally. This statement describes a real situation that many software organizations are facing in their day-to-day lives. In fact, our research group has obtained this useful feedback from many partners (international and Spanish software companies) after carrying out many R&D projects.

The situation described previously poses other collateral problems. For instance, it may involve difficulties to manage, maintain, monitor, evolve and measure processes.

This paper aims to propose a MDE-based solution to support process execution and orchestration. For this purpose, this paper extends another previous paper [7] in which a process definition metamodel is presented within PLM_4BS (Process Lifecycle Management for Software-Business) framework. PLM_4BS is based on a continuous improvement lifecycle in order to manage the software process. This lifecycle defines four phases (modeling, execution and orchestration, monitoring and continuous improvement), although hitherto, PLM_4BS only supported the modeling phase [7].

This paper uses MDE (Model-Driven Engineering) [8] to integrate the execution and orchestration of processes within PLM_4BS because: it (*i*) is one of the most entrenched paradigms within software engineering area; and (*ii*) suitable results have been achieved when MDE has been applied to real environments (e.g., testing [13], healthcare environments [17] or Web engineering [16, 29], among others).

To achieve the aforementioned goals, this paper defines: (*i*) a specific process execution and orchestration[1] metamodel that lets specify the executable context of software processes after defining the process in the modeling phase; and (*ii*) a systematic and automatic protocol that makes it possible to generate executable code from an execution and orchestration model. This executable code is based on two standards: WS-BPEL (Business Process Execution Language for Web Services [9]) and XMI BPMN 2.0 format [10]. Both standards have been chosen because they are supported by most process engines (named BPMS or BPM Suite) according to conclusions obtained from different studies, such as [11]. This way, we are able to improve applicability of PLM_4BS to real environments since if an organization wants to use our proposal, it does not need to change its BPMS.

[1] The process orchestration is understood in this paper as the centralized coordination of events that allows conditioning the evolution and execution of process flow.

This paper is organized as follows: after this introduction, Sect. 2 analyzes the related work on model-based proposals to execute and orchestrate processes. Section 3 introduces our background and Sect. 4 describes our model-driven solution to execute and orchestrate processes. Finally, Sect. 5 presents some discussions, conclusions and future work.

2 Related Work

After modeling a process using a specific Process Modeling Language (PML) [18], it is necessary to define the execution context in order to perform and orchestrate it. The scope of this paper is framed into a model-based PML that includes mechanisms for defining this execution context. Nowadays, there are few proposals with some degree of executability. UML_4SPM [19] is a MOF-compliant metamodel to model software processes. Authors also propose to combine UML_4SPM and BPEL in order to execute the process.

Ferreira's proposal [20] consists in an UML-based modeling language to design software processes and a set of transformations rules to transform these UML process models into executable code. This code is implemented conform to Little-JIL [21], which is an ad-hoc, executable, programming graphical language to coordinate and run tasks among autonomous systems.

Di Nitto et al. [22] suggest an UML1.3-based framework to model SPs. However, they neither extend the UML metamodel or stereotypes nor introduce new concepts. The authors define transformation rules of a small subset of UML to generate executable workflow models. Later, these models can be deployed in an ad-hoc workflow management system [23] developed by these authors.

Chou's approach [24] uses activity diagrams of UML1.4 to model processes and establish theoretical transformation rules to generate executable code from activity diagrams. This code is implemented following an ad-hoc object-oriented programming language. The main disadvantage of this approach is the lack of an automatic generation of code from activity diagrams, what provokes that developers have to rewrite their software applications according to Chou's language.

Moreover, there is a standard proposal focused on software domain: SPEM2.0 [25]. SPEM2.0 is a standard that describes an UML-based metamodel that is used to define software development processes and software systems. However, SPEM2.0 does not provide mechanisms to execute the process. For this reason, Bendraou et al. propose xSPEM (eXecutable SPEM) [26], which provides a definition of an executable SPEM based on Petri-net. xSPEM adds some features to model and store states of the process when this one is executed.

3 Background

This section describes the context of PLM_4BS and its architecture (Sect. 3.1), which is based on MDE and a complete lifecycle. This paper focuses on supporting the second phase of the lifecycle of PLM_4BS (named "Execution and Orchestration Phase").

However, our proposal uses defined information in the first phase (named "Modeling Phase"). Consequently, a brief description of this first phase is also presented as background (Sect. 3.2).

3.1 MDE-Based Architecture of PLM$_4$BS

BPM can be considered a management strategy with a clear multidisciplinary nature that has conditioned the appearance of different views, definitions and perspectives of the process lifecycle and continuous improvement. However, orchestration of processes is an aspect that has not been clearly defined [1,27,28]. This is relevant because over the last decade, more companies used different interconnected tools to run their processes [30]. Therefore, it is necessary and important to support this feature, in a theoretical way, in the continuous improvement lifecycle of processes.

Considering the aforementioned arguments, the architecture of PLM$_4$BS includes a BP lifecycle comprising four phases: (1) modeling, (2) execution and orchestration, (3) monitoring and (4) continuous improvement. They are integrated within PLM$_4$BS using the MDE paradigm in order to take advantage of the benefits this paradigm entails [8].

Figure 1 shows conceptually this lifecycle of PLM$_4$BS as well as the phases that are completely and incompletely defined at present. The former (i.e., completely defined phases) are represented using a continuous line (these are: *(1) modeling* and *(2) execution and orchestration*) whereas the latter are represented by means of dashed lines (these are: *(3) monitoring* and *(4) continuous improvement*). It is important to point out that this paper is focused on describing the second phase of our lifecycle (i.e., execution and orchestration phase). The modeling phase (the first one) is briefly described in Sect. 3.2 as background because it constitutes an input to the execution and orchestration phase. Moreover, the third and fourth phases are conceived as future work, even though we are currently working on them. Finally, the phases of our process improvement lifecycle are further described below:

1. **Modeling Phase.** At this phase, the process engineer is able to model and describe his/her processes in a structured manner. PLM$_4$BS proposes a simple, flexible and highly semantic metamodel to support this phase. It is explained in Sect. 3.2 and takes the form of a MOF-compliant metamodel.
2. **Execution and Orchestration Phase.** Today, this phase is critical and essential since companies are being driven by the need to extensively automate their processes to execute and orchestrate them with EMS (Enterprise Management Systems). At this phase, the process defined by the process engineer at the previous phase must be executed and orchestrated in a BPMS. For this purpose, the process engineer must specify execution parameters as well as parameters for the communication and integration with external systems.

Nevertheless, most BPMSs have inflexible PMLs, that is, these tools do not allow executing processes that have been defined following other PMLs [11]. To solve this situation, PLM$_4$BS provides MDE mechanisms based on three steps.

On the one hand, PLM$_4$BS defines an ***execution and orchestration metamodel*** that defines execution parameters to run the process into a BPMS. Any instance of this

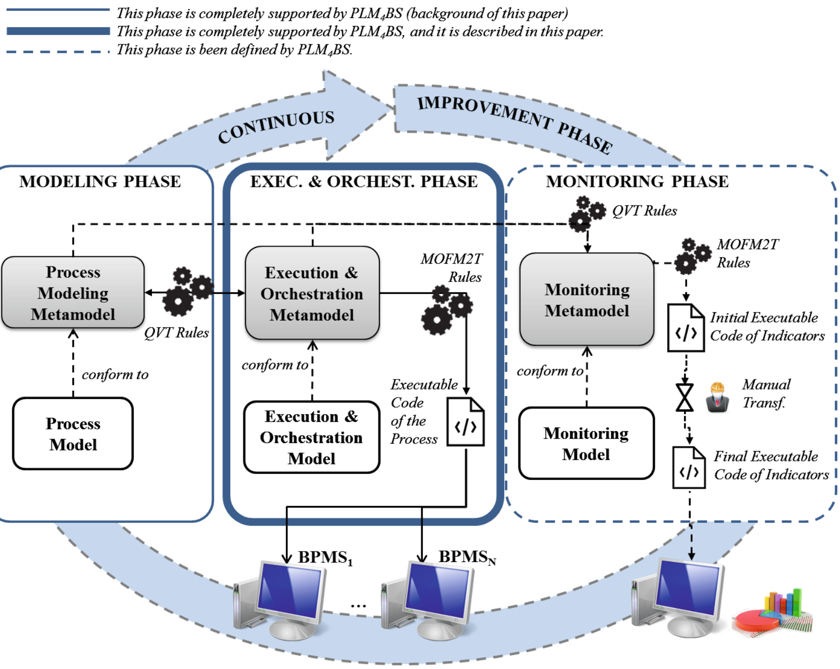

Fig. 1. Theoretical architecture of PLM_4BS based on MDE and a continuous improvement lifecycle of processes. This paper focuses on describing in detail how PLM_4BS supports the execution and orchestration phase (second phase).

metamodel is systematically obtained using model-to-model (M2 M) transformation rules from the process modeling metamodel (modeling phase). Section 4 describes some of these rules, which are formalized using QVT Query/View/Transformation [31].

On the other hand, a ***systematic and automatic transformation protocol*** has been defined to generate executable code from the mentioned execution metamodel. This protocol is based on model-to-text (M2T) transformation rules using MOFM2T [32]. The process engineer should be able to instance and run processes into any process engine when the process execution context is defined.

All these mechanisms to support the execution and orchestration phase are explained in detail in Sect. 4.

3. **Monitoring Phase.** Once the process is deployed into a BPMS, it is time to evaluate its effectiveness. This evaluation provides a granular view of the overall productivity of each process and it is based on the definition of key performance indicators.

In this case, PLM_4BS provides two types of mechanisms to back up this phase. Firstly, the process modeling metamodel includes concepts (such as metric and indicator) that help the process engineer measure processes. Indicators are defined during the modeling phase. Secondly, and after identifying each indicator, PLM_4BS will define a monitoring metamodel that will include elements to allow the process measurement.

PLM_4BS plans to generate this monitoring metamodel from metamodels defined into previous phases. For this purpose, a set of M2 M transformation rules will be defined in PLM_4BS. Finally, a set of M2T transformation rules will be also defined in order to generate a measurement database and code scripts to manage each defined indicator. At present, this phase is not fully supported, thus, we are researching into different alternatives.

4. **Continuous Improvement Phase.** Finally, after evaluating processes performance (through assessment indicators and metrics), an organization should start an internal improvement process to achieve higher quality, efficiency, effectiveness and performance levels during processes execution. If necessary, the organization can iterate over our BPM lifecycle as many times as necessary in order to achieve business goals.

3.2 Metamodel to Support the Modeling Phase

PLM_4BS proposes a flexible and highly semantic metamodel (based on UML2.5) to support the modeling phase. This metamodel takes the form of a MOF-compliant metamodel and follows the guidelines defined in ISO/IEC TR 24744 standard [33]. Our Process Modeling Metamodel (PMM) will not be explained here, since it is out of the scope of this paper and it would become too extensive (a complete description can be found in [34]). However, a brief description of the main metaclasses of PMM is described below (Table 1). These metaclasses are the most important metaclasses to obtain the Process Execution and Orchestration Metamodel (PEOMM). It is also worth highlighting that our PMM contains other metaclasses (such as «Product», «Stakeholder» or «Indicator», among others), but they are not relevant for this paper.

Table 1. Main metaclasses of the modeling metamodel of PLM_4BS.

Metaclass	Meaning
Process	It represents any process that is composed of a set of ordered elements (i.e., *«ProcessElements» metaclass* linked themselves) to produce products («Product» metaclass)
ProcessElements ControlElement Activity HumanActivity OrchestrationActivity ComplexActivity	It represents any element of the process workflow and has been specialized in two metaclasses: *«ControlElement»* and *«Activity»* The former defines elements that allow establishing the process structure using different kinds of control elements: *«InitialElement»* or *«FinalElement»* (i.e., the first or last activity); *«Conditional»*, which enables creating disjoint branches of the workflow; and *«Fork»* or *«Join»*, which allow starting and ending parallel branches of the workflow The latter represents an action that should be executed to develop the process and has been grouped into three metaclasses: *«OrchestrationActivity»*, which represents an orchestration activity (i.e., an activity performed by a machine); *«ComplexActivity»*, which allows including a process within another process; and *«HumanActivity»*, which represents an activity that someone performs

4 A Model-Driven Solution to Execute and Orchestrate Processes

This section presents a model-driven solution to support the execution and the orchestration of processes into PLM_4BS. For this purpose, our solution is composed of: (*i*) a Process Execution and Orchestration Metamodel (PEOMM); and (*ii*) transformation rules to generate PEOMM from the modeling phase and, later, generate executable code. Both aspects are described in Sects. 4.1 and 4.2, respectively.

4.1 Defining a Metamodel to Support Execution and Orchestration

PEOMM has been defined with more granularity (i.e., lower level of abstraction) than PMM, which was briefly introduced in Sect. 3. PEOMM has the form of a MOF-compliant metamodel and incorporates required attributes normalized according to ISO/IEC TR 24744 [33]. PEOMM aims to represent the process structure from the point of view of its execution context.

For this purpose, it provides a complete and theoretical specification to allow executing process models within BPMS. PEOMM (Fig. 2) also describes static and dynamic semantics of this execution. On the one hand, static semantics refers to: (*i*) static information (i.e., specific properties or attributes) of each concept defined in PEOMM; and (*ii*) semantic constraints to ensure building well-formed models. On the other hand, dynamic semantics refers to: (*i*) what information is generated and managed in a dynamic manner (i.e., at runtime); and (*ii*) how and when each element of PEOMM can be instantiated. This semantics enables each element to react and evolve at runtime along its own lifecycle.

Before going further, it is worth clarifying that the syntax used is not enough to semantically define our metamodel. Consequently, we have used OCL [2] (as recommended by OMG) to add formal constraints, which, in turn, limit possible instantiations and therefore valid process models. All our OCL constraints will not be explained here, since they are out of the scope of this paper and it would become too extensive. Nevertheless, as an illustrative example, just a couple of OCL constraints will be explained in detail.

The main metaclass in PEOMM is the *«ExecutionNodeClass» metaclass,* which represents the executable view of any element in the process. These elements are interrelated to build the process execution flow. Such relationships are modeled by means of the *«ExecutionFlow» metaclass*.

The *«ExecutionNodeClass»* metaclass has two kinds of properties: static properties (i.e., *«name»*, *«description»* and *«isInitial»*) and dynamic properties (i.e., *«status»*). The *«status»* property establishes the lifecycle of each executable element. This lifecycle is composed of five allowed status whose transitions are formalized using a state machine (Fig. 3). Subsequently, each transition is triggered by one unique operation of the *«ExecutionNodeClass»* metaclass. These operations (see Fig. 2) are stereotyped as: (*i*) *«CONTR»* (CONSTructor), which defines the constructor to create a new instance of the *«ExecutionNodeClass» metaclass*; or (*ii*) «*SOP*» (Status OPeration), which identifies operations to update or query the internal status of the executable element.

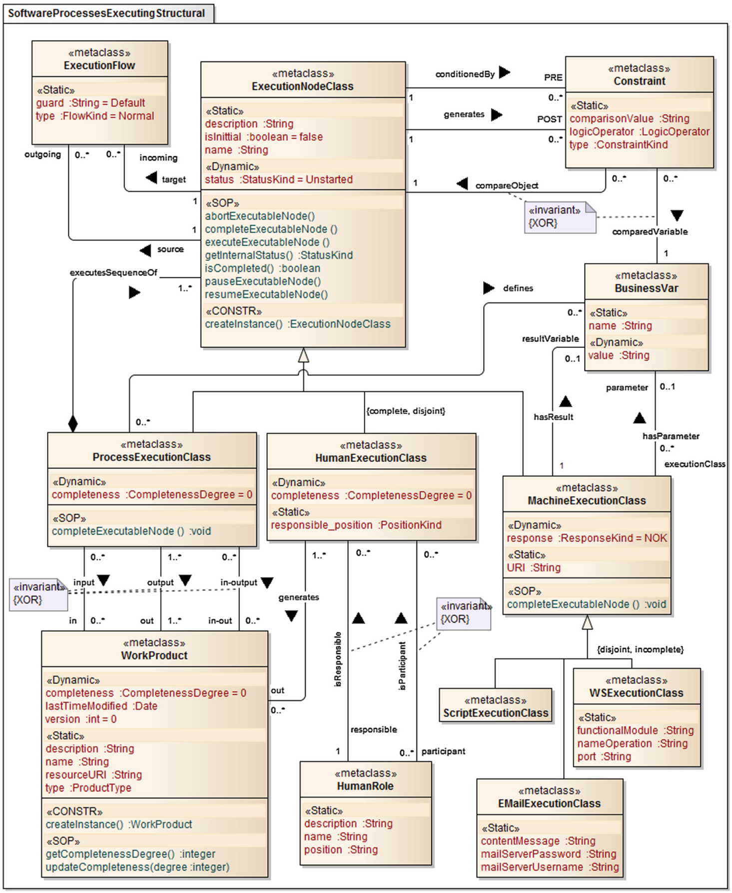

Fig. 2. Metamodel to support the execution and orchestration of processes

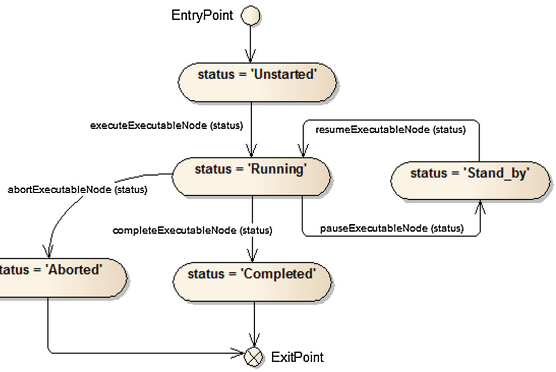

Fig. 3. Lifecycle of the *«ExecutionNodeClass» metaclass*

Each operation is formally defined by means of an object-oriented language proposed by UML to define executable semantics [15]. Table 2 shows just a couple of operations since explaining all of them is out of the scope of this paper and it would become too extensive. Operations shown in Table 2 are: *(i) «createInstance»*, which defines how an execution element is instanced; and *(ii) «executeExecutableNode»*, which allows running the execution of an executable element.

Table 2. Some operations of the *«ExecutionNodeClass» metaclass*

```
createInstance (String name,
                String descp, boolean isInittial)  {
     this.name = name;
     this.description = descp;
     this.isInittial = isInittial;
     this.status = "Unstarted";
}
```

```
executeExecutableNode ()  : void   {
     if (this.status.equals ("Unstarted") ||
         this.status.equals ("Stand_by"))
             this.status = "Running";
}
```

The *«ExecutionNodeClass» metaclass* is also specialized in three metaclasses in order to distinguish among different executable elements:

- The *«ProcessExecutionClass»* metaclass. It represents the highest-level execution entity in any process that defines the execution structure of a process. In this sense, the *«ProcessExecutionClass» metaclass* is composed of a set of execution nodes (i.e., *«ExecutionNodeClass»*). This metaclass also includes the *«completeness»* property, which aims to indicate the degree of completeness of the process. This property is also used to establish when the executable process is completed, that is to say, an executable process is completed as follows: *(i)* when its internal status is *«Running»* (this condition is already checked in the *«ProcessExecutionClass»* super-metaclass); and (*ii*) when its degree of completeness is 100%.
- The *«MachineExecutionClass»* metaclass. It is very important because it allows orchestrating and defining the coordination of events among information systems during the process execution. For this purpose, this metaclass includes one static property (*«URI»*, which is the uniform resource identifier of the target system) and one dynamic property (*«response»*, which stores the response code after performing the orchestration activity). Moreover, the *«MachineExecutionClass»* metaclass has been also specialized in three metaclasses in order to distinguish among different automatic events: *«ScriptExecutionClass»*, which allows executing script code; *«WSExecutionClass»*, which makes it possible to invoke Web services; and *«EMailExecutionClass»*, which allows defining notification via email.
- The *«HumanExecutionClass»* metaclass. It represents any executable element that must be performed by human agents. These agents are modeled into PEOMM with

the *«HumanRole»* metaclass, which represents an actor who can be participant or responsible, but cannot perform both roles (PEOMM shows this semantic constraint using the *«XOR»* logical operator between the *«isParticipant»* and *«isResponsible»* associations). Both roles have been considered because many companies (e.g., software companies) have many work teams where each member cooperates in developing activities depending on the involvement degree. The difference among these roles is that participants can complete some aspects of the product, but they cannot complete the activity (this action is only available for responsibles).

So far, this paper has explained how PEOMM defines the internal semantics of each *«ExecutionNodeClass» metaclass* using a status machine whose transitions are triggered by operations. Nonetheless, it is also important to establish what conditions have to be true before executing each *«ExecutionNodeClass» metaclass* and after running them. The former are named pre-conditions. They are used to capture a conjunction of events that lead to the execution of an *«ExecutionNodeClass»* and allows defining conditions such as, «*a specific activity may not be executed until either the previous activity is completed or a specific business rule is true*». The latter are named post-conditions and they allow defining conditions such as, «*after executing the current activity, the output work products must have been completed*».

PEOMM models these pre- and post-conditions with the *«Constraint» metaclass*. This metaclass enables comparing (with logic operators) a specific comparative value either of business variables or the internal status of *«ExecutionNodeClass»*. The concept of business variable is key in PEOMM because it helps store values and results used in orchestration activities. The *«BusinessVar» metaclass* models business variables in PEOMM.

Finally, PEOMM also takes into account the dynamic behavior of the results (*«WorkProduct» metaclass*) of a software process because they may evidence the process completion in, e.g., audits. In addition, the products evolve during the process execution, (i.e., its version or finishing percentage, for instance).

After introducing the previous metaclasses, Table 3 describes one of the most important OCL constraints to build well-formed execution models. This constraint is defined at the *«ProcessExecutionClass» metaclass* and checks three conditions: (*i*) a process cannot contain itself, in order to avoid an indefinite execution model because of recursive definitions; (*ii*) each executable process can only contain one executable node typed as initial; and (*iii*) each work product should have been generated by an instance of the *«HumanExecutionClass» metaclass,* which should also belong to the *«ProcessExecutionClass» metaclass.*

4.2 Defining a Transformation Protocol

The architecture of PLM_4BS (Sect. 3.1) considers the use of MDE to obtain the PEOMM and its executable version of the process from PMM. For this purpose, PLM_4BS defines a transformation protocol based on three steps:

1. **Generating systematically the basic Process Execution and Orchestration Model (PEOM).** The basic PEOM is considered the first version of PEOM and it is systematically obtained from the process modeling model using a comprehensive

Table 3. OCL constraint of the *«ProcessExecutionClass» metaclass*

```
context ProcessExecutionClass inv:
  let activitiesColl : SortedSet =
    self.ExecutionNodeClass→select(oclIsTypeOf(HumanExecutionClass))
  in
  self.ExecutionNodeClass→
       select(oclIsTypeOf(ProcessExecutionClass))→count(self)=0
  and
  self.ExecutionNodeClass→
       iterate(node: ExecutionNodeClass; acc:Integer=0 |
               if node.isInittial then acc + 1 else acc endif)=1
  and
  self.out→forAll (p1 : WorkProduct|activitiesColl→
                    exist(a:HumanExecutionClass |
                          a.WorkProduct→exist(p2:Product|p1=p2)))
  and
  self.out→size() = activitiesColl→
       iterate(a : HumanExecutionClass;
               acc : Integer=0 | acc+ a.Product→size());
```

set of M2 M transformation rules. Table 4 describes the *«toHumanExecution Class»* rule using QVT (the others can be found in [34]). This rule describes how *«HumanExecutionClass»* (PEOMM) is obtained from *«HumanActivity»* (PMM). Firstly, this rule initializes all static and dynamic properties (line 3 and 4) of the *«HumanExecutionClass» metaclass* and resolves all relationships with *«WorkProduct»* and *«HumanRole» metaclasses* (lines 6–9). This QVT rule also uses some auxiliary functions: *«isInittialActivity»*, which checks if the *«HumanActivity» metaclass* is the first metaclass in the process; *«createPreConditions»* and *«createPosConditions»*, which elaborate the constraints associated with the *«HumanExecutionClass»*.

2. **Generating manually the final PEOM.** Once the previous step is carried out, the process engineer can add his/her knowledge to the basic PEOM in order to complete the execution context. This unsystematic and manual transformation generates the final version of PEOM. At this point, it is important to highlight that, if the process engineer detects deficiencies in the structure of the execution model, these changes must be extended to the process modeling model in order to avoid inconsistency among models. This procedure can be indefinitely repeated in order to achieve a coherent execution and organization model.
3. **Generating executable code.** Finally, PLM_4BS defines a comprehensive set of M2T transformation rules to obtain executable code from the final PEOM. These rules have been defined in MOFM2T and allow generating WS-BPEL code that can be executed in most BMPS [11]. Table 5 describes the *«defineBPELStructure»* rule using MOFM2T (the others can be found in [17]). This rule describes how the WS-BPEL structure of the process is obtained from PEOM.

Table 4. QVT rule to obtain the «HumanExecutionClass» metaclass

```
mapping HumanActivity::toHumanExecutionClass () :
        SPExecutionMetamodel::HumanExecutionClass   {
  description := self.description;       status := "Unstarted";
  isInittial  := isInittialActivity ();  name   := self.name;
  responsible_position := self. responsible_position;
  generates += self.Product→forAll (p : Product |
               p.resolveone(WorkProduct);
  responsible := self.isResponsible.resolveone(HumanRole);
  participant += self.isParticipant→forAll (rol : Stakeholder |
                 rol.resolveone(HumanRole);
  PRE  := createPreConditions (); POST := createPosConditions ();
}
```

Table 5. MOFM2T rule to generate WS-BPEL structure of the process

```
[template public defineBPELStructure (process : ProcessExecutionClass)]
  [for (node : ExecutionNodeClass | process.ExecutionNodeClass)]
    [if (node.oclIsTypeOf (HumanExecutionClass)]
       <extensionActivity>
       <b4p:peopleActivity name="[node.name/]"
                           inputVariable ="ToBeSpecified"
                           outputVariable="ToBeSpecified">
       <b4p:localTask reference="[ node.name /]_TSK" />
       </b4p:peopleActivity>
       </extensionActivity>
    [elseif (node.oclIsTypeOf (WSExecutionClass)]
       <invoke name="[ node.name /]" partnerLink="[ node.name /]_PL"
           portType="[node.port/]" operation="[node.nameOperation/]"
           inputVariable="[ node.hasParameter.name /]"
           outputVariable="[node.hasResult.name/]"createInstance="yes"/>
    [elseif (node.oclIsTypeOf (EMailExecutionClass)]
       <invoke name="[ node.name /]" partnerLink="[ node.name /]_PL"
           portType="ToBeSpecified"  operation="ToBeSpecified"
           inputVariable="[ node.hasParameter.name /]"
           outputVariable="[node.hasResult.name/]"createInstance="yes"/>
    [elseif (oclIsTypeOf (ProcessExecutionClass)]
       <!-- … -->
    [else] <!-- ScriptExecutionClass -->
       <empty name="[ node.name /]" />
    [/if]
  [/for]
[/template]
```

5 Discussion, Future Work and Conclusions

In recent years, standards and guidelines (such as PMBOK, PRINCE2, CMMI or ISO 9001, among others) recommend that organizations should formally manage their processes in order to achieve lower costs and improve quality and productivity.

To meet these goals, companies should carry out an effective BPM of their processes to achieve the continuous improvement of such processes.

However, in the context of software organizations, applying BPM is not a simple task due to features of software processes. This situation provokes that software companies usually focus on defining their processes although, later, execution and orchestration are manually and/or unilaterally performed by each involved role. Consequently, software process becomes difficult to execute, manage, maintain, monitor, evolve and measure.

At present, there are many PMLs [18], but just a few of them include mechanisms for supporting the execution of the process. In addition, none of them is mature enough to comply with the commitment pursued. SPEM2.0 standard could be the solution, but its complexity and non-executability makes it impossible. Regarding executability, each proposal presented in Sect. 2 offers mechanisms to perform the process into ad hoc systems, what make harder its application in real environments because companies already use specific process engines. It is interesting to underline that none of these proposals mention mechanisms to support the orchestration of processes.

This paper proposes a MDE-based solution to execute and orchestrate the software process in real environments since it is oriented to be applied, in an integrated way, to enterprise management systems (such as BPM suite [11]). In fact, some papers have been published to report successful cases [12, 14].

Finally, the publication of this paper opens new and interesting future lines of work. On the one hand, we plan to support monitoring and continuous improvement phases of the architecture of PLM_4BS in order to assess the execution of software processes. On the other hand, we aim to research how simulation mechanisms can be included in PLM_4BS so as to support decision-making procedures related to resource allocation or deadlock identification, among other aspects.

Acknowledgments. This research has been supported by the POLOLAS project (TIN2016-76956-C3-2-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish the Ministry of Economy and Competitiveness.

References

1. Van-der-Aalst, W.M.P.: Business process management: a personal view. Bus. Process Manage. J. **10**(2), 5 (2004)
2. ISO/IEC. ISO/IEC 19507:2012 Information technology, Object Constraint Language (OCL). International Organization for Standardization, formal/2012-05-09 (2012)
3. Ruiz-González, F., Canfora, G.: Software process: characteristics, technology and environments. SPT Softw. Process Technol. **5**, 5–10 (2004)
4. Piattini-Velthuis, M., Ruiz-González, F., Canfora, G.: Software process: characteristics, technology and environments. SPT Softw. Process Technol. **5**, 5–10 (2004)
5. Papazoglou, M., Ribbers, P.: E-Business: Organizational and Technical Foundations. Wiley, New York (2006). ISBN-13: 978-0470843765
6. Pedraza, G., Estublier, J.: Distributed orchestration versus choreography: the FOCAS approach. In: Wang, Q., Garousi, V., Madachy, R., Pfahl, D. (eds.) ICSP 2009. LNCS, vol. 5543, pp. 75–86. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01680-6_9

7. García-García, J.A., Alba, M., Escalona, M.J.: Software Process Management: A Model-Based Approach. Information Systems Development: Building Sustainable Information Systems, pp. 167–178. ISBN: 978-1-4614-7539-2 (2013)
8. Schmidt, D.C.: Model-driven engineering. IEEE Comput. **39**(2), 25–31 (2006). IEEE Computer Society
9. OASIS. Web Services Business Process Execution Language. Organization for the Advancement of Structured Information Standards (2007). http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html
10. OMG. XMI BPMN, Business Process Modeling Notation (2011). http://www.omg.org/spec/BPMN/2.0/
11. Meidan, A., García-García, J.A., Escalona, M.J., Ramos, I.: A survey on business processes management suites. Comput. Stand. Interfaces (2017). doi:10.1016/j.csi.2016.06.003
12. Garcia-Garcia, J.A., Enriquez, J.G., Garcia-Borgoñon, L., Arevalo, C., Morillo, E.: A MDE-based framework to improve the process management: the EMPOWER project. In: IEEE 15th International Conference of Industrial Informatics (2017)
13. Salido, A., García-García, J.A., Ponce, J., Gutiérrez, J.: Tests management in CALIPSOneo: a MDE solution. J. Softw. Eng. Appl. (2014). doi:10.4236/jsea.2014.76047
14. García García, J.A., Escalona, M.J., Martínez-García, A., Parra, C., Wojdyński, T.: Clinical process management: a model-driven & tool-based proposal. In: Information Systems Development: Transforming Healthcare through Information Systems. ISBN: 978-962-442-393-8 (2015)
15. OMG. Semantics of a Foundational Subset for Executable UML Models v1.1. Object Management Group (2013). http://www.omg.org/spec/FUML/1.1/
16. García-García, J.A., Escalona, M.J., Domínguez-Mayo, F.J., Salido, A.: Methodological tool solution in the model-driven engineering paradigm. J. Softw. Eng. Appl. (2014). doi:10.4236/jsea.2014.74022
17. Martínez-García, A., García-García, J.A., Escalona, M.J., Parra, C.L.: Working with the HL7 metamodel in a Model Driven Engineering context. J. Biomed. Inf. (2015). doi:10.1016/j.jbi.2015.09.001
18. García-Borgoñón, L., Barcelona, M.A., García-García, J.A., Alba, M., Escalona, M.J.: Software process modeling languages: a systematic literature review. Inf. Softw. Technol. **56**, 103–116 (2014)
19. Bendraou, R., Sadovykh, A., Gervais, M.P., Blanc, X.: Software process modeling and execution: the UML4SPM to WS-BPEL approach. In: 33rd Conference on Software Engineering and Advanced Applications, pp. 314–321. ISBN: 0-7695-2977-1 (2007)
20. Ferreira, A.L., et al.: An approach software process design and implementation using transition rules. In: Software Engineering and Advanced Applications Conference (2011)
21. Wise, A.: Little-JIL 1.5 Language Report. Department of Computer Science, University of Massachusetts, Amherst, MA, UM-CS-2006-51 (2006)
22. Di Nitto, E., Lavazza, L., Schiavoni, M., Tracanella, E., Trombetta, M.: Deriving executable process descriptions from UML. In: Proceedings of the 24th International Conference on Software Engineering ICSE, pp. 155–165 (2002)
23. Cugola, G., Di Nitto, E., Fuggetta, A.: JEDI event based infrastructure and its application to development of OPSS WFMS. Trans. Softw. Eng. **27**(9), 827–850 (2001)
24. Chou, S.C.: A process modeling language consisting high level UML-based diagrams and low level process language. J. Object Technol. **1**(4), 137–163 (2002)
25. OMG. SPEM, Software & Systems Process Engineering Metamodel specification. Object Management Group (2008). http://www.omg.org/spec/SPEM/
26. Bendraou, R., Combemale, B., Cregut, X.: Definition of an executable SPEM 2.0. In: 14th Asia-Pacific Software Engineering Conference, APSEC 2007. IEEE, pp. 390–397 (2007)

27. Havey, M.: Essential Business Process Modelling. ISBN-13: 978-0596008437 (2005)
28. Hill, J.B., et al.: Gartner's Position on Business Process Management. Business Issues. Gartner, Stamford (2006)
29. Escalona, M.J., Gutierrez, J., et al.: Practical experiences in web engineering. In: Advances in Information Systems. Advances in Information Systems Development (2007)
30. Bosch, J.: From Software Product Lines to Software Ecosystems, pp. 111–119 (2009)
31. OMG. Query/View/Transformation (2017). http://www.omg.org/spec/QVT/1.0/
32. OMG.MOF Model to Text Transformation Language (MOFM2T) (2017). http://www.omg.org/spec/MOFM2T/1.0/
33. ISO/IEC. ISO/IEC TR 24744:2007 Software and systems engineering Lifecycle management Guidelines for process description (2007)
34. García-García, J.A.: A proposal for the use of the model-driven paradigm (MDE) for the definition and execution of business processes. Ph.D. thesis (2015). https://documat.unirioja.es/servlet/autor?codigo=3722430

An Axiom Based Metamodel for Software Process Formalisation: An Ontology Approach

Edward Kabaale[2(✉)], Lian Wen[1,2], Zhe Wang[1,2], and Terry Rout[1]

[1] Institute for Integrated and Intelligent Systems, Griffith University, 170 Kessels Road, Brisbane, QLD 4111, Australia
{l.wen,z.wang,t.rout}@griffith.edu.au

[2] School of Information and Communication Technology, Griffith University, 170 Kessels Rd, Brisbane, QLD 4111, Australia
edward.kabaale@griffithuni.edu.au

Abstract. Software development usually follows well known process models and standards for development processes. However, these are usually diverse and described in natural language which complicates their automation, adaptivity and verification. The need for process formalisation has long been highlighted, and we have provided a formalisation and translation algorithm to that effect in earlier work. However, to systematically and faithfully formalise heterogeneous processes from different standards and process models, there is a need to utilise uniform concepts to underpin the formalisation process. Metamodels and ontologies have been explored recently to lay a foundation for structuring and expressing additional rigour to process formalisation. In this study, we develop an axiom based metamodel utilising powertype patterns as a conceptual framework to underpin homogeneous process formalisation. The advantage of an axiomatic and powertype based metamodel approach lies in its potential to determine the metamodel basic constituents and formalism as well as its extensibility and adaptability. We formalise the metamodel using ontologies while adopting use cases from ISO/IEC 29110 and ISO/IEC 24744 standards for metamodel illustrations. Ontology based process descriptions enable process automated verification and adaptivity capability through the use of ontology reasoning support engines.

Keywords: Software process · Metamodel · Powertype · Axiom · Ontology

1 Introduction

Software Engineering (SE) focuses on sound processes and methods for quality software development within budget and time frame. Over the recent decades, the process dimension of SE has received increased attention from both researchers and practitioners [1]. One of the main objectives of this dimension is to enhance the software product quality through formal definition and improvement of the process by which software is developed and maintained. Consequently, it boosts a wide spectrum of approaches to process definition such as

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 226–240, 2017.
DOI: 10.1007/978-3-319-67383-7_17

ISO/IEC 12207 [2], ISO/IEC 29110 [3], process assessment and improvement such as ISO/IEC 33061 [4] and CMMI [5], and process metamodels such as OOSPICE [6], SPEM [7] and SEMDM [8] that specify the conceptual foundations for process modeling. However, these are mainly described in natural language and published in manuals and booklets [9–11], this presents several challenges such as difficulty in finding or updating information in different versions of the same document, lack of automated auditing and verification that limits adherence and monitoring (e.g. constraint checking of required relations among activities, work products and roles). Even though these process modeling techniques enhance software development activities, they still need formal enhancement to enable process automation and verification [10–13].

A formal software process specification is a specification expressed in a language whose vocabulary, syntax and semantics are formally defined and well understood. Its a precise and concise specification that supports formal software process definition and management, automated analysis, verification and validation, understanding, evolution management, classification, improvement and aiding in choosing the appropriate process for a given project [11]. Formal methods such as Petri nets, algebra, ontologies, bayesian networks and composition trees have been used in modeling and formalising software process before. For example composition trees have been used to formally model and compare software processes in [14,15]. Ontology based approaches [10,16–18] have also been proposed to model,validate, constrain and query software process descriptions. Ontologies are being deployed in industry to formalise information models and standards that would otherwise be costly to develop, integrate and share, and monitor through automated queries and verification [16]. Even though, some practitioners especially in small entities use informal process descriptions in industry [19], there is a great need and usage of formal process descriptions in practice as well [11]. However, due to the great diversity and complexity of software processes from different standards and process models coupled with varying situational contexts [20], software process formalisation still lacks a standardized, consistent and faithful way making it error prone, time consuming and thus expensive [21].

Metamodels have been proposed as a way of increasing process modeling rigour and formality for automation [9]. These metamodels lay a foundation in terms of concepts, rules and conceptual relationships among concepts used in process modeling [22]. So processes grounded in metamodels offer a higher degree of formalisation and better support for consistent extensions and modification [9]. Moreover, they impose well formedness rules on process models and process instances instantiated from them. These help in maintaining process consistency and completeness [23]. However, the current process metamodels such as SPEM and SEMDM are too generic and complex to guide software process formalisation [22]. For example, SEMDM was intended to be general methodology metamodel and cover not only process modeling but also other areas like computer supported collaborative work (CSCW)[9]. Indeed such genericity and complexity hinders their comprehension that would be of great help to software process formalisation. Therefore, there is a need for a customised and simple metamodel tailored for homogeneous software process formalisation. To develop this metamodel, we utilise the powertype pattern as introduced in process metamodeling by [24]. The powertype pattern enables tailoring of software process formalisation to specific project requirements and contexts.

Even though, metamodels provide a rigous underpinning and consistent terminology to various aspects of SE, they only deal with conceptual definitions, standardisations and syntax of process models necessitating the need for formal semantics and reasoning of such conceptual definitions [25,26]. We therefore formalise the proposed metamodel using OWL ontologies. Ontologies consititute formal models that define formal semantics and inference services for a shared conceptualisation. These can be used to draw interesting logical conclusions through for example (meta)model checking, model enrichment, dynamic classification, information retrieval and querying of software process models across the metamodel hierarchy thus improving software engineering processes [27].

In this study, we propose an axiom-based metamodel to underpin a homogeneous software processes formalisation. We design this metamodel through rising the abstraction levels of common process elements which define the structure of the process models. The metamodel is grounded in an ontology, reusable, and adaptable. In Sect. 2, we discuss background information on software process meta (modeling) and process instantiation, powertypes and ontologies. In Sect. 3, we abstract the axiom metamodel design from existing process constructs, and illustrate a general overview of the metamodel with processes from ISO/IEC 29110. In Sect. 4, we discuss the formalisation of the metamodel through OWL DL that equips it with formal semantics and reasoning capabilities. Finally, we conclude the paper and identify some future research directions.

2 Background

2.1 Process Metamodeling

Process metamodels are a feasible approach to reduce process modeling complexities through rising the abstraction levels, reuse, and formalisation [22]. According to [28] process metamodels describe a *conceptual framework for expressing and composing software process models*. They describe the relevant software process sub-models, basic concepts, rules and relationships among concepts with notations for expressing process models. They allow capturing informal, behaviour, functional, and strategic views of software processes [22]. Such information can then be used to reason on software process modeling for changes, formalisation, improvements and updates [29]. Software process models are the key result of the process modeling activity and instances of process metamodels. They serve as abstract representations of software processes. These prescribe a software process in terms of the activities to be carried out, the roles and work product types involved. Software processes are then instantiated in a specific project endeavour to develop the desired software product, which in itself is seen as an instantiation of the software process.

Process metamodeling is an important conceptual tool in underpinning the definitions of formal software process models. This has largely been popularised by the OMG standards where the UML is now the de facto standard formalism for software modeling. The modeling layers in UML metamodel are defined based on *strict* metamodeling architecture that only allows *instanceOf* relations between adjacent layers, i.e., variables defined at level M_n can only be realised at M_{n-1}. This is termed as *shallow* instantiation where attributes and constraints are defined at the class level and only realised at the instance level [33]. Where as the OMG standard approach works well for modeling languages defined at OMG level M_2 and used at level M_1, its insufficient for process standardisation that requires *deep* instantiation [24]. That is attributes and constraints defined at metamodel M_2 are realised (enacted) on real world projects at level M_0 spanning multiple modeling levels. When attempt to use UML shallow instantiation for process metamodeling, it results into modeling challenges such as *accidental complexity* [24]. To overcome these challenges, Gonzalez-Perez and Henderson-Sellers [24] introduced the use of powertypes as a way of deep instantiation for process metamodeling.

Powertype Based Metamodeling Framework. The Powertype pattern is a flexible and scalable modeling technique that combines instantiation and generalisation semantics in process metamodeling [9]. Mainly introduced as an alternative solution to the inconsistencies, ambiguities and accidental complexities that result from the use of strict metamodeling technique in process standardization [24]. The powertype pattern has been extensively applied in process modeling [24] and underpins the development of an international standard for metamodeling, i.e., ISO/IEC 24744 Software Engineering - Metamodel for Development Methodologies (SEMDM)[8], therefore, in this study we utilise it to underpin modeling and tailoring of software process formalisation to specific software project contexts through the developed metamodel.

Essentially a powertype is a class whose instances are subclasses of another class called a partitioned class [24]. In this regard a powertype is more like a metaconcept with an extra twist that, its instances can also be subclasses of another class. The powertype and partitioned class are closely related and together with the relation between them form a *powertype pattern*. The powertype class represents groups of instances that are used to classify the partitioned class according to a partitioning discriminator (powertype attribute value, e.g., name). For example, *birds* can be classified according to *birdSpecies* such as *eagle* and *penguin*. The powertype pattern enables the combination of generalisation and instantiation across metamodeling layers through *dual* representation of concepts (also known as *clabject*) [33] where the instance of the powertype and a subtype of a partitioned class are the same thing. From our example *eagle* as an instance of bird species is the same thing as *Eagle*, the subtype of bird. However, metamodels only deal with the syntactical structures but not the formal semantics of process models and therefore, the need to ground the metamodel in an ontology.

2.2 Ontologies

New technologies such as semantic web that provide reasoning support services like consistency checking, information retrieval and querying of software process models are beneficial to improving software engineering processes [34]. OWL[1] is a Semantic Web language designed and standardised by W3C to represent rich and complex knowledge about things, groups of things, and relations between things [35]. Its a knowledge representation language underpinned by Description Logic (DL) that enables expressed knowledge to be reasoned on by human and artificial agents for consistency and inferring implicit knowledge from the explicit one. While there are different OWL flavours such as OWL Full, OWL Lite and OWL DL. Here we are interested in utilising a newer version of OWL DL, i.e., OWL DL 2 [35].

OWL representations commonly referred to as ontologies, can be published and stored in the World Wide Web. An OWL DL ontology is mainly composed of two main components; The Terminological knowledge represented in the TBox (Class Level) and the Assertional knowledge forming the ABox (Instance Level). The TBox defines the intensional knowledge by which a concrete world can be described. This knowledge is represented by axioms in the form of logical sentences. The ABox on the other hand, represents assertional knowledge that complies with the intensional knowledge in the TBox.

In order to use the modeling capabilities of OWL and the potential of DL reasoning in a layered architecture, the OWL based modeling language has to provide essential features to support metamodeling and reasoning across layers; To this effect, the current W3C standard Web Ontology Language (OWL) supports metamodeling mainly in two flavours; OWL Full that implements full metamodeling like in RDF is suitable for formalising the powertype metamodeling approach but is undecidable even for basic inference problems [36]. OWL 2[2] supports a decidable fragment of metamodeling based on *contextual semantics* [36] through the *Punning* technique where the same identifier is used to denote both the ontology class and an individual. It is common place in conceptual and ontology engineering to classify entities either as classes or instances depending on their context [36], given the fact that there is no clear cut borderline between classes and instance classification. In fact some entities are classified as classes in higher abstraction levels and instances in lower abstraction levels making the borderline blurred. Therefore, we use OWL 2 *punning* to model powertypes and clabjects concepts into OWL 2 ontologies for automated verification using various off the self ontology reasoning engines such as FACT++ and HERMIT[3].

3 Axiom Metamodel Design

Process metamodel approaches should identify the most appropriate concepts not only to represent process models but also process assessment [22, 29].

[1] https://www.w3.org/OWL/.
[2] https://www.w3.org/TR/owl2-overview/.
[3] http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/.

The process is the main concept of any software process metamodel [29]. Therefore, every process has some common basic elements such as activity, work products and roles [29,30]. However, different metamodels and standards use them differently in terms of their granularity, formality and abstraction. These form the tangible internal structure of the process and are important in developing a process model of such processes [17,30–32].

To enhance homogeneous software process formalisation, understandability and reduce model maintenance efforts and costs, process modeling should be supported by a very high level of abstraction [17,31]. To this end, we rise the abstraction levels of these common elements and generalise them to design new abstract concepts for the metamodel. Key abstract concepts for the metamodel are stated as axioms. Axioms are statements which are accepted as true [37]. In this regard, their accuracy doesn't need to be proven [37] and can be used as a basis for argument or inference. Axioms have been used widely in providing foundations for theoretical works [38,39] in SE. In here, we use these axioms to provide a theoretical foundation for the essential metamodel concepts for process formalisation. Moreover, through these axioms new metamodel concepts are accepted as true and valid [37]. The main metaconcepts for the metamodel are **Achievable**, **Doable**, **Tangible** and **Assessable**. The relationship between these axioms is visualised in Fig. 1. To demonstrate these concepts we use the software implementation process from ISO/IEC 29110. ISO/IEC 29110: Systems and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs) [3] is an international standard (IS) for VSEs employing not more than 25 people on small software projects (less than six people month). It has two main processes, i.e., software implementation(SI) and project management (PM). These have been mainly drawn from other major standards such as ISO/IEC 12207 and ISO/IEC 15289.

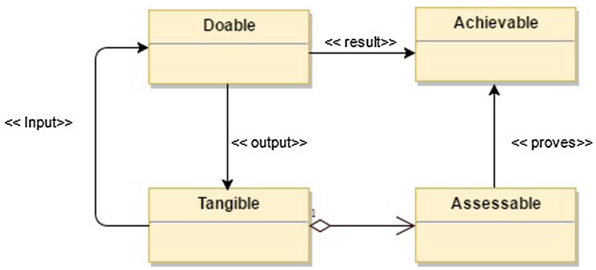

Fig. 1. Relationship between metamodel concepts

Achievable Axiom: *For every process performed, there is something to be achieved.* The purpose of the process can be achieved through the process objectives and the demonstration of the outcomes. We collectively term these as *Achievable*. See Fig. 2 for details. The outcomes are examined for work products and the execution of the base practices to achieve the work products.

These collectively can help to determine the capability of the performed process. The achievable axiom represents what needs to be achieved when performing a selected process. The process purpose is a high level objective of performing a process whose achievement is demonstrated through the outcomes. The ISO/IEC 29110 process objectives are provided by one or more outcomes from ISO/IEC 12207 standard. Therefore the objectives are the specific goals through which the process purpose is accomplished [30]. Because of this relationship, we collectively term this as achievable, after all process objectives and outcomes are meant to ensure successfully accomplishment of a process purpose.

We utilise the xKind metaconcepts from the powertype pattern to tailor the approach to different specific project contexts [9,31] by aggregating process characteristics that match a given project context. For example, in Fig. 2, we have subtypes of achievable as purpose, objectives and outcomes but we also have the same subtypes as instances of the achievablekind where we can assert their characteristics such as capability levels. Through enactment of the purpose subtype on a specific project we are able to specify the individual purpose for such a project for example *developing a website for project A*. Collectively such instances aggregate the characteristics of the achievablekind instances.

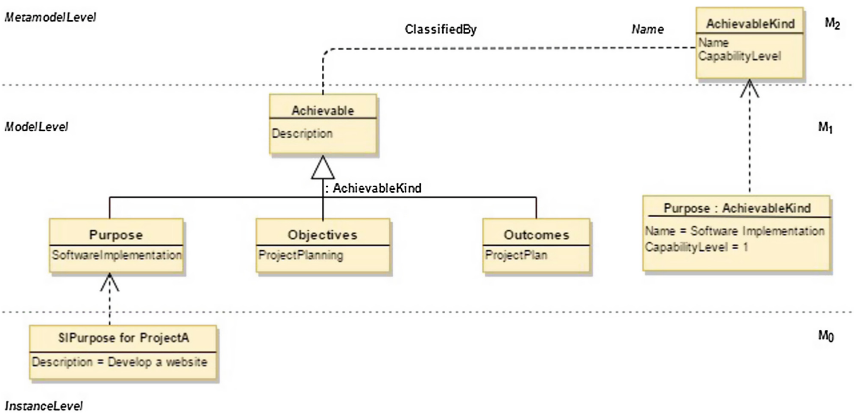

Fig. 2. Achievable Axiom showing process purpose, objectives and outcomes

Doable Axiom: *In order to fulfil the achievables, there is something to be performed.* The basic concepts in performing a process can be generalised into a common abstract concept named *Doable*, see Fig. 3, which represents anything that is performed to fulfill the achievables. This includes processes, activities, tasks and steps. The difference between these is the level of granularity at which they are performed. Whereas the process is a more general concept in the process hierarchy, the task and step on the other hand are more atomic for enactment [30]. The activity hierarchy shows that activities are described at different granularity levels with varying attributes. For example, the activity hierarchy is represented as *domain concept* in OOSPICE, *WorkUnits* in SEMDM and *activity* in [30]. Makinen proposes the use of abstract class *activity* so that uniform attributes can be inherited by all other concrete classes within the activity hierarchy, but then activity itself is also part of the activity hierarchy. Our *doable* class is a generalisation of all things performed to fulfil the process purpose including the process, activity, task and steps. From Fig. 3, we have the main process as *software implementation* with various activities like *requirements analysis* and these are supported by tasks such as *elicit requirements*. The duration of the doable class will be the aggregation of all these activities sharing the same attributes whose values are taken at the project level. On the other hand, the doablekind class provides attributes for the process model level describing the characteristics of all the kinds of processes, activities and tasks.

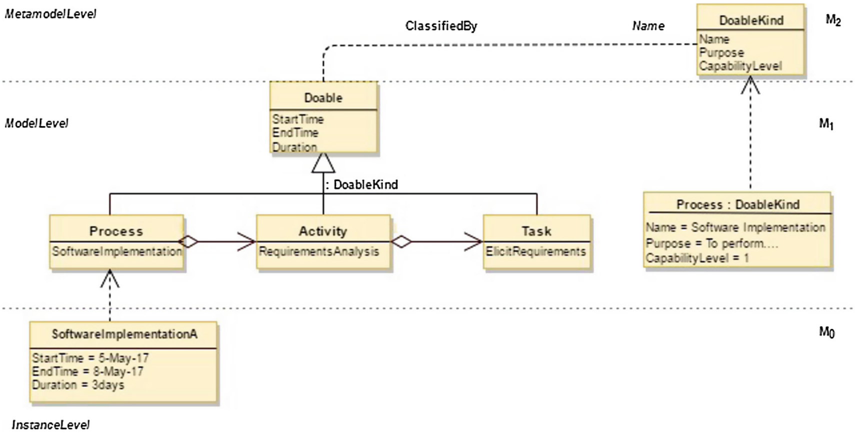

Fig. 3. Doable Axiom showing software implementation process example

Tangible Axiom: *In order to perform the doable, it requires something as input to produce the desired output.* The main purpose of performing a process is to produce outputs. *Tangible* are the inputs and outputs of performing the *doable* see Fig. 4. The tangible within ISO/IEC 29110 at the abstract level are input, internal or output products. When they are outputs, they are always associated with a *destination*. This destination can either be another process for example *project plan* from project management process to software implementation process [3] or just going outside of the process, for example *acceptance document* goes outside of the implementation process to the customer. The tangible can also be inputs to the *doable* for example a *project plan* is an input to the software implementation process. Other times the tangible can be internal work products such as *ChangeRequest* from within the process [3].

Assessable Axiom: *The tangible can be objectively observed and measured to prove if the achievable are fulfilled.* The assessment of the quality of execution and outputs produced is normally done through process assessment.

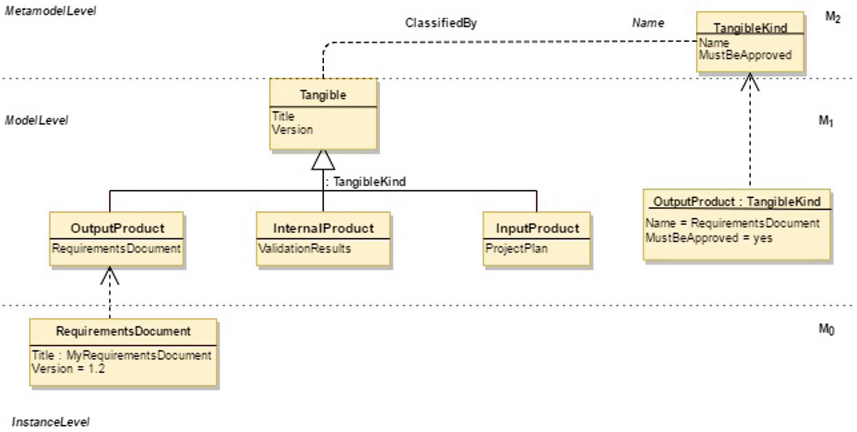

Fig. 4. Tangible Axiom showing work products

Such assessments are normally carried out using prescribed assessment models such as ISO/IEC 33061 [4] and CMMI [5]. These models outline the abstract properties for the process assessment such as process purpose and outcomes [29]. We refer to this collectively as *Assessable*. Process assessment indicators are grouped into two categories as process performance and capability assessment. Its the process performance category that assesses the accomplishment of a process purpose through the process outcomes. Moreover, ISO/IEC 29110 doesn't state any process capability beyond process performance [40]. We therefore limit our assessment to only process performance, i.e., process capability level one. The main assessment indicators for process performance are the *work products* and *base practices* [4] and have been used to develop a process assessment model (PAM) and method for VSEs operating at process capability level one [40]. The work products have been already discussed in the tangible axiom, see Fig. 4. The base practise according to [4] is *an activity that addresses the purpose of a particular process. A coherent set of base practices is associated with each process in the process dimension.* And importantly base practices are described at an abstract level and have been linked to taskkind by [31]. The taskind classes that represent base practices have also been discussed in the doable axiom see Fig. 3.

According to [29] a metamodel that directly supports the concept of capability levels enhances the definitions of process models that are dynamically tailorable along their capability levels. It should be noted, however, that from the *Doable* its only the process that is assessable because its associated with a purpose and outcomes while others are not [29]. An *assessable process metamodel* is one that incorporates the necessary formal properties for assessment so that no external process reference model is needed [29]. A good metamodel should address the different aspects of a process model such as the process hierarchy (*Doable*), work products (*Tangible*), objectives (*Achievable*) as well as the assessment (*Assessable*) of the entire process [29]. Its in this regard that we provide our axiom based metamodel for process formalisation as discussed above that addresses the key different aspects of process models.

4 Ontology Based Process Metamodeling

The need to fix formal semantics for metamodels and their instantiated process models has long been highlighted by different authors [25–27]. Metamodel formal semantics are needed for satisafiability checking at the process model layers as well as checking the consistency of the instantiated process models [27]. The current approaches [27] emphasise fixing formal semantics for metamodels and the corresponding process models helping to maintain consistency between metamodels and their instantiated process models [26]. For example in [27], both the metamodel and process models are transformed to ontologies where the metamodel is transformed to the OWL DL TBox and the process model is transformed to the OWL DL ABox. In [26] domain ontologies are used to define formal semantics for process models while meta-ontologies or foundational ontologies are used to define formal semantics for metamodels. A reference ontology for the domain of software engineering standards has been developed by [31] where a common and unambiguous terminology is sought for all (current and future) software engineering standards developed by ISO/IEC JTC1's SC7. Software engineering standard harmonization ontologies [32] have also been developed to harmonise concepts and term usage across different process models and references models in software engineering domain. All these studies lay a theoretical foundation for the work presented in this paper.

However, the semantics and consistency of the instantiated processes in respect to the metamodel are largely undefined in the current approaches. In earlier work [13], we have shown how the semantics of the instantiated process can be used for process reasoning, verification and conformance to the process model. This provides our motivation for the current section by extending the semantic mapping from the metamodel further to the process instances. This can guarantee the semantic consistency right from the metamodel upto the instantiated process. Hence, we map the metamodel at M_2 to a OWL DL TBox, the clabject (dual entity) at M_1 to OWL DL TBox/ABox (OWL punning) and the instantiated process M_0 to OWL DL ABox. See Fig. 5 for details. With such a mapping, we are able to utilise ontology reasoning support across the metamodeling layers such as satisfiability checking between metamodels and process models, process models and process instances as well as process metamodels and process instances. Moreover, we can enforce and check the well formedness constraints that the metamodel imposes on the process models and process instances respectively.

To illustrate our approach, we take the Tangible metaconcept for an example from our approach as shown by Fig. 5. At level M_2, we have the $TangibleKind$ class(Powertype) that classifies or partitions the $Tangible$ class (partitioned class) at M_1 using *attribute name* as a *partitioning discriminator*. The $Tangible$

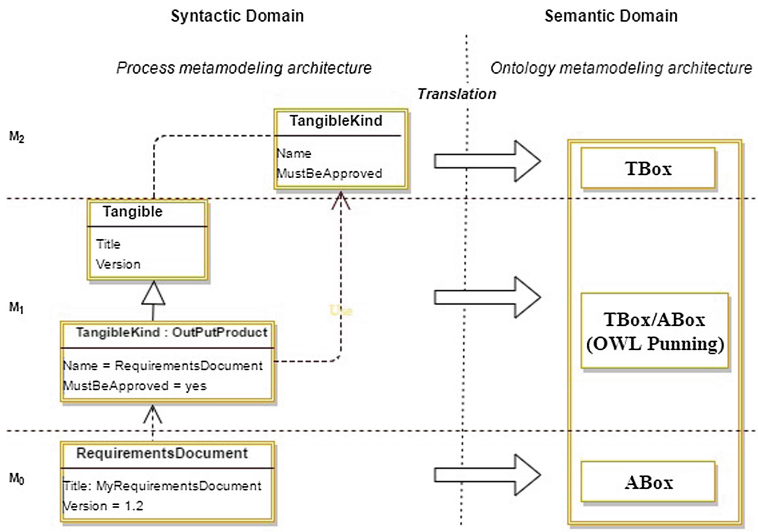

Fig. 5. Ontology based process metamodeling architecture

class at M_1 is subtyped into different kinds such as output, internal and input products (only output products-requirementsdocument is shown in Fig. 5). At the same time the instances of the $TangibleKind$ class are exactly the subtypes of the $Tangible$ class forming a dual concept also known as the *clabject* in our case the *RequirementsDocument* that has both the class facet from the $Tangible$ class and the object facet from the $TangibleKind$. The class facet of the clabject represents the actual output products from enacting the process such as the *RequirementsDocument* which can further be instantiated in a specific project to yield *MyRequirementsDocument* version 1.2. On the other hand, the instance facet of the *RequirementsDocument* is used as a template from which other versions of requirement documents may be later instantiated.

The metamodel is translated to an OWL DL ontology through a translation algorithm developed in our earlier work [13] for consistency checking, verification and query answering. The $TangibleKind$ and $Tangible$ classes at M_2 and M_1 respectively are translated to $OWL\ DL\ TBox$. The class facet of the *clabject* from the $Tangible$ class is translated to the $TBox$ while the object facet of the *clabject* from the $TangibleKind$ class is translated to the $OWL\ DL\ ABox$ at the same level through OWL 2 *Punning* technique. OWL 2 Punning is similar to clabject in process metamodeling because it treats one identifier in this case *RequirementsDocument* as an ontology class and an individual based on the context in which the identifier is used in the ontology. For example, the following axioms state the fact that $RequirementDocument$ is a $Tangiblekind$, and that $MyReuirementDocument$ is a $RequirementDocument$:

$$ClassAssertion(RequirementDocument, MyRequirementDocument) \quad (1)$$

$$ClassAssertion(Tangiblekind, RequirementDocument) \quad (2)$$

We can note from these axioms that the symbol (clabject) *Requirements Document* is used in (1) as a class and in (2) as an individual. This serves our purpose earlier stated, for example requirements document as a subtype of *Tangible* class can be treated as a class where *RequirementsDocument* can be further instantiated into *MyRequirementsDocument*1.2, but on the other hand, it can also be treated as an instance of the *TangibleKind* class and therefore translated to OWL DL TBox where the characteristics of the requirement document *kind* can be asserted such as their approvals. This approach can provide practical support and enhance dynamic software process formalisation, tailoring, assessment and process run time modeling and verification. The formal approach presented in this paper is part of an ongoing work towards software process formalisation and automation where process monitoring, adaptability, and verification can be enabled. Furthermore, ontology reasoning engines can be used to perform automated reasoning and verification on the former model to ensure process (model) well formedness.

5 Conclusion and Future Works

This paper presents an axiom based metamodel towards systematic and faithfully software process formalisation as part of an ongoing work for software process automation and verification. The main aim of the metamodel is to provide uniform formal concepts at an abstract level that may be utilised for software process formalisation. Powertype pattern has been utilised to develop the metamodel with the view of enabling process model extensibility and flexibility with ability to model run time processes and their verification. Powertypes also enable tailoring of processes to different specific projects through the use of *xKind* and *clabject* concepts. This helps to create different hierarchical views/contexts for the modeled process, a limitation earlier identified with UML *strict* metamodeling. Finally we formalise the metamodel using OWL DL into formal ontology. Especially, we formalise the powertype and clabject using the OWL DL 2 punning technique. This enables various types of verification at different levels to be carried. For example, we are able to utilise ontology reasoning tool support across the metamodeling layers such as satisfiability checking between process models and metamodels, process models and process instances as well as process metamodels and process instances.

Software development processes are typically too complex to be modeled and maintained without the help of tools [11]. As this work is ongoing, we are yet to evaluate it in a practical setting. However, the work presented in this paper and the algorithm developed in earlier work [13] forms a conceptual foundation for the development of a software process translation tool that we are currently developing. This tool will enable us to evaluate our approach in a practical setting in future work.

References

1. Fuggetta, A.: Software process: a roadmap. In: ICSE 2000: Proceedings of the International Conference on Software Engineering (ICSE)(2000)
2. ISO/IEC FDIS 12207:2017. Systems and software engineering - Software life cycle processes (2017)
3. ISO/IEC TR 29110-5-1-2:2011, Software engineering Lifecycle profiles for VSEs: Management and engineering guide: Generic profile group: Basic profile (2011)
4. ISO, ISO/IEC 33061:2015 Information technology - Process capability assessment model for software life cycle processes (2015)
5. CMMI Product Team, CMMI for Development, Version 1.3, Software Engineering Institute, Carnegie Mellon University (2010)
6. OOSPICE, Software Process Improvement and Capability Determination for Object-Oriented/Component-Based Software Development (2002)
7. Object Management Group: Software and Systems Process Engineering Meta-Model 2.0, formal/2008-04-01. Object Management Group, USA (2008)
8. ISO/IEC, 2007. ISO/IEC 24744. Software Engineering Metamodel for Development Methodologies. ISO, Geneva (2007)
9. Henderson-Sellers, B., Gonzalez-Perez, C.: A comparison of four process metamodels and the creation of a new generic standard Inform. Softw. Technol. (2005)
10. Gallina, B., Szatmári, Z.: Ontology-based identification of commonalities and variabilities among safety processes. In: Abrahamsson, P., Corral, L., Oivo, M., Russo, B. (eds.) PROFES 2015. LNCS, vol. 9459, pp. 182–189. Springer, Cham (2015). doi:10.1007/978-3-319-26844-6_13
11. Diebold, P., Scherr, S.: Software process models vs. descriptions: What do practitioners use and need? J. Softw. Maint. Evol. Res. Pract. (2017)
12. Tarhan, A., Giray, G.: On the use of ontologies in software process assessment: a systematic literature review. In: EASE (2017)
13. Kabaale, E., Wen, L., Wang, Z., Rout, T.: Representing software process in description logics: an ontology approach for software process reasoning and verification. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 362–376. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_26
14. Wen, L., Tuffley, D., Rout, T.: Using composition trees to model and compare software process. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 1–15. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21233-8_1
15. Wen, L., Rout, T.: Using composition trees to validate an entry profile of software engineering lifecycle profiles for very small entities (VSEs). In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 38–50. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_4
16. Kharlamov, E., Grau, B.C., Jiménez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., Horrocks, I.: Capturing industrial information models with ontologies and constraints. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 325–343. Springer, Cham (2016). doi:10.1007/978-3-319-46547-0_30
17. Liao, L., Qu, Y., Leung, H.K.N.: A Software Process Ontology and its Application (2005)

18. Clarke, P.M., Calafat, A.L.M., Ekert, D., Ekstrom, J.J., Gornostaja, T., Jovanovic, M., Johansen, J., Mas, A., Messnarz, R., Villar, B.N., O'Connor, A., O'Connor, R.V., Reiner, M., Sauberer, G., Schmitz, K.-D., Yilmaz, M.: Refactoring software development process terminology through the use of ontology. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2016. CCIS, vol. 633, pp. 47–57. Springer, Cham (2016). doi:10.1007/978-3-319-44817-6_4
19. Kabaale, E., Nabukenya, J.: A systematic approach to requirements engineering process improvement in small and medium enterprises: an exploratory study. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 262–275. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21843-9_21
20. Jeners, S., P. Clarke, P., OConnor, R.V., Buglione, L., Lepmets, M.: Harmonizing software development processes with software development settingsA systematic approach. Commun. Comput. (2013)
21. Garca, F., Serrano, M., Cruz-Lemus, J., Ruiz, F., Piattini, M.: Managing software process measurement: a metamodel-based approach. Inf. Sci. (2007)
22. Martins, P.V., da Silva, A.R.: PIT-ProcessM: a software process improvement meta-model. In: QUATIC, 7th International Conference (2010)
23. Pereira, E., Bastos R., da C. Mra, M., Oliveira T.: Improving the consistency of SPEM based software processes. In: Proceedings of the 13th ICEIS (2011)
24. Gonzalez-Perez, C., Henderson-Sellers, B.: A powertype-based meta- modelling framework. Softw. Syst. Model. (2006)
25. Henderson-Sellers, B.: Bridging metamodels and ontologies in software engineering. J. Syst. Softw. **84** (2011)
26. Saeki, M., Kaiya, H.: On relationships among models, meta models and ontologies. In: Proceedings of Workshop on Domain-Specific Modeling (2007)
27. Staab, S., Walter, T., Gröner, G., Parreiras, F.S.: Model driven engineering with ontology technologies. In: Aßmann, U., Bartho, A., Wende, C. (eds.) Reasoning Web 2010. LNCS, vol. 6325, pp. 62–98. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15543-7_3
28. Lonchamp, J.: A structured conceptual and terminological framework for software process engineering. In: The ICSP 2. IEEE Computer (1993)
29. Gonzalez-Perez, C., McBride, T.M., Henderson-Sellers, B.: A metamodel for assessable software development methodologies. Softw. Qual. J. (2005)
30. Makinen, T., Varkoi, T.: Analyzing a Process Profile for Very Small Software Enterprises. In: SPICE (2008)
31. Gonzalez-Perez, C., Henderson-Sellers, B., McBride, T., Low, G.C., Larrucea, X.: An Ontology for ISO software engineering standards: 2) Proof of concept and application. Comput. Stand. Interfaces (2016)
32. Pardo-Calvache, C.J., Garca-Rubio, F.O., et al.: A reference ontology for harmonizing process-reference models (2014)
33. Atkinson, C., Kühne, T.: The essence of multilevel metamodeling. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 19–33. Springer, Heidelberg (2001). doi:10.1007/3-540-45441-1_3
34. Jekjantuk, N., Groner, G., Pan, J.Z.: Modeling and reasoning in metamodeling enabled ontologies. Int. J. Softw. Inf. (2010)
35. OWL 2 Web Ontology Language Primer, 2nd edn., https://www.w3.org/TR/2012/REC-owl2-primer-20121211
36. Motik, B.: On the properties of metamodeling in OWL. J. Logic Comput. **17**(4), 617637 (2007)

37. Suh, N.P.: Axiomatic Design: Advances and Applications. Oxford University Press, New York (2001)
38. Kim, S.J., Suh, N.P., Kim, S.: Design of software systems based on AD. Ann. CIRP **40**(1), 16570 (1991)
39. Arsenyan, J., Bykzkan, G.: Modelling collaborative software development using axiomatic design principles. IAENG (2009)
40. Varkoi, T.: Process assessment in very small entities-An ISO/IEC 29110 based method. In: 7th International Conference QUATIC. IEEE (2010)

Towards a Semi-automated Tool for Interoperability Assessment: An Ontology-Based Approach

Gabriel S.S. Leal[1,2,3(✉)], Wided Guédria[3], Hervé Panetto[1,2], and Erik Proper[3]

[1] CNRS, CRAN UMR 7039, Vandœuvre-lès-Nancy, France
{gabriel.da-silva-serapiao-leal, herve.panetto}@univ-lorraine.fr
[2] Université de Lorraine, CRAN UMR 7039, Boulevard des Aiguillettes, B.P. 70239, 54506 Vandœuvre-lès-Nancy, France
[3] ITIS, TSS, Luxembourg Institute of Science and Technology (LIST), 5, Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
{gabriel.leal, wided.guedria, erik.proper}@list.lu

Abstract. Interoperability is an essential requirement to be verified when enterprises are starting and maintaining a collaborative relationship. To ensure that such a requirement is continuously met, interoperability needs to be assessed. Various assessment approaches have been proposed in the literature to identify strengths and weakness of an enterprise in terms of their ability to interoperate. However, the main existing approaches are addressing specific aspects of interoperability and focusing on one type of measurement. To assess different aspects of interoperability of the same company, one may use multiple approaches which might cause redundancy and confusion considering the different metrics. Therefore, the objective of this paper, is to propose an assessment approach based on the so called Ontology of Enterprise Interoperability. The proposed approach is supported by a semi-automated tool aiming at reducing the time and paperwork required for evaluation. An example of a networked enterprise is used to validate the approach.

Keywords: Interoperability assessment · Interoperability requirements · Ontology · Networked enterprise

1 Introduction

The development of interoperability among members of a network is a major issue, considering the overall collaboration and cooperation, faced by the Networked Enterprises (NE) [1]. Regarding the term “Interoperability”, the most accepted definition is provided by IEEE where it is seen as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [2]. Later, this definition was extended by Vernadat [3] for the enterprise domain and considered as: “Enterprise Interoperability (EI) provides two, or more, business entities the ability to exchange or sharing information and of using the functionality of one another in a distributed and

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 241–254, 2017.
DOI: 10.1007/978-3-319-67383-7_18

heterogeneous environment" [3]. Hence, in this context, the interoperability *per se* happens when two or more enterprise systems (humans, machines, software, etc.) belonging to the members of the network, successfully interoperate with each other. Therefore, the ability to interoperate is a crucial requirement that needs to be verified when two or more enterprises need to collaborate. As soon as this requirement is not achieved, it becomes a problem that requires being solved [4]. Thus, to avoid problems and better support enterprises to collaborate with their partners, the interoperability between their systems needs to be assessed and continuously improved. Indeed, assessing the enterprises' systems ability to interoperate is frequently the initial step toward a new collaboration development (e.g. the creation of a new network, the arrival of a new member, etc.) or an improvement program (e.g. reducing the negative impacts caused by interoperability problems or transformations). For determining their systems' strengths and weakness regarding interoperability, enterprises should benefit from the use of Enterprise Interoperability Assessment (EIA) approaches. It involves identifying the needs, or gaps, between where companies envision themselves in the future and the companies' current states.

So far, comparative studies have been conducted to analyse interoperability assessment approaches [1, 5–9]. Based on the analysis' results, we identified two relevant issues: First, the existing approaches are focusing on a particular kind of measurement [10] and assessing a specific aspects of interoperability (i.e. Organisational, Conceptual or Technical) [10, 11]. Second, the majority of the studied approaches are only performed manually, which is difficult (i.e. tedious and time-consuming) and very expensive. Considering the first issue, we argue that the application of multiple approaches for covering all interoperability aspects might cause redundancy and confusion when measuring the same aspect using different metrics and viewpoints. Hence, we adopt the Maturity Model for Enterprise Interoperability (MMEI) [1] as it is the only one covering all aspects of interoperability. Further, having a common foundation for sharing contextual knowledge across multiple stakeholders (i.e. assessors, sponsors and participants) is a necessity to perform the assessment [12]. The use of an ontology [13] for formally specifying the various relevant concepts from EIA domain is paramount. Thus, we adopt the Ontology of Enterprise Interoperability for representing (OoEI) the EI domain. Taking into account the second issue, the implementation of a semi-automated tool is needed for improving the assessment process efficiency by reducing the time and paperwork required for evaluation, and by ensuring more accurate results [12, 14]. The decisional core defined in the OoEI will support the development of such tool.

Therefore, the objective of this paper is to propose an ontology-based approach for interoperability assessment. It is built on the concepts from the OoEI and supported by a semi-automated tool for Interoperability assessment. The originality of the proposed approach is twofold: (i) the guidance provided during the EIA process from the scope definition to the practices recommendation, passing through the EI Problem/Solution identification; and (ii) the use of tool for supporting the information collection and the ontology reasoning for identifying problems and associated recommendations. It is worth noting that the tool is still under development.

The remainder of this paper is as follow – Sect. 2 gives an overview of the relevant related work used for developing the proposed approach. It is followed by Sect. 3 where the semi-automated tool architecture is depicted. The approach general steps and how it is supported by the proposed tool is also illustrated in this section.

Section 4 describes an illustrative example based on an active networked enterprise in Luxembourg. Section 5 discuss the current version of the proposed approach and tool. The conclusion and future works are brought forward in Sect. 5.

2 Literature Review and Background

This section presents the related work used for proposing the ontology-based assessment approach. As a sound understanding about interoperability is paramount for supporting the EIA, different frameworks and models that have been found in the literature about the Interoperability, Enterprise and Networked Enterprise domains are brought upward. This will allow the identification of the central concepts of these domains and their relations. Existing EIA approaches and the interoperability requirements that should be satisfied to reach the objectives of the network are depicted and discussed. Further, some existing ontology-based assessment approaches are highlighted for illustrating the usefulness and relevance of such approach.

2.1 Frameworks, Ontologies, and Models for Enterprise Interoperability

In the past years, researchers and practitioners have proposed various frameworks, ontologies, and models to describe the interoperability and enterprise interoperability domains. For instance, the Classification Framework for Interoperability [5] proposing a classification among the different types of interoperability associated with systems' models, the European Interoperability Framework (EIF) [11] describing the different interoperability levels and focusing on the interoperability between public entities from various government around the Europe, the Framework for Enterprise Interoperability (FEI) [10] heightening the aspects of interoperability and the barriers associated, the reference model for sustainable interoperability in networked enterprises [15] providing formal methods categorised in interoperability practices layers, and the Ontology of Enterprise Interoperability (OoEI) [4] which formally describes the interoperability domain while providing support aid for interoperability problem diagnosis. 1Among the cited works, we adopt in this paper the OoEI as it is defined based in the main EI frameworks and it considers interoperability from a problem-solving perspective, not restricted to communication matters [4]. The ontology is described below briefly.

The Ontology of Enterprise Interoperability. The ontology includes a systemic core centred on the notion of the system and its properties, an interoperability core considering three EI dimensions (derived from [10]) represented by the concepts *Interoperability Barrier* (i.e. conceptual, organizational, and technical), *Interoperability Concerns* (i.e. business, process, service, and data) and *Interoperability Approaches* (integrated, unified and federated) and a decisional core that constitutes the basis to build a decision-support system for EI. OoEI implements the *Interoperability* concept as a subclass of the *Problem* concept. Problems of interoperability exist when there is a *Relation*, of any kind, between incompatible *Systems* in a super-system they belong to or system they will form. The *Incompatibility* concept is a subclass of a more generic *ExistenceCondition* class aiming at explicitly formalising the fact that Incompatibility is the source of interoperability problems. Figure 1 illustrates the main concepts of the OoEI.

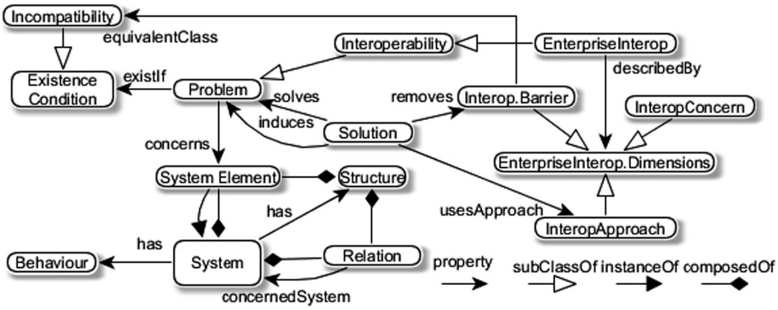

Fig. 1. An overview of the OoEI

2.2 Enterprise Interoperability Assessment

Numerous interoperability assessment approaches have been proposed in the literature, which can be classified by the different interoperability aspects (Technological, Organisational or Conceptual) and the kind of measures they are using. Based on a literature review and surveys [1, 5–9] there are four kinds of assessment: the *Potentiality* assesses the interoperability between a system towards its environment, before any interoperation [10]. The *Compatibility* assesses the interoperability between two known systems [10], the *Transformation* assesses the potential impacts that any change can cause in the overall system, before/during/after any interoperation and finally the *Performance* which assesses the cost, delay and quality of the interoperations [10] during the collaboration.

The main existing EIA approaches are: the Levels of Interoperability in Information Systems (LISI) Reference Model [16] providing measures for assessing the technical maturity of systems. The Levels of Conceptual Interoperability Model (LCIM) [17] and Organisational Interoperability Maturity Model (OIMM) [18] extending LISI for conceptual and organisational maturity assessment, respectively. The Maturity Model for Enterprise Interoperability (MMEI) [1] proposing potential measures for assessing all interoperability aspects and concerns. The Interoperability score [19] measuring the interoperability of complex networks using the operational thread as its foundation and providing a single number measure of how well the systems interoperate along the thread. The formal measures for semantic interoperability assessment [20] providing two measures for comparing two specific systems: the maximal potential and the minimum effective semantic interoperability. Despite not addressing interoperability directly, the Capability Maturity Model Integration (CMMI) framework (including the CMMI models for development (CMMI-Dev) [20], for services (CMMI-SVC) [21], for acquisition (CMMI-ACQ) [22] and the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [23]) can also be useful for appraising the process maturity of the organisation and thus, guiding the development or improvement of processes that meet the business goals of an organisation. Similarly, the ISO/IEC 33000 series can also support the EIA, especially the ISO/IEC 33020 [24] which proposes a measurement framework for the assessment of process capability and organisational maturity.

Among the studied approaches, we adopt the MMEI as it defines a common framework for assessing and measuring interoperability maturity and the gap between the current state and the desired state; and provide information about best practices that allow enterprises to improve their interoperability readiness [1].

The Maturity Model for Enterprise Interoperability. The model defines twelve areas of interoperability which are the result of the crossing between the Interoperability Barriers and Concerns. These areas of interoperability contain the EI criteria that each concern should satisfy to avoid interoperability barriers and to achieve a given maturity level. The EI criteria are major assets to support the management of the EIA as they can be used as indicators to identify interoperability problems. Table 1 presents the areas of interoperability and the maturity level 2 EI criteria regarding the potentiality kind of measurement.

Table 1. The areas of interoperability and their objects of evaluation.

	Conceptual	Technical	Organisational
Business	Use of standards for alignment with other models	Standard and configurable IT infrastructures are used	Human resources trained for interoperability
Process	Use of standards for alignment with other models	Standard Process tools & platforms	Procedures for processes interoperability are in place
Service	Use of standards for alignment with other models	Standards and configurable service architecture and interface	Procedures for services interoperability are in place
Data	Use of standards for alignment with other models	Automated access to data based on standard protocols	Rules and methods for data management are in place

Further, the MMEI proposes an assessment methodology that is composed of five phases: the Assessment preparation, the Information Collection, the Information Validation, the Interoperability Assessment and the Maturity Level determination. More details about each phase can be found in [1].

2.3 Ontology-Based Assessment Approaches

Ontologies provide a basis for the shared understanding of some area of interest among a community of people who may not know each other at all, and who may have very different cultural backgrounds [25]. Ontologies have been used successfully in the past to design interactions between entities efficiently within supply chain networks [26], human actors from a given organisation [25], concepts from the interoperability domain [4], etc.

A literature review focusing on the existing ontologies in the EI domain did not uncover any ontological approach for supporting interoperability (semi) automated assessment. However, ontology-based approaches developed for other fields may be reusable for investigating the advantages and disadvantages of such approach. For instance, the Ontology-based framework proposed by [27] which tackles the problem of automatic risk assessment in unpredictable road traffic environments, the ontology-based approach to support the risk assessment for the intelligent configuration of supply networks [28] and the ontology-based approach for assessing records management systems [12].

Besides the use of ontology-based systems, we encounter in the literature different techniques for automating the assessment process. For example, the automated risk assessment for improving Information technology change management [29] and the Software-Mediated Process Assessment (SMPA) [30]. However, these approaches present some disadvantages. For example, in most of the time, the measurement frameworks (including the evaluation criteria, rating scale, practices for improvements) are rigid i.e. the systems are built to support a particular model (e.g. a standard). Another inconvenience is that the systems do not allow to modify or correct the automatically generated report.

Thus, the advantages of using an ontological approach for performing a semi-automated assessment are: (i) it establishes a common foundation for sharing contextual knowledge across and participating agents [12]; (ii) It facilitates common domain understanding [12]; (iii) it reduces time and paperwork required for evaluation, and ensures more accurate results [12], and (iv) it allows the instantiation of different assessment models (iv) it give freedom to assessors to provide insights, to modify and correct the assessment results based on their experience.

3 The Semi-automated Tool for Interoperability Assessment

In this section, the main elements that compose the semi-automated tool system are presented. The use of the term "semi-automated" is deliberate. We argue that the insights and expertise of concerned persons (e.g. assessors) are valuable to the assessment. For example, lead assessors use their expertise to validate or not the proposed results of the tool. The purpose of our approach, therefore, is not to replace manual ratings performed by assessors but it shall rather contribute to the decision support by providing relevant information on the current state of the assessment process as it is executed.

This section is structured as following: First, we describe the ontological core of the system. It is followed by the description of the layers composing the system architecture. Further, the user's profiles and the system's functionalities are presented. Finally, we present how the proposed system supports the different assessment steps.

3.1 The Ontological Core

The ontology used as basis for developing the semi-automated tool is the Ontology for Enterprise Interoperability. We describe here, additional classes and their relations for representing the relevant interoperability assessment concepts. Figure 2 illustrates the ontology.

An *Assessment* concerns to an *ObjectOfAssessment* that can be something or someone. In our particular case, the Object of Assessment are the *Relations* between *Systems* (which can be an enterprise system, an enterprise or a networked enterprise). An *Assessment* has a *Type* which can be: Potentiality, Compatibility, Transformation or

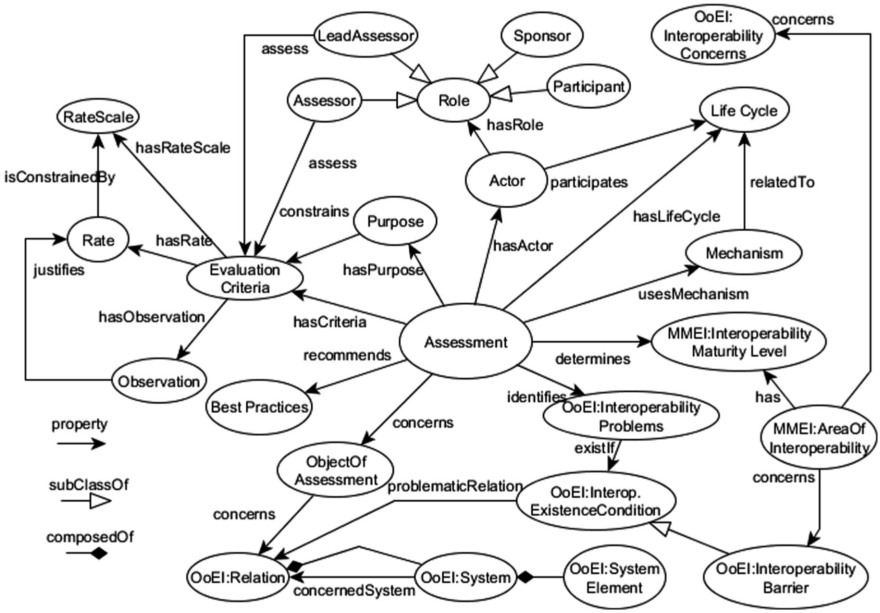

Fig. 2. The ontological core of the proposed system

Performance Assessment. The *Type* concept constrains the *ObjectOfAssessment*. For instance, the Compatibility Assessment aims at assessing the interoperability of a two specific system. Thus, the *ObjectOfAssessment* in this kind of assessment is the relations of concerned systems (e.g. the relation between two employees, the semantic relationships between two enterprise models, etc.). The *Assessment* has a *LifeCycle* which are the phases of the assessment. In our case, the instances of this concept are the assessment steps defined in the MMEI [1].

An *Assessment* has an *Actor*. An *actor* has a *Role* which can be: (i) *Assessors*, who evaluate the *ObjectOfAssessment*; (ii) the *LeadAssessor* also assess the *ObjectOfAssessment* but also is responsible for aggregating the assessment from multiple assessors if it is the case; (iii) the *Sponsor* is who ask for the assessment, and define the assessment scope; and finally the (iv) *Participant*, who provides relevant information through interviews, workshops, etc. For the *Assessment* takes place, at least one *Assessor* should be attributed. The *Assessor* uses *Mechanisms* related to each *LifeCycle* of an *Assessment*. For example, SCAMPI as well as MMEI use, among others, interviews and document analysis for data gathering during the "Information collection" phase. One of the most important concepts in this ontology is the *EvaluationCriteria*. It has a *Rate* (linguistic variable characterising the achievement of the concerned criterion) which is constrained by a *RateScale*. In our specific case, we adopted the rate scale defined in MMEI (i.e. an evaluation criteria can be Fully Achieved, Largely Achieved, Partially Achieved or Not Achieved). An *EvaluationCriteria* also has an *Observation*, allowing assessors to enter a commentary about their given rate. The *EvaluationCriteria* in its turn is related to the *AreasOfInteroperability*. Each *AreaOfInteroperability* has a

MaturityLevel that is determined by the assessment. As outputs, an assessment determines the *MaturityLevel"* of enterprises, identifies a set of "*InteroperabilityProblem*" and recommends a set of "*BestPractices*".

For modelling the ontology, we adopted the Ontology Web Language (OWL) [31] as it is an open standard for semantic knowledge representation. The tool used for modelling and building it was the Protégé 5.0 [32]. The logic rules for determine the maturity levels are written in Semantic Web Rule Language (SWRL). The rules are based on the EI criteria and fuzzy rules defined in MMEI [1].

3.2 The System Architecture

In order to accommodate the different components, the system architecture distinguishes three layers as illustrated in Fig. 3.

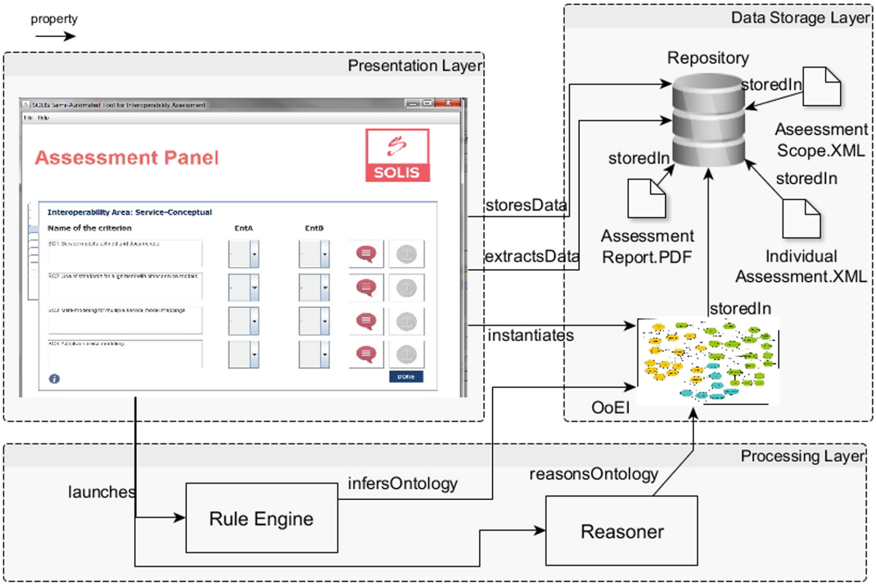

Fig. 3. The architecture of the enterprise interoperability assessment system based on OoEI

The *Presentation layer* includes all the user interfaces which support the user to operate the system. For example, it supports the data collection (e.g. the rate attributed to evaluation criteria) and results' presentation (e.g. the identified potential interoperability problems). The *Data Storage layer* consist in a repository for storing generated files e.g. the assessment report, aggregation report, the instantiated ontology etc. The *Processing layer* containing (i) a logic rules engine for inferring the logic rules into the ontology and (ii) a reasoner for checking the ontology consistency and for reasoning the ontology in order to identify potential problems and recommendations.

3.3 The Users' Profiles

There are three user' profiles: The assessor, the lead assessor and the administrator.

The *Assessors* are expected to assess the concerned areas of interoperability of a given assessment. After collecting data (e.g. through interviews) they are responsible for entering the rating for each evaluation criteria into the system. They are also responsible for entering a comment justifying their evaluation. For example, an assessor gives a rate "*Partially Achieved*" to the Evaluation Criteria "*Services are dynamically composable and applications are networked*"; for explaining this rate, the assessor justifies with an observation, which is "*among the four assessed enterprises, only two of them have connected applications*". In certain cases, they also may attach files for proving their rates and comments (e.g. if an assessor state that a process model is documented, it is expected that the file containing the model is uploaded into the system). The *Lead Assessor* is expected to have a clear understanding of the assessment workflow and operates the semi-automated tool in order to facilitate the entire assessment. He is responsible for aggregating and readjusting (if needed) the rating for each assessed area of interoperability. The observations and upload files are helpful for the lead assessor when aggregating and validating the final rating. He is also responsible for generating the assessment results report. Such a report, contains the current state of the assessed network, the criteria ratings, and the recommended best practices that the network needs to follow. A lead assessor can be also an assessor. Finally, the *Administrator* is responsible for maintaining and (re)configuring the system. He does not participate to the assessment.

3.4 The Tool's Functionalities

The tool's functionalities are available accordingly to the user profile. For example, Assessor can open or edit their assessment but they cannot generate recommendation reports. The main functions for the Lead assessor are listed below:

– *Create the Assessment Scope*: Enters the name of the systems (e.g. enterprises) and the areas of interoperability to be assessed. Select the type of measurement. An XML file containing these information is created.
– *Assign assessor*: Assigns at least one assessor to a given Assessment Scope. The XML file from the concerned Assessment Scope is then updated with the assigned assessors.
– *Edit Assessment Scope*: Edits any information from an Assessment Scope. The XML file from the concerned Assessment Scope is then updated.
– *Delete Assessment Scope*: Selects and deletes a given Assessment Scope.
– *Aggregate Assessments*: Selects and aggregates the assessment files generated by the assessors, from the concerned Assessment Scope. The XML file is then updated with the aggregation result.
– *Generate Report*: Generates a report containing the identified potential problems and the related best practices.

The main functions for the Assessor are:

– *Open an Assessment*: Opens as assessment that was assigned to the assessor. It allows the assessor to enter the criteria rating.
– *Save an Assessment*: Create a XML file containing the individual criteria rates from a given assessor.
– *Edit an Assessment*: Edits any information (criteria rate) from an Assessment. The XML file is then updated with the changes.
– *Conclude an Assessment*: Sends a notification to the Lead assessor. The concerned Assessment cannot be edited anymore.

3.5 The Assessment Steps

The approach steps presented here are based on the MMEI assessment methodology described in [1]. Figure 4 illustrates the steps which are supported by the proposed semi-automated tool.

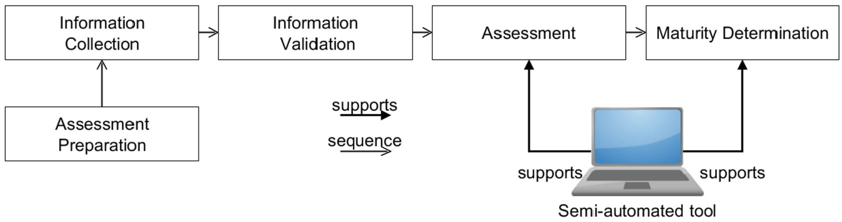

Fig. 4. The proposed EIA approach steps

Interoperability assessment. Having defined the assessment scope, the lead assessor selects, in the tool, the concerned areas of interoperability to be assessed, the kind of measurement. He also assign the concerned assessors. Further, having collected and analysed the information, the assessors enter the criteria rating using the semi-automated tool interface. The tool generates an XML file containing the evaluation ratings and stores it in a dedicated repository. Further, the lead assessor aggregates the assessments provided by the multiple assessors by using tool. It extracts the information from each XML file provided by the assessors and aggregates the criteria rating values according to the algorithm proposed in MMEI [1]. The final results are storage in another XML file. Finally, the proposed tool instantiates the ontology (i.e. opens the OWL file containing the ontology and instantiates it) with the data from the later XML file created. The result is an OWL file containing the current state of the concerned enterprise(s). **Maturity Determination**. Having the ontology duly instantiated, the lead assessor launches the rule engine through the proposed tool. The considered rules are those concerning the selected type of measurement. Moreover, the reasoner infers the ontology in order to identify EI problems. The inferred and reasoned facts (i.e. the outcome of the rules execution) are stored in an XML file. The identified rules that are not fulfilled represent the potential interoperability problems. Based on the inferred ontology and the identified non fulfilled criteria, the tool proposes the related best practices and point outs the potential influences of the non-fulfilment of criteria.

4 Illustrative Example

This section illustrates an example based on an active network of enterprises in Luxembourg. First, the network business scenarios is briefly presented. Further, the potentiality assessment of an enterprise from the network based on the MMEI criteria is presented. The information used to define the scenario were gathered through interviews and analysis of provided documents by the network. The name of the network and its members remains classified for security reasons. Thus, we will refer to them as “TheNetwork” and “EntA”.

Scenario. TheNetwork is a platform and accelerator for Luxembourgish and international start-ups that want to scale up their business by commercialising their product or service from Luxembourg to international markets. It was created with the objective to offer a unique one-stop shop for entrepreneurs that want to conquer physical and or digital markets globally. It offers a quick access to an established unique international partner network of investors, financial institutions, marketing experts, innovation managers, etc. In this scenario, four main types of participants can be figured out: the Mediator (i.e. TheNetwork and its members), the Customer (Start Ups), the Service Providers (i.e. EnTA, and other partners) and Investors. Their relations are illustrated on Fig. 5.

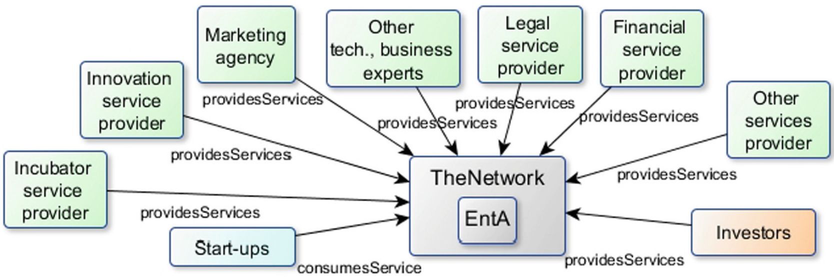

Fig. 5. General view of TheNetwork.

The potentiality assessment. As the tool is still in development, the ontology instantiation and inference were made using the protégé interface. The assessment presented here was made by one assessor and the selected interviewees (i.e. participants) are members of the board of directors of each enterprise.

The Maturity level achieved by EntA is equal to 0 which means that companies do not have an appropriate environment for developing and maintaining interoperability. It is important to note that a lower interoperability maturity for a company does not systematically mean a dysfunction at all levels and for all functions of the enterprise. The maturity is only evaluated from the interoperability point of view and cannot be applied for another purpose. Considering the maturity level determination and the Business-Conceptual area of interoperability as an example, a recommended best practice is: Document the business model with the intent to facilitate the business model sharing as the information will be not only in the minds of the employee who defined it. It will also avoid business semantics problems i.e. the meaning of terms used to express business issues will be explicitly documented.

Applying the proposed tool has allowed us to identify more rapidly the potential interoperability problems and the related best practices.

5 Discussion

The proposed semi-automated tool is still in development. The current version of the system is able to store the information entered by the assessors and to aggregate the criteria rating. The code for the ontology instantiation is not yet completed. For checking the consistency of the ontology and to generate the assessment results, we utilise the protégé interfaces i.e. the instantiation step is done manually. Thus, the following functionalities are still to be coded or improved: (i) Send Notification that will be part of the Assign Assessor (for sending an email when a new assessment is assigned to the assessor) and Conclude Assessment (for sending an email to the lead assessor when an assessment is completed by an assessor) functions; (ii) the Automated ontology instantiation; (iii) the Automated retrieval of information from the inferred ontology. It will be done by the implementation of SPARQL queries. And finally, (iv) the recommendation of best practices and their potential influences in the overall system. The current version provides recommendation but it not considers the different potential impacts if adopted. Further in Sect. 4 a potential application of the proposed approach is illustrated. As future work, we intend to use other case studies, for evaluating and validating the proposed approach. Indeed, the case study method allows researchers to retain the holistic and meaningful characteristics of real-life events (e.g. organisational and managerial processes, etc.) [33].

6 Conclusion

In this paper, we have proposed an Ontology-based Interoperability Assessment for Networked Enterprise. The proposed approach is supported by a semi-automated tool that allows to determine the maturity level of an organization in terms of interoperability and point out some improvement actions that can be undertaken. We argue that adopting an ontological approach, allow us to provide a common understanding for interoperability assessment, despite the different views that exist. This is done through an investigation on the different concepts of Interoperability Assessment and Enterprise interoperability to identify the core concepts related to these domains. The general steps of the proposed approach as well as the different stakeholders are also described. The current version of the tool is presented. It has the objective to facilitate the assessment approach by providing a semi-automated calculation of the maturity level for each area of interoperability. An illustrative example of an active NE in Luxembourg has been investigated to validate the proposed ontology, by depicting the potentiality assessment of a particular enterprise.

As future work, we intend to improve the current state of the tool and develop it further with extra functionalities. We also intend to improve the enterprise information collection by performing two steps: (1) implement online forms where employees can connect and enter relevant information. (2) Develop and implement an application programming interface (API) architecture for gathering automatically the information from different sources (e.g. information systems such as Enterprise Resource Planning, Project management systems, etc.). The first step is an ongoing work, and the second one is being investigated.

Acknowledgments. This work has been conducted in the context of the PLATINE project (PLAnning Transformation Interoperability in Networked Enterprises), financed by the national fund of research of the Grand Duchy of Luxembourg (FNR), under the grant C14/IS/8329172/R2.

References

1. Guédria, W., Naudet, Y., Chen, D.: Maturity model for enterprise interoperability. Enterp. Inf. Syst. **9**(1), 1–18 (2015)
2. Institute of Electrical and Electronics Engineers. IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries (1990)
3. Vernadat, F.B.: Interoperable enterprise systems: principles, concepts, and methods. Annu. Rev. Control **31**, 137–145 (2007)
4. Naudet, Y., Latour, T., Guédria, W., Chen, D.: Towards a systemic formalisation of interoperability. Comput. Ind. **61**(2), 176–185 (2010)
5. Panetto, H.: Towards a classification framework for interoperability of enterprise applications. Int. J. Comput. Integr. Manuf. **20**(8), 727–740 (2007). Taylor & Francis: STM, Behavioural Science and Public Health Titles
6. Ford. T.C.: Interoperability measurement, Ph.D. thesis, Department of the Air Force Air University, Air Force Institute of Technology (2008)
7. Cestari, José Marcelo A.P., Loures, Eduardo R., Santos, E.A.P.: Interoperability assessment approaches for enterprise and public administration. In: Demey, Y.T., Panetto, H. (eds.) OTM 2013. LNCS, vol. 8186, pp. 78–85. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41033-8_13
8. Leal, G., Guédria, W., Panetto, H., Lezoche, M.: Towards a comparative analysis of interoperability assessment approaches for collaborative enterprise systems. In 23rd International Conference on Transdisciplinary Engineering, pp. 45–54. IOS Press (2016)
9. Cornu, C., Chapurlat, V., Quiot, J.M., Irigoin, F.: Customizable interoperability assessment methodology to support technical processes deployment in large companies. Annu. Rev. Control **36**(2), 300–308 (2012)
10. Chen, D., Dassisti, M., Elvesaeter, B., Panetto, H., Daclin, N., Jaekel, F.W., Knothe, T., Solberg, A., Anaya, V., Gisbert, R.S., Kalampoukas, K., Pantelopoulos, S., Kalaboukas, K., Bertoni, M., Bordegoni, M., Cugini, U., Pulli, M., Perjons, E., Assogna. P.: Deliverable DI.2: Enterprise Interoperability-Framework and knowledge corpus-Advanced report. INTEROP Network of Excellence (2006)
11. European Commission. Annexe on the Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of Regions Towards interoperability for European public services (2010)

12. Alalwan, J., Thomas, M.: An ontology-based approach to assessing records management systems. e-Service J. **8**(3), 24–41 (2012)
13. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquisition **5**(2), 199–220 (1993)
14. Krivograd, N., Fettke, P., Loos, P.: Development of an intelligent maturity model-tool for business process management. In: 47th Hawaii International Conference of Systems Science, pp. 3878–3887. IEEE (2014)
15. Jardim-Goncalves, R., Agostinho, C., Steiger-Garcao, A.: A reference model for sustainable interoperability in networked enterprises: towards the foundation of EI science base. Int. J. Comput. Integr. Manuf. **25**(10), 855–873 (2012)
16. Department of Defense. C4ISR Architecture Working Group Final Report - Levels of Information System Interoperability (LISI). Washington, DC (1998)
17. Tolk, A., Muguira, J.A.: The levels of conceptual interoperability model. In: Fall Simulation Interoperability Workshop, USA (2003)
18. Clark, T., Jones, R.: Organizational interoperability maturity model for c2. In: Proceedings of the Command and Control Research and Technology Symposium, Washington (1999)
19. Yahia, E., Aubry, A., Panetto, H.: Formal measures for semantic interoperability assessment in cooperative enterprise information systems. Comput. Ind. **63**(5), 443–457 (2012)
20. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI for development: guidelines for process integration and product improvement. Pearson Education (2011)
21. Forrester, E., Buteau, B., Shrum, S.: CMMI for services: guidelines for superior service. Pearson Education (2011)
22. Gallagher, B., Phillips, M., Richter, K., Shrum, S.: CMMI for acquisition: guidelines for improving the acquisition of products and services. Addison-Wesley Professional (2011)
23. SCAMPI Upgrade Team, Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document (2011)
24. ISO/IEC 33020:2015: Information technology – Process assessment – Process measurement framework for assessment of process capability (2015)
25. Dietz, J.L.G.: Enterprise Ontology – Theory and Methodology. Springer, Heidelberg (2006)
26. Lu, Y., Panetto, H., Ni, Y., Gu, X.: Ontology alignment for networked enterprise information system interoperability in supply chain environment. Int. J. Comput. Integr. Manuf. **26**(1–2), 140–151 (2013)
27. Mohammad, M.A., Kaloskampis, I., Hicks, Y., Setchi, R.: Ontology-based framework for risk assessment in road scenes using videos. Procedia Comput. Sci. **60**, 1532–1541 (2015)
28. Palmer, C., Urwin, E.N., Niknejad, A., Petrovic, D., Popplewell, K., Young, R.I.: An ontology supported risk assessment approach for the intelligent configuration of supply networks. J. Intell. Manuf., 1–26 (2017)
29. Araujo Wickboldt, J., Armando Bianchin, L., Castagna Lunardi, R., Girardi Andreis, F., da Costa Cordeiro, W.L., Bonato Both, C., Zambenedetti Granville, L., Paschoal Gaspary, L., Trastour, D., Bartolini, C.: Improving IT change management processes with automated risk assessment. In: Bartolini, C., Gaspary, L.P. (eds.) DSOM 2009. LNCS, vol. 5841, pp. 71–84. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04989-7_6
30. Shrestha, A., Cater-Steel, A., Toleman, M.: Virtualising process assessments to facilitate continual service improvement in IT service management. In: The Australasian Conference on Information Systems (2015)
31. Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C.: A practical guide to building OWL ontologies using the protege-OWL Plugin and CO-ODE tools edition 1.0 (2004)
32. Protégé Editor and Knowledge Acquisition System (2010), http://protege.stanford.edu/
33. Yin, R.K.: Case Study Research: Design and Methods, 5th edn. Sage Publications (2013)

SPI and Functional Safety

How Does Scrum Conform to the Regulatory Requirements Defined in MDevSPICE®?

Özden Özcan-Top[1(✉)] and Fergal McCaffery[1,2]

[1] Regulated Software Research Centre and Lero,
Dundalk Institute of Technology, Dundalk, Ireland
{ozden.ozcantop, fergal.mccaffery}@dkit.ie
[2] STATSports Group, Dundalk, Ireland

Abstract. Medical device software development is subject to high regulations due to the potential risk of harming patients with unsafe medical devices. These regulations require software development to be performed with high discipline and evidence to be provided for auditory purposes. It's not easy to manage both conformance to regulations and efficiency in medical device development. Therefore, there is a transition towards agility in safety critical systems development, to build high quality systems, shorten time to market, improve customer and employee satisfaction and ensure both safety and reliability. In this study, we evaluated one of the most highly adopted agile software development methods, Scrum from a regulatory perspective. We investigated to what extend the regulatory requirements defined in MDevSPICE® are met with implementation of the Scrum method and what additional processes and practices have to be performed to ensure safety and regulatory compliance in the healthcare domain.

Keywords: MDevSPICE® · Scrum · Regulatory compliance · Safety critical domain · Agile software development

1 Introduction

The safety critical nature of medical device software requires Medical Device (MD) regulations are in place to ensure the safety of these devices. Manufacturers have to comply with the requirements to market an MD within a particular region. International standardizing bodies and regional regulatory authorities issue these requirements as standards or guidance. In the US, the Food and Drug Administration (FDA) issues the regulation through a series of official channels, including the Code of Federal Regulation (CFR) Title 21, Chap. 1, Subchapter H, Part 820 [1]. In the EU, the corresponding regulation is outlined in the general Medical Device Directive (MDD) 93/42/EEC [2], the Active Implantable Medical Device Directive (AIMDD) 90/385/EEC [3], and the In-vitro Diagnostic (IVD) Medical Device Directive 98/79/EC [4] - all three of which have been amended by 2007/47/EC [5].

Software development in medical domain is typically performed with traditional, plan-driven approaches like Waterfall and V-Model. The V-Model is perceived to be the *best fit* with regulatory requirements [6]. Some of the reasons why these methods are still valid today, despite their rigidness and limitations, can be listed as follows: (a) It is pretty

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 257–268, 2017.
DOI: 10.1007/978-3-319-67383-7_19

straightforward to produce the necessary deliverables required to achieve regulatory audits with these models. (b) Verification, validation and risk assessments are particularly important in medical device software development and these processes are planned and executed in parallel with a corresponding development phase of the V-Model. (c) In these models, each phase must be completed before the next phase begins. This approach works well when there is high confidence in the requirements defined.

Ensuring regulatory requirements continuously is only one of the challenges that medical companies face. Some of others are managing the change during development, being timely to market, ensuring high quality, safety and high productivity. Agile software development methods have positive results for overcoming these challenges [7]. Therefore, there is a transition going on in medical device development companies to achieve agility as well as safety and reliability.

In this study, we evaluated Scrum [8], to understand the level of regulatory compliance when they are implemented. A mapping between these methods and the medical device software process assessment framework, MDevSPICE® has been performed for this purpose. The second purpose of this research is to reveal additional practices that have to be performed to ensure compliance when there is no specific adaptation of Scrum for the medical domain.

The rest of the paper is structured as follows: In Sect. 2, we provide the background for this research which includes brief descriptions of MDevSPICE® and Scrum. We also provide a literature review of Scrum in the medical device development domain. In Sect. 3, we described the research methodology. In Sect. 4, we present the mapping and discuss the additional practices that have to be considered. In Sect. 5, we provide conclusions for this research.

2 Background

2.1 MDevSPICE®

MDevSPICE® is a medical device software process assessment framework developed with the purpose of integrating the regulatory requirements from the relevant medical device software standards and guiding medical device software developers to produce medical software that will be safe and reliable. It has been built upon 19 medical software development and software engineering standards, some of which can be seen on Fig. 1.

The MDevSPICE® process assessment model is a two-dimensional model of the process quality characteristic of process capability. In one dimension, the process dimension, the processes are defined. In the other dimension, the capability dimension, a set of process attributes are grouped into capability levels. Processes in this process assessment model are described in terms of their Purpose, Process Outcomes, Base Practices and Work Products. Although the set of Process Outcomes is necessary and sufficient to achieve the Purpose of the process, the Base Practices together with Work Products provide a possible way to achieve the Process Outcomes. The list of processes in MDevSPICE® process assessment model is given in Fig. 2.

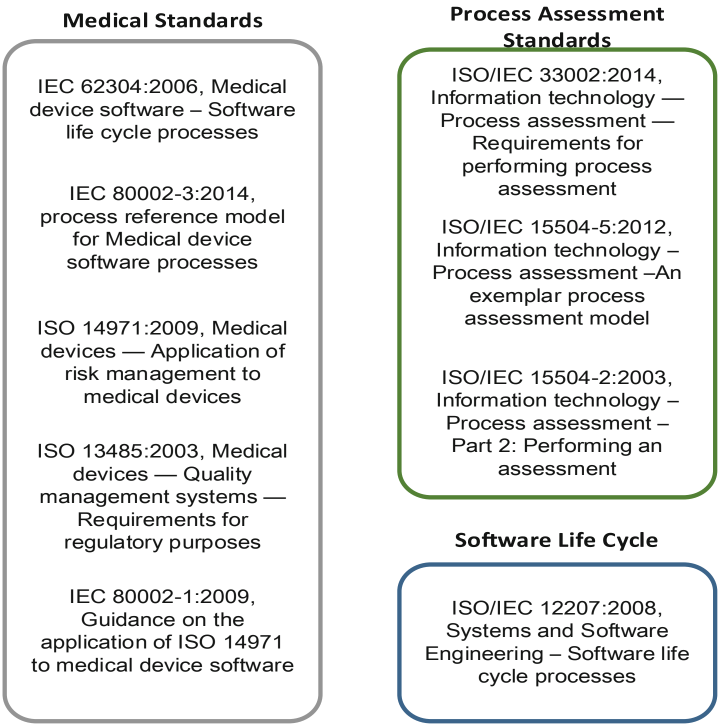

Fig. 1. Some of the standards and guidelines within MDevSPICE®

Safety classifications reflect the degree of harm that can result from medical device usage. Every medical device has to be assigned a safety class. Different international safety classification systems are in use throughout the world. There are three medical device safety classifications under US and EU regulations. Based on IEC 62304:2006, Class A devices are not intended to support or sustain human life, and may not lead to unreasonable risk of illness or injury. Class B medical devices may cause damage or harm to humans. Class C devices are usually those that support or sustain human life, and present a potential risk on illness or injury. Hand-held surgical instruments are Class A devices. An example of a Class B medical device is a powered wheelchair. An example of a Class C device is an implantable pacemaker.

The software safety classification of a medical device will determine the amount of IEC 62304 requirements that have to be fulfilled, with class A requiring much less practices to be put in place than for Class C. Additionally, the higher the safety classification the greater the amount of overhead associated with defining, implementing and providing objective evidence that the defined processes have actually been implemented. In this context, for each process, MDevSPICE® defines what outcomes have to be achieved and which base practices need to be performed for these safety classes.

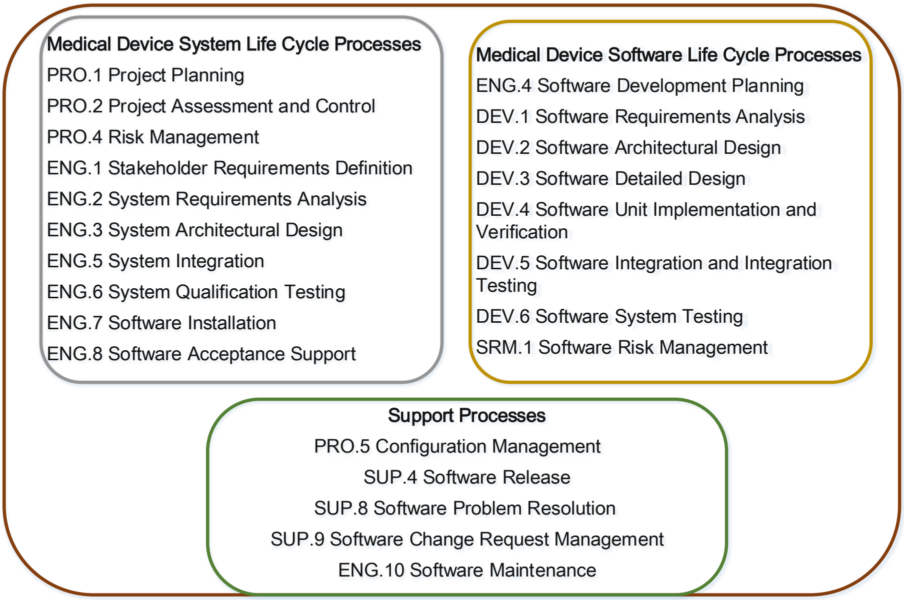

Fig. 2. MDevSPICE® processes

2.2 Scrum

Scrum was developed by Schwaber and Sutherland with the purpose of providing a management framework for software development [8, 9]. Scrum does not provide any specific technical practices for implementation.

The fundamental idea behind Scrum is to apply process control theory to software development to achieve flexibility, adaptability and productivity [7]. It relies on a set of values, principles and practices which can be adopted based on specific conditions. Scrum gives value on providing frequent feedback, embracing and leveraging variability, being adaptive, balancing upfront and just-in-time work, continuous learning, value-centric delivery and employing sufficient ceremony [10]. It offers effective solutions by providing specific roles, artifacts, activities and rules.

A Scrum Team consists of a Product Owner, a Scrum Master and the Development Team roles. Scrum Teams are self-organizing and cross-functional so that they could accomplish their work by themselves, rather than being directed by others outside the team and without depending on others not part of the team [9]. There are special events in Scrum which have been developed to create regularity and to minimize the need for meetings and are time-boxed.

2.3 Scrum Implementation in Safety Critical Domain

In the literature, we see many examples of Scrum implementation in the safety critical domain [11–16]. We briefly discuss some of these studies below:

Wolff [11] presents implementation of a formal specification language and Scrum with combination in an aircraft project. Executable specifications were used in order to validate system functionality, to understand the requirements and design of the system more precisely. In addition to conventional software implementation tasks within a sprint, formal specification investigation tasks were also defined.

Regulated Scrum [12] is an example of an adapted approach which has been implemented and validated in a highly regulated organization. Scrum was enhanced to ensure regulatory compliance in the medical domain. Some of the enhancements of the approach are having quality assurance people who ensure regulatory compliance at the end of each sprint (called continuous compliance), using templates to guide the development process, implementing coding standards and performing peer code review, establishing end-to-end traceability from the requirements elicitation stage to the code base with the help of tool support (called living traceability), risk management and continuous integration.

Another implementation of Scrum in a European space industry company with Test Driven Development, Continuous Integration and Pair Programming was discussed in [13]. Siemens Healthcare integrates Scrum into their software development process and additionally implements "feature orientation" practice to resolve the challenge of managing the flow of requirements coming from several product lines [14].

This literature review shows that Scrum was not used in the safety critical domain with their original versions, but, tailored for this domain and also combined with supplementary practices to ensure safety and regulatory compliance.

3 Research Approach

The purpose of this research is to reveal to what extend the regulatory requirements defined in MDevSPICE® are met when implementing Scrum. We defined the following research questions in relation to this purpose:

RQ1: How well the regulatory requirements of a safety Class B type medical device are met by through implementation of Scrum? ***RQ2:*** Which processes of MDev-SPICE® are covered by implementing Scrum? ***RQ3:*** Which base practices of MDev-SPICE® are covered by an implementation of Scrum? ***RQ4:*** What additional practices regarding those processes specified need to be performed in order to fully achieve a process at *Level 1: Performed Process*?

Research Steps

1. Listing Scrum practices at a fine granularity level.
2. Mapping MDevSPICE® base practices with Scrum Practices.
3. Identifying which processes were affected from the mapping.
4. Identifying the coverage ratio and deciding which MDevSPICE® base practices need to be included for those processes to satisfy a fully-achieved level.

Abrahamsson *et al.* [7], compared different agile software development methods to show which phases of software development were supported by these methods.

Based on the comparison, Scrum covers project management, requirements specification, integration test and system test phases

However, instead of selecting these processes mentioned above first, and then checking the coverage within MDevSPICE®, we preferred to do the mapping in the other way around. We first listed the Scrum practices and then mapped them to MDevSPICE® base practices. With this approach we were able to identify which processes of MDevSPICE® were covered with a basic Scrum implementation.

Limitation of the Research

Scrum could be taken as a prescriptive method with the descriptions of how the Scrum events will be performed and artifacts will be developed. However, Scrum is not defined at the practice description level provided by MDevSPICE®. Mapping of the method was limited to the given information in the following resource: The Scrum Guide™ by Ken Schwaber and Jeff Sutherland [17].

4 The Mappings and Discussions

Scrum and practices were mapped against MDevSPICE® (IEC 62304) Class B requirements. As the level of detail for the Scrum practices was limited, we needed to make some assumptions during the mapping. We assumed that process artifacts such as project plans or project monitoring reports would be developed during a Scrum implementation, as evidence required for the audits needed to be collected. Although it is very likely that some base practices would be performed during software development using Scrum, we couldn't rate a 100% coverage for them, as they might not be performed at the level of the detail required in MDevSPICE®. The coverage ratio is calculated based on the formula of: "the number of achieved base practices in a process/all base practices in a process".

4.1 Scrum Mapping

Scrum Method was described in terms of its roles, events and artifacts. Below, we provide the mapping for the roles and events. The artifacts which are basically product backlog and sprint backlog were not included in the mapping separately, as they were part of the events. Even though MDevSPICE® does not emphasize any specific roles, we mapped the activities that needs to be performed by the Scrum roles to the base practices of MDevSPICE®, shown in Table 1. In Table 2, the mapping between the Scrum events and the MDevSPICE® Processes and Base Practices are provided (**RQ2–RQ3**). The bold written text in the 3rd column of Tables 1 and 2 show the mapped processes. The other text in the same column refer to the mapped base practices (BPs).

According to the mapping shown in Tables 1 and 2, Scrum is related to 5 processes of MDevSPICE® when it is implemented fully (RQ2). Within the mapping process, we also evaluated and calculated the coverage ratio of the MDevSPICE® base practices for Scrum. Table 3, shows the coverage ratio for each mapped process. The coverage evaluation performed by one of the authors for base practices from a Scrum perspective, was subjective, but peer reviewed by the other author. Therefore, depending on

Table 1. Mapping of scrum roles‘activities and MDevSPICE® processes & base practices

Scrum roles	Specific activities of the roles	MDevSPICE® processes & base practices
Product owner	"— deciding which features and functionality to build and the order in which to build them — communicating to all other participants a clear vision of what the Scrum team is trying to achieve — being responsible for the overall success of the solution being developed or maintained"	**PRO.1 Project planning** PRO.1.BP1: Define the scope of work PRO.1.BP3: Evaluate feasibility of the project PRO.1.BP6: Define needs for experience, knowledge and skills PRO.1.BP7: Identify and monitor project interfaces PRO.1.BP9: Allocate resources and responsibilities PRO.1.BP11: Implement the project plan **ENG.1 Stakeholder requirements definition** ENG.1.BP1: Identify stakeholders
Scrum master	"— helping everyone involved understand and embrace the Scrum values, principles, and practices — helping the organization through the challenging change management process that can occur during a Scrum adoption — protecting the team from outside interference and takes a leadership role in removing impediments that inhibit team productivity"	**PRO.1 Project planning** PRO.1.BP2: Define life cycle model for the project
Dev-team	"— a diverse, cross-functional collection of these types of people who are responsible for designing, building, and testing the desired product"	**PRO.1 Project planning** PRO.1.BP4: Define and maintain estimates for project attributes PRO.1.BP5: Define project activities and tasks PRO.1.BP8: Define project schedule PRO.1.BP10: Establish project plan PRO.1.BP11: Implement the project plan

Table 2. Mapping of scrum events and MDevSPICE® processes & base practices

Scrum events	Descriptions of the events	MDevSPICE® processes and base practices
Sprint planning	"The work to be performed in the Sprint is planned at the Sprint Planning. This plan is created by the collaborative work of the entire Scrum Team"	**PRO.1 Project planning** PRO.1.BP4: Define and maintain estimates for project attributes PRO.1.BP5: Define project activities and tasks PRO.1.BP7: Identify and monitor project interfaces
Daily Scrum	"A 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 h"	**PRO2. Project assessment and control** PRO.2.BP3: Report progress of the project PRO.2.BP4: Perform project review
Sprint review	"A meeting held at the end of the Sprint to inspect the Increment and adapt the Product Backlog. The timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product are reviewed"	**PRO2. Project assessment and control** PRO.2.BP1: Monitor project attributes PRO.2.BP2: Monitor project interfaces PRO.2.BP3: Report progress of the project PRO.2.BP4: Perform project review PRO.2.BP5: Act to correct deviations
Sprint retrospective	"A meeting to inspect how the last Sprint went with regards to people, relationships, process, and tool"	**PRO2. Project assessment and control** PRO.2.BP6: Collect project experiences
Product backlog grooming	"Product Backlog (PB) is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product." "PB lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value" "PB Grooming is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Dev-Team perform"	**ENG.1 Stakeholder requirements definition** ENG.1.BP2: Obtain requirements ENG.1.BP3: Define constraints ENG.1.BP4: Define user interaction ENG.1.BP5: Identify critical requirements ENG.1.BP6: Evaluate requirements ENG.1.BP7: Agree on requirements **ENG.2 System requirements analysis** ENG.2.BP1: Establish system requirements ENG.2.BP3: Optimize project solution ENG.2.BP4: Analyze system requirements ENG.2.BP5: Evaluate and update system requirements ENG.2.BP7: Communicate system requirements **DEV.1 Software requirements analysis** DEV.1.BP1: Define and document all software requirements DEV.1.BP2: Prioritize requirements DEV.1.BP6: Evaluate and update requirements DEV.1.BP7: Baseline and communicate software requirements

the implementation details and perception of the methods, different coverage ratios than we provided could be obtained. However, the purpose of giving this ratio is to provide readers and practitioners with an indication of how much value is achieved with basic Scrum implementation and how much needs to be done more from a regulatory perspective.

Below, we discuss why processes #3, #4, and #5 in Table 3 did not have a full coverage ratio and what additional practices need to be performed for compliance to medical requirements (**RQ4**).

Table 3. Coverage of mapped MDevSPICE® processes from scrum perspective

	Mapped MDevSPICE® processes	Coverage ratios
1.	PRO.1 Project planning	100%
2.	PRO.2 Project assessment and control	90%
3.	ENG.1 Stakeholder requirements definition	55%
4.	ENG.2 System requirements analysis	71%
5.	DEV.1 Software requirements analysis	33%

#3 ENG.1 Stakeholder Requirements Definition Process: (Coverage Ratio: 5 BPs/9 BPs). The following base practices of ENG.1 are assumed to be achieved by the product owner and the development team in product backlog grooming sessions: *ENG.1.BP1: Identify stakeholders, ENG.1.BP2: Obtain requirements, ENG.1.BP3: Define constraints, ENG.1.BP6: Evaluate requirements, ENG.1.BP7: Agree on requirements*. However, the other base practices of this process need special attention which are not addressed in Scrum.

For an IEC 62304 Class B type medical software, user interaction has to be defined and evidence has to be provided. Based on the *ENG.1.BP4: Define user interaction* base practice the following information has to be defined for a medical device:

– Intended medical indication, e.g. conditions(s) or disease(s) to be screened, monitored, treated, diagnosed, or prevented; – Intended patient population, e.g. age, weight, health, condition; – Intended part of the body or type of tissue applied to or interacted with; – Intended user profile; – Intended conditions of use, e.g. environment including hygienic requirements, frequency of use, location and mobility; and – Operating principle.

In a product backlog grooming session, we may assume that all stakeholder requirements are specified. However, as part of the *ENG.1.BP5: Identify critical requirements* practice of MDevSPICE®; it has to be ensured that *health, safety, security, environment and other stakeholder requirements and functions that relate to critical qualities and shall address possible adverse effects of use of the system on human health and safety* are identified as well.

In medical device software development, every change on the product, whether it is on the artifacts or the code has to be made in a controlled way. This is one of the major contradictions between agile and the regulated worlds. For a change to be controlled, a version control system should be in place and baselines established. This is referred to in *ENG.1.BP8: Establish stakeholder requirements baseline* base practice. However, a product backlog is a dynamic list which is continuously changing and no baselines are taken over it.

The other major requirement in medical device software development is to build traceability links between artifacts as this plays a significant role in defect management and change management. This is referred to in *ENG.1.BP9: Manage stakeholder requirements changes*. The purpose is to "Maintain stakeholder requirements traceability to the sources of stakeholder need". However, there is no specific emphasis on the development of a traceability schema in Scrum method.

#4 ENG.2 System Requirements Analysis Process: (Coverage Ratio: 5 BPs/7 BPs). We may assume that base practices: *ENG.2.BP1: Establish system requirements, ENG.2.BP3: Optimize project solution, ENG.2.BP4: Analyze system requirements, ENG.2.BP5: Evaluate and update system requirements, ENG.2.BP7: Communicate system requirements* are performed in product backlog grooming sessions, as there are mechanisms to achieve them. However, the following two base practices need to be handled separately.

As part of *ENG.2.BP2: Assign a safety class to the medical device based on the regional regulations* process, at the system requirements analysis phase, a safety class has to be assigned to the product as the specific regulations apply based on the safety class in order to prevent potential harm to human life. As mentioned also in base practice ENG.1.BP9, bilateral traceability between the stakeholder requirements and the system requirements needs to be established as part of *ENG.2.BP6: Ensure consistency* base practice.

#5 DEV.1 Software Requirements Analysis Process: (Coverage Ratio: 3 BPs/9 BPs) We assumed that base practice, *DEV.1.BP1: Define and document all software requirements* is partially achieved, as there are specific issues that needs to be addressed for this BP. Based on FDA rules, software requirements have to be documented in a software requirements specification document and this document should contain details of the software functions.

It is important to determine the interfaces between the software requirements and other elements of the operating environment such as third party software. This is achieved as part of base practice, *DEV.1.BP3: Determine the impact the requirements have on the operating environment*. At this stage, it is expected that the acceptance criteria for the software tests are defined from software requirements *(DEV.1.BP4: Develop acceptance criteria for software testing based on the software requirements.)* Scrum does not have such a rule.

As mentioned above, consistency of system requirements to software requirements has to be ensured. This is achieved through establishing and maintaining bilateral traceability between system requirements and the software requirements *(DEV.1.BP5: Verify all software requirements.)*

The 7th base practice of DEV.1 requires establishing a baseline of software requirements and also providing communication of the software requirements. Due to use of communication channels in Scrum, we feel that the second part of this base practice can be achieved. However, the baseline of software requirements should also be added.

In medical device software development, special attention is given to risk analysis and mitigation. With base practices, *DEV.1.BP.9: Re-evaluate and maintain medical* *device risk analysis* and *DEV.1.BP8: Establish and maintain risk control measures in software requirements*, it is ensured that risks regarding the software requirements are identified and risk control measures are defined. Risk management should be a part of daily or weekly Scrum review meetings.

Although we have mapped the Stakeholder, System and Software Requirements Analysis processes with the product backlog grooming practice in Scrum, it is necessary to ensure that distinction between these requirement types are clear, the traceability links are established, and the changes made to them is managed.

In MDevSPICE®, there another process, **ENG.4 Software Development Planning** includes very specific practices for regulatory requirements compliance. Some of these base practices include *assigning the software safety class of the software system, having a software integration test plan, a verification plan, a software risk management plan and configuration management plan.* Although Scrum proposes effective ways to manage projects, these plans are not part of a basic Scrum method. Therefore we assumed that *ENG.4 Software Development Planning* is not covered with Scrum, even though it is a "planning" process.

5 Conclusions

In this study, we evaluated if a Scrum implementation could meet the regulatory requirements defined in MDevSPICE®, the software process assessment framework for medical device software development. Scrum was selected due to its high recognition and adoption in software development world. The research approach included the mapping of Scrum practices to MDevSPICE® processes and base practices. With this approach, we were able to define MDevSPICE® processes and base practices that could be achieved in a basic Scrum implementation, more importantly the additional base practices that have to be performed for ensuring safety and regulatory compliance.

We also identified the coverage ratio of MDevSPICE® processes from a Scrum perspective. Even though the coverage ratios are calculated from a subjective point of view, they provide important information to readers and practitioners about which MDevSPICE® processes are covered to what extent.

The significance of this study is that it presents a coverage analysis at the MDevSPICE® base practice level which is very detailed and has never been performed before. The coverage ratios showed the level of the gap between methods. The study has also revealed conflicting practices such as "controlled change management over continuous and dynamic change". In addition, the discussions made around the additional practices that need to be performed, complete the missing pieces to ensure safety and be successful over a regulatory audit in the medical device domain. The results of this study also provide guidance us for the development of an agile integrated medical device software development framework.

As future work we will extend the mapping by adding XP, other agile methods which propose a whole software development life cycle coverage such as Dynamic Systems Development Method and scaling agile frameworks such as Disciplined Agile Delivery and SAFE.

Acknowledgement. This research is supported by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund through Lero - the Irish Software Research Centre (http://www.lero.ie) grant 13/RC/2094. This research is also partially supported by the EU Ambient Assisted Living project – Maestro.

References

1. FDA: Chapter I - Food and drug administration, department of health and human services subchapter H - Medical devices, Part 820 - Quality system regulation, May 2015. http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820
2. Directive 93/42/EEC of the European Parliament and of the Council concerning medical devices (1993)
3. Council directive 90/385/EEC on active implantable medical devices (AIMDD) (1990)
4. Directive 98/79/EC of the European parliament and of the council of 27 October 1998 on in vitro diagnostic medical devices (1998)
5. Directive 2007/47/EC of the European Parliament and of the Council concerning medical devices (2007)
6. Mc Hugh, M., Cawley, O., McCaffcry, F., Richardson, I., Wang, X.: An agile v-model for medical device software development to overcome the challenges with plan-driven software development lifecycles. In: 2013 5th International Workshop on Software Engineering in Health Care (SEHC), pp. 12–19. IEEE (2013)
7. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development Methods: Review and Analysis. VTT Finland (2002)
8. Schwaber, K.: Scrum development process. In: Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G. (eds.) Business Object Design and Implementation, pp. 117–134. Springer, London (1997). doi:10.1007/978-1-4471-0947-1_11
9. Schwaber, K., Sutherland, J.: Software in 30 days: How Agile Managers Beat the Odds, Delight their Customers, and Leave Competitors in the Dust. Wiley, Hoboken (2012)
10. Rubin, K.S.: Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional, Boston (2012)
11. Wolff, S.: Scrum goes formal: agile methods for safety-critical systems. In: Proceedings of the First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches, pp. 23–29. IEEE Press (2012)
12. Fitzgerald, B., Stol, K.-J., O'Sullivan, R., O'Brien, D.: Scaling agile methods to regulated environments: an industry case study. In: 2013 35th International Conference on Software Engineering (ICSE), pp. 863–872. IEEE (2013)
13. Ahmad, E., Raza, B., Feldt, R., Nordebäck, T.: ECSS standard compliant agile software development: an industrial case study. In: Proceedings of the 2010 National Software Engineering Conference, p. 6. ACM (2010)
14. Kircher, M., Hofman, P.: Combining systematic reuse with agile development: experience report. In: Proceedings of the 16th International Software Product Line Conference, vol. 1, pp. 215–219. ACM (2012)
15. Faber, R.: Architects as service providers. IEEE Softw. **27**(2), 33–40 (2010)
16. Spence, J.W.: There has to be a better way![software development]. In: 2005 Proceedings of Agile Conference, pp. 272–278. IEEE (2005)
17. Sutherland, J., Schwaber, K.: The scrum guide. The Definitive Guide to Scrum: The Rules of the Game. Scrum.org (2013)

Testing in Automotive SPICE and TestSPICE: Synergies and Benefits

Tomas Schweigert[1(✉)] and Klaudia Dussa-Zieger[2]

[1] SQS Software Quality Systems, Cologne, Germany
tomas.schweigert@sqs.com
[2] imbus AG, Möhrendorf, Germany
klaudia.dussa-zieger@imbus.de

Abstract. The paper describes the overall structure of Automotive SPICE and TestSPICE. It presents a comparison between the processes in Automotive SPICE® and TestSPICE and then focuses on the testing processes and some specific test topics. The paper shows potential benefits of TestSPICE for the automotive industry. It deals with topics like test data management, test automation and test techniques, which are not or only roughly mentioned in Automotive SPICE. The paper also gives a short explanation of the agile extension of TestSPICE.

Keywords: TestSPICE · Automotive SPICE · ISO 29119 · Process assessment · Process improvement · Testing process · Agile

1 Introduction

Automotive SPICE is a very well-known and well-established assessment model in the automotive domain which is compliant to ISO/IEC 15504 (colloquially also called the SPICE standard). It has been developed to assess the development processes of an automotive supplier. This includes the core engineering processes as well as supporting processes, management processes and so on. Automotive SPICE has been around for more than ten years now. Its first version 2.3 was published in 2005, while the current version 3.0[1] was released in July 2015.

TestSPICE has been developed since 2010 [8–15]. It is also compliant to ISO/IEC 15504, but topic-wise TestSPICE is solely focused on testing. It describes in great detail the core processes in testing and also highlights different specific aspects, e.g. test data management or test automation.

While Automotive SPICE does include five testing processes, the description of these processes remains very generic. If a supplier wants to know how well he is doing with respect to specific test topics, e.g. regression testing, he will find no guidance or help within Automotive SPICE.

Therefore the idea came up to compare the two assessment models and where appropriate combine the models and define the process scope in an assessment according to the needs of the customer.

[1] Automotive SPICE 3.0 is compliant to the latest version of the SPICE standard ISO/IEC 33004.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 269–278, 2017.
DOI: 10.1007/978-3-319-67383-7_20

The paper first briefly describes the overall architecture of Automotive SPICE and TestSPICE. It will then give a high-level comparison and further go into detail on some topics. The paper closes with a summary and conclusions.

2 The Overall Architecture of Automotive SPICE

The Automotive SPICE process assessment model (PAM) is intended for use when performing assessments of the process capability on the development of embedded automotive systems [1, 3].

The processes within Automotive SPICE are structured using three process categories, namely the Primary Life Cycle Process, the Organizational Life Cycle Processes and the Supporting Life Cycle processes, and within the categories process groups are used for clustering, see Fig. 1. Overall architecture of Automotive SPICE.

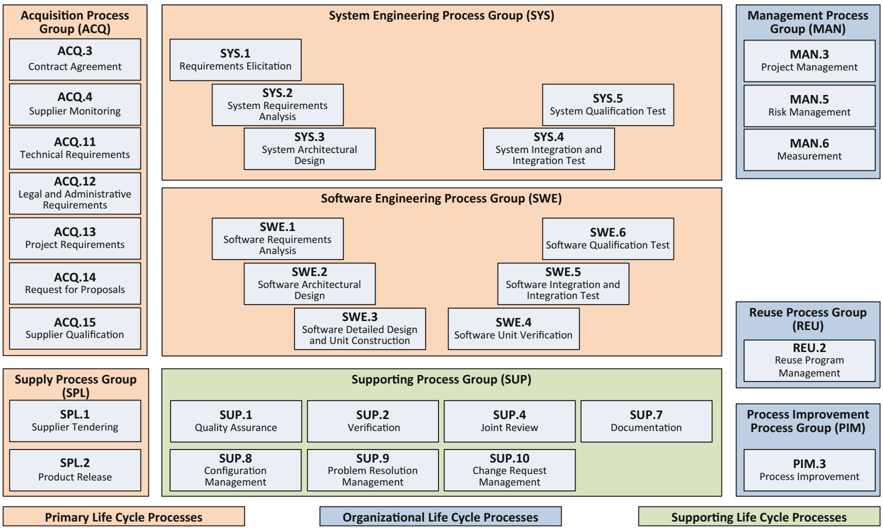

Fig. 1. Overall architecture of Automotive SPICE. Source: VDA QMC Provisional Automotive SPICE Training

Altogether 32 processes are included in the model. All processes are described compliant to the SPICE standard with a purpose statement, process outcomes and their indicators, i.e. base practices, generic practices and work products.

From the point of view of testing the interesting processes are located within the Software and System Engineering Process Group. For the purpose of illustration the processes within these two process groups are depicted in a V-shape. However, Automotive SPICE does not prescribe a particular life cycle model. Due to the graphical representation the testing processes are all located on the right hand side of the V. Automotive SPICE distinguishes testing activities at different points in development, from software unit tests all the way to system qualification tests.

Looking closely at the different test processes Automotive SPICE requires a test strategy, test execution and test documents, but does not prescribe any test methods. The greatest level of details is a reference to the new testing standard ISO 29119-3 on test documentation [6].

3 The Overall Architecture of TestSPICE

The TestSPICE process assessment model is intended for use when performing assessments of the process capability in testing projects [2, 4]. Analogous to Automotive SPICE the model is structured using process categories and corresponding process groups. The process categories are the Business Life Cycle processes, Technical Life Cycle processes and Agreement Life Cycle processes, see Fig. 2. In addition the model provides Agile Extension processes which are described in an informative annex.

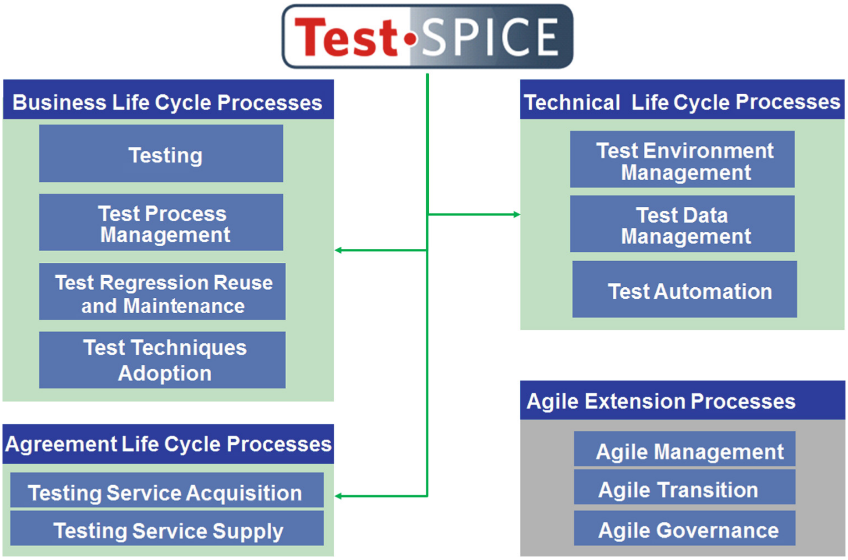

Fig. 2. Overall architecture of TestSPICE

Within each process category further process groups have been defined clustering the different aspects in testing, e.g. Test Data Management.

TestSPICE is an assessment model which stands by itself. However, it does acknowledge other standards in the testing area. For example, the Test Technique Adoption process group directly reflects the test techniques which are described in ISO 29119 part 4 [7]. In addition, a comparison to the test processes defined in ISO 29119 part 2 [5] is provided as an informative annex of the model.

4 Comparing Automotive SPICE and TestSPICE

When comparing both models first a high-level comparison is presented. In a second step some differences and similarities are highlighted.

4.1 High-Level Comparison Between Automotive SPICE and TestSPICE

The following table shows a high-level comparison between Automotive SPICE and TestSPICE along the different process groups, some individual processes and a few selected topics from the testing world.

As can be seen in the table a number of processes are handled similarly by both models while certain processes are only covered in one of either model. For example the supporting process and generic project management processes are only covered in Automotive SPICE. Processes with more details on some test topics are only covered in TestSPICE, e.g. Test Environment Management, Test Data Management or Test Automation. (see Table 1)

Table 1. High-level comparison between Automotive SPICE and TestSPICE

TOPIC	AUTOMOTIVE SPICE	TESTSPICE	COMMENT
Acquisition	Very elaborated due to the needs of the automotive industry	Traditional model, but focused on testing only	Elaboration not needed to understand the acquisition of test service providers
Supply	Traditional model	Traditional Model, but focused on testing only	
Supporting processes	Full set of SPICE processes	Not included	The idea of TestSPICE is to provide a pure testing model. Other processes may be used as plugin when scoping the assessment
Management Processes	Subset of SPICE Processes	Not included	The idea of TestSPICE is to provide a pure testing model. Other processes may be used as plugin when scoping the assessment
Process Improvement	Subset of SPICE processes	Not included	The idea of TestSPICE is to provide a pure testing model. Other processes may be used as plugin when scoping the assessment

(*continued*)

Table 1. (*continued*)

TOPIC	AUTOMOTIVE SPICE	TESTSPICE	COMMENT
Reuse	Reuse program Management	Reuse Program management	
Test Strategy	Each test process has a strategy practice	Test Strategy development and Test Strategy deployment on organizational level	Vital part of understanding the test process
Test Planning	Not explicitly addressed, must be seen as part of project management	Test Requirements analysis, Test Planning, Test Monitoring & Control	In the assessment: of Automotive SPICE: PA 2.1
Test Preparation	Develop specification, select test cases	Provision of required Test Inputs (the Test Basis), Test Analysis & design Test Environment Design (and Configuration Planning), Test Data Provision Planning, Test Automation Design	Core of the TestSPICE Model, processes much more elaborated
Test Execution	Perform the test	Test Realization and Execution, Test Results Ana-lysis and Reporting	TestSPICE allows a deep dive
Test environment management	Implicitly	Complete process group	Test environments are a critical part of each testing. Automotive SPICE: PA 2.1, 3.1, 3.2
Test Data management	None	Complete process group	Test environments are a critical part of each testing.
Test Automation management	None	Complete process group	Vital if improvement addresses efficiency
Test Techniques	None	All Techniques mentioned in ISO/IEC/IEEE 29119	Helps to choose the right and adequate testing techniques
Agile	None	Three process groups as informative extension to the model	Helps to make decisions, support the transition and the day to day work in agile organizations.

4.2 Detailed Comparison

Test Strategy

On the first view the difference seems clear. While in TestSPICE the test strategy is an organizational topic, test strategy in Automotive SPICE is a task to be done at every test stage[2]. But this is not completely correct as the "Organizational Test Strategy Development" process includes a practice "Define test stages and their content". The idea behind this approach is not to waste time and energy in endless discussions about test stage naming and content, but to check if test stages and their content are defined. The practice also requires to define the goals, responsibilities and main activities of each test stage. The goal could be derived the purpose statement of the test processes of Automotive SPICE.

In real life project we have another problem. Test stages and testing combinations often do not follow the formal Automotive SPICE process boundaries due to the nature of the product, project size or organizational limitations.

For example:

- SW can only be tested as a whole: SWE.5 and SWE.6 overlap
- SW integration cannot be tested in isolation and is tested together with system integration: SWE.6 and SYS.4/SYS.5 overlap
- For HW (e.g. electrical, hydraulics) integration testing cannot be isolated from system testing: SYS.4 and SYS.5 overlap
- System requirements and software requirements are identical to a certain extent and therefore tested all in one: SWE.6 and SYS.5 overlap

This shows that a supplier needs an organizational test strategy to deal with these issues.

TestSPICE provides a complete set of tasks to develop a sound test strategy:

- Establish goals for the test process
- Define test stages and their content
- Define test methods
- Define the frame conditions of the project specific tailoring
- Identify the legal and organizational requirements
- Define a test automation approach
- Synchronize technical testing approaches
- Develop guidelines for the resolution of technical conflicts

Automotive SPICE provides a template for a test strategy in the automotive domain (by reference to ISO 29119-3). Trying to implement this template 1:1 without a detailed advice will lead to unforeseen difficulties. Using TestSPICE in addition will help to develop a test strategy that identifies and solves issues early and takes all test stages into account. As a result assessments can be scoped correctly and evidences are correctly mapped to testing processes.

[2] In TestSPICE the term „test stage" is synonymous to „test level".

Test environment management

Software and system testing require well defined test environments. This sounds trivial, but in the complex organizational environment of the automotive domain the ownership of and the responsibility for the test environments need to be clearly defined. Remember that during the integration of embedded automotive system a number of test environments, i.e. specialized technical test benches, are in use which often present a scarce resource.

Automotive SPICE does not have explicit Test Environment Management processes. But the issue is implicitly addressed in

- GP 2.1.6 Identify, prepare, and make available resources to perform the process according to plan.
- GP 3.1.4 Identify the required infrastructure and work environment for performing the standard process.
- GP 3.2.5 Provide adequate process infrastructure to support the performance of the defined process.

TestSPICE uses a high elaborated model of test environment management:

- Collect and analyze requirements regarding the impact on the test environment
- Design and configure the test environment
- Assemble the test environment
- Test the test environment
- Operate the test environment
- Support the test environment users
- Disassemble the test environment

The implementation of these processes makes sure that test environment is defined and in place when the test are executed, therefore enhancing a smooth and efficient test execution.

Test data management

Automotive SPICE does not give any advice regarding test data. Test data are mentioned as potential criteria for unit testing and as a topic of the test plan.

TestSPICE includes a complete test data management process group. This is mainly triggered by the fact that in the financial industry complex data bases are used for testing. But also in the automotive industry practical experience in assessment interviews shows that test data become more and more important.

The Test Data Management process group consists of the following processes:

- Test Data Requirements Management
- Test Data Provision Planning
- Test Data Set Up
- Test Data Maintenance and Support

Analogous to the management of the test environment, test data management supports the efficient and smooth execution of the tests.

Test automation

Test automation is not mentioned in Automotive SPICE but it is described as a good practice in the Automotive SPICE provisional Assessor Training. Test automation in general makes regression testing more efficient, in case of unit testing it enables the regression testing of small software increments.

Test automation is a part of the technical life cycle of the TestSPICE model. It is obvious that test automation has strong dependencies to test environment and test data management. In earlier times test automation was something like capture and replay of test cases. In modern times complex and high performing frameworks are used to automate tests.

As a result the Test Automation Process Group consists of the following processes:

- Test Automation Needs & Requirements Elicitation
- Test Automation Design
- Test Automation Implementation
- Test Case Implementation
- Test Automation Usage
- Test Automation Process Monitoring.

If it is intended to use a framework, the test automation design and the test automation implementation process deal with the setup of the framework, while the test case implementation process deals with the automation of single test cases. Automated test cases are a valuable help during test execution. This is true for the automotive industry as well as for other industries.

Test techniques

ISO 29119-4 deals with test techniques, showing a huge list of potential test techniques. Automotive SPICE mentions the use of test techniques but does not give any hint which techniques are adequate or which are not.

TestSPICE does also not advocate the use of a specific test technique either, but advocates the explicit decision making. Starting with the organizational test strategy development process, where the "Define test methods" is mentioned as a base practice,

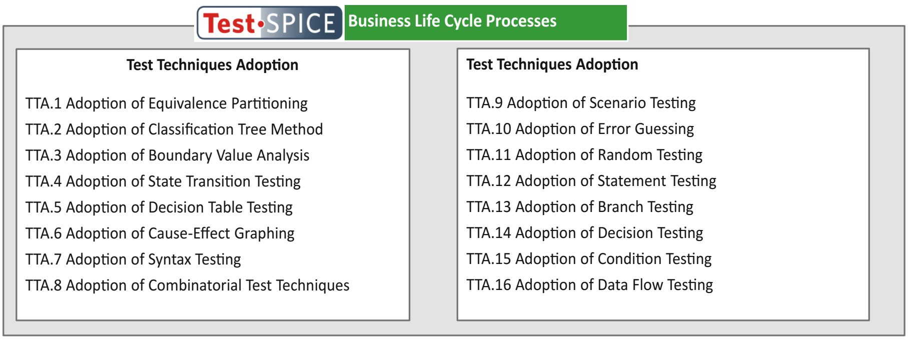

Fig. 3. Test Techniques Adoption process group

a set of detailed selection processes is described in the Test Techniques Adoption process group for each test technique mentioned in ISO 29119-4 Fig. 3.

TestSPICE Support for agile projects

Even if agile is one of the current big hypes, Automotive SPICE does not deal with any type of agile approach. It could be said that Automotive SPICE neither advocates agile nor does it advocate more traditional life cycles.

TestSPICE does also not advocate any life cycle but it deals with the complaint of the agile community that traditional assessment models like CMMI [16] or SPICE do not support agile development. Neither to promote, nor to disregard agile development, the TestSPICE SIG decided to handle this issue in an annex that contains the agile processes.

The architecture of the agile extension was driven by the fact that many organizations face the plugin of agile practices in some projects, but have neither an idea how to run agile projects nor have ideas of becoming an agile organization.

The agile extension therefore consists of three process groups on project, tactical and strategic level:

- The Agile Management Process Group (AMP)
- The Agile Transition Process Group (ATP)
- The Agile Governance Process Group (AGP)

These processes help an organization through its way from strategic decision making regarding agile over the implementation of agile in an organization to run agile projects. Especially in the automotive domain where the software department leans towards agile development, while hardware and mechanics mostly develop traditionally, help on this topic is invaluable.

5 Summary and Conclusions

Looking at both models a number of facts can be summarized. Both models comply to the requirements set up by the SPICE standards. Both models use capability as their quality characteristic. Therefore both assessment models are compatible and the processes coming from both models can be combined as seen fit for an assessment.

However, the scope or the goal of both models is different. While Automotive SPICE encompasses all development, management and supporting processes for an embedded automotive system, TestSPICE strongly focuses on the testing processes in all related aspects. In other words TestSPICE refines the testing processes which are described on a coarse level in Automotive SPICE and adds some more test-related processes.

Taking this into account a natural conclusion is to combine processes from both models where the respective expertise is required. If a customer is asking for an assessment of his development processes, but also wants to assess his test automation approach, this can be done by adding the TAU processes to the Automotive SPICE processes. Vice versa, if in case of a TestSPICE assessment the customer also wants to look at quality assurance, SUP.1 process can be incorporated in the assessment scope.

Automotive SPICE and TestSPICE complement each other nicely. A combined usage of their processes presents benefits for all parties involved and therefore this approach should be further pursued.

References

1. Automotive SPICE v3.0. http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf. Accessed 21 June 2017
2. TestSPICE PAM v3.1. https://www.testspice.infolast. Accessed 21 June 2017
3. Höhn, H., Sechser, B., Dussa-Zieger, K., Messnarz, R., Hindel, B.: Software Engineering nach Automotive SPICE, 1st edn. Dpunkt Verlag, Heidelberg (2009)
4. Dussa-Zieger, K., Ekssir-Monfared, M., Schweigert, T., Philipps, M., Blaschke, M.: The current status of the TestSPICE project. In: Systems, Software and Services Process Improvement 21th European Conference, EuroSPI (2017)
5. ISO/IEC/IEEE 29119-2:2013 – Software and systems engineering – Software testing – Test process
6. ISO/IEC/IEEE 29119-3:2013 – Software and systems engineering – Software testing – Test documentation
7. ISO/IEC/IEEE 29119-4:2015 – Software and systems engineering – Software testing – Test techniques
8. Schweigert, T., Nehfort, A., Ekssir-Monfared, M.: The feature Set of TestSPICE 3.0, The 21st Conference EuroSPI, pp. 309–316, ISBN 978-3-662-43895-4
9. Schweigert, T., Ekssir-Monfared, M., Ofner, M.: An agile Management Process Group for TestSPICE®, The 20th Conference EuroSPI, pp. 238–236, ISBN 978-3-642-39178-1
10. Blaschke, M., Philipp, M., Schweigert, T.: Get the Test Process under Control The TEST SPICE approach, Proceedings of the 2010 Spice Days.(el. Published)
11. Blaschke, M., et al.: The TestSPICE approach, Test Process Assessments follow in the footsteps of software process assessments, Testing Experience 12/2009 S. 56ff
12. Knüvener, C.: TestSPICE – SPICE für Testprozesse, SQ-Magazin 17/2010 S; 26–27
13. Schweigert, T., Nehfort, A.: Technical Issues in Test Process Assessment and their current and future Handling in Test SPICE, EuroSPI Industrial Proceedings, Delta (2011)
14. Steiner, M., et al.: Make test process assessment similar to software process assessment - the TestSPICE approach. J. Softw. Maint. Evol. published online 2010 at Wiley online library, wileyonlinelibrary.com. doi:10.1002/SMR507
15. Alone, S., Glocksien, K.: Evaluation of Test Process Improvement Approaches: An Industrial Case Study. Göteborg, Sweden (2013)
16. CMMI® for Development, Version 1.2, Carnegie-Mellon, Software Engineering Institute, CMU/SEI-2006-TR-008, http://www.sei.cmu.edu/reports/06tr008.pdf. Accessed 17 July 2017

Deep Learning in Automotive: Challenges and Opportunities

Fabio Falcini and Giuseppe Lami(✉)

Consiglio Nazionale delle Ricerche,
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo",
Via Moruzzi 1, 56124 Pisa, Italy
{fabio.falcini,giuseppe.lami}@isti.cnr.it

Abstract. The interest of the automotive industry in deep-learning-based technology is growing and related applications are going to be pervasively used in the modern automobiles. Automotive is a domain where different standards addressing the software development process apply, as Automotive SPICE and, for functional safety relevant products, ISO 26262. So, in the automotive software engineering community, the awareness of the need to integrate deep-learning-based development with development approaches derived from these standards is growing, at the technical, methodological, and cultural levels. This paper starts from a lifecycle for deep-learning-based development defined by the authors, called W-model, and addresses the issue of the applicability of Automotive SPICE to deep-learning-based developments. A conceptual mapping between Automotive SPICE and the deep learning lifecycles phases is provided in this paper with the aim of highlighting the open issues related to the applicability of automotive software development standards to deep learning.

Keywords: Deep learning · Automotive SPICE · Software development lifecycle · ADAS (advanced driver assistance systems) · W model

1 Introduction

In the last 20 years automotive witnessed a continuous trend to innovation. In such a period of time, cars moved from being basically mechanical/electromechanical devices, to being very complex vehicles where electronics and software are playing a predominant role. Electronics is so pervasive in today's cars that almost all the main features and functionalities are controlled by software.

Because software, car manufacturers today have to face several challenges. Mastering functional safety and cyber-security issues are the today's hottest challenges for car manufacturers (as well as the daily work for engineers and managers). Nevertheless, while the functional safety and cyber-security are still two open battlefronts, a new one is going to be open and it will determine in the next future a revolutionary change in the way cars are designed, developed and used: the autonomous driving.

Autonomous driving relies on advanced driver assistance systems (ADAS) [2], these technologies often are based on Artificial Intelligence (AI). Injecting AI in the

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 279–288, 2017.
DOI: 10.1007/978-3-319-67383-7_21

automotive industry will require deep changes in terms of skills, technologies, and development paradigms.

Today Deep Learning, a special type of AI technology, is considered the most suitable approach to face the complexity of autonomous driving applications. Actually, deep learning is considered by automotive companies as a viable technology not only for implementing autonomous driving, but also for improving other functional domains such as engine management [3] and vehicle cyber-security assurance [4].

Several standards are applicable in automotive. In particular, in the last years standards addressing software development have been released and applied, the most relevant and impacting are Automotive SPICE [10] and ISO 26262 [11] standards. The Automotive SPICE standard provides a framework at process-level that disciplines, at high level of abstraction, the software development activities and allows their capability assessment in matching pre-defined sets of process requirements. ISO 26262, titled "Road vehicles – Functional safety" and released in late 2011, target safety critical development and it scope is expectedly not limited to software engineering. Both standards, as far as the software is concerned, rely conceptually on the traditional development paradigm: the V-model. The V-model is far away from being applicable in the case of development of Artificial Neural Networks for AI applications. That represents an open issue for developers of AI-based automotive applications and devices and for car manufacturers too.

In this paper the authors address the topic of the applicability of the existing reference standards for automotive (in particular Automotive SPICE) to the development of AI-based applications (with special reference to the Deep Learning ones).

The paper is structured as follows: in Sect. 2 an overview of the Deep Learning is provided; in Sect. 3 an authors' defined development paradigm for Deep Learning called W-model is introduced and discussed. In Sect. 4 the applicability of Automotive SPICE to Deep Learning development is systematically discussed in order to point out the existing lacks and open issues. In Sect. 5 conclusions are finally provided.

2 Deep Learning Overview

The principal characteristic of artificial neural networks is the ability to improve their problem-solving capabilities through a "learning" process triggered by exemplary input [5]. This feature makes the use of artificial neural networks particularly suitable in scenarios in which there is no detailed, complete or predictable information about the problem as usually in automotive driving situations. Another relevant feature is their parallel structure, which benefits from the use of powerful hardware to obtain timely and thus usable computation results.

Deep learning, which is synonymous of deep neural network (DNN), is a specific kind of artificial neural network able to model complex non-linear relationships using multiple hidden layers of units between the input and output layer (Fig. 1 – right). Deep learning excels on finding patterns when input are massive analog data – this means not a few numbers in a tabular format, but instead images of pixel data or audio data. Until 2006 advances, DNNs were outperformed by shallow neural network that relied on feature engineering which is the embedding of the domain knowledge in the solution design.

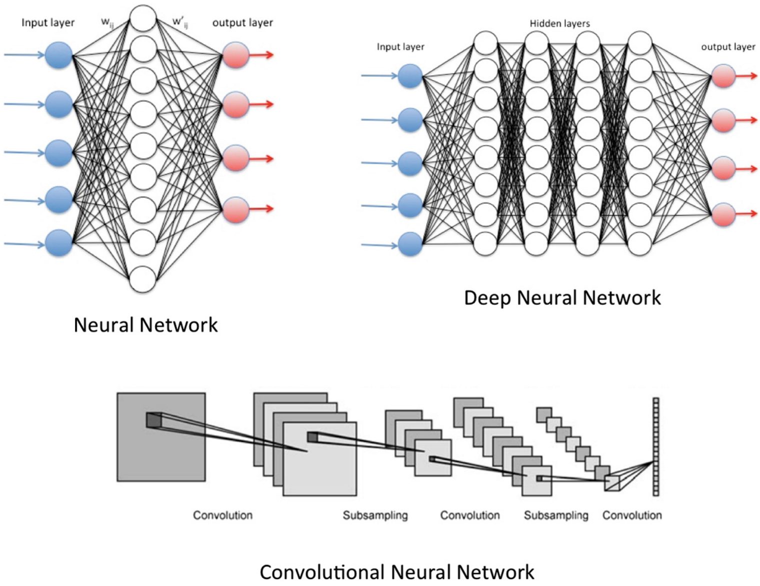

Fig. 1. Examples of neural networks.

The structure of a deep neural network is flexible and can be customized by selecting attributes such as the number of hidden layers, number of units (also called "nodes") per layer, number of connections per unit etc. These attributes, known as hyper-parameters, define the structure as well as the behavior of a deep learning-based system.

Summarizing, the fundamental characteristics of deep learning are:

- Input-output mapping through the learning process
- Nonlinearity - DNNs are composed of an interconnection of nonlinear computational elements (a.k.a. neurons or nodes)
- Adaptation capability - DNNs have a built-in adaptation ability to the changes in the environment
- Fault tolerance - due to DNN distributed nature, localized faults in hidden layers leads to a degradation of performance rather than a system failure.

Their processing capability is stored in the inter-unit connection weight obtained by a process of adaptation to a set of training patterns.

Convolutional neural networks (CNN), also known as ConvNets, are a type of deep neural network (Fig. 1 – bottom) conceived to manage data in form of arrays with some degree of spatial structure [6, 7].

They are designed to emulate the behavior of a visual cortex and perform very well on visual recognition tasks because the convolution operation (in shape of matrix products) itself is capable of capturing the features of images. Convolutional neural networks have special layers called convolutional layers and sampling layers that allow the encoding of the images properties [8].

Convolutional Neural Networks, for their characteristics such as input data segmentation and a high degree of parametrization (up to hundreds of thousands), are of special interest for visual applications in the automotive context such as object detection, vehicle detection, road marking detection and more.

2.1 Existing Solutions

Google is making remarkable and highly visible investments in the development of autonomous vehicles. Its prototypal self-driving vehicles embed deep learning based technology already able to detect pedestrians in various and challenging scenarios. Google deep learning systems have achieved outstanding performance, making the error rate for machine vision lower than the one of a human being (5% error rate is human benchmark). This achievement, also due to new hardware architectures using multiple Graphics Processing Units (GPU), is pushing the migration of features based on traditional image processing technology to deep learning-based solutions.

This is just the beginning and, even so, remarkable elements of artificial intelligence are already available in circulating vehicles. In the ADAS domain, Tesla is reported to feature onboard the implementation of a neural network functionality for vision, sonar and radar processing that runs on the powerful NVIDIA DPX2 processor in the driving control unit [13].

Several other suppliers are already active players. The Intelligent Surround View (ISV) system, by AdasWorks, is an example of an ADAS neural network-based implementation, which processes the environment around the car using the visual information coming from several cameras [14]. In addition, DENSO R&D labs together with other important companies R&D labs are actively researching in this direction.

In the infotainment domain, the 2015 BMW 7 Series is reported to be the first car to feature an innovative voice recognition solution based on deep learning technology that works also in absence of the wireless car connectivity [15].

Other automotive applications of deep learning, currently under development, range from engine fault diagnosis and emissions management to detection of vehicle network intrusion.

From an hardware perspective, the level of electronic support required to embed deep learning in high-performance and safety-related automotive applications is so demanding that companies are aggressively developing new generations of chips: an example is the Mobileye's upcoming cutting-edge EyeQ 5 proprietary chip. In regard of commercially available electronics components, Intel is positioning with the new Xeon Phi chip to compete in this market, which has been so far ruled by Nvidia Tegra chip.

3 The W Model for Deep Learning Software Development

Software construction (i.e. the mere activity of software coding) is relatively a simple job in the case of Deep Learning. Basically it consists in developing the code of the nodes belonging to the different layers the neural network is composed of. In fact, the sophisticated features implemented by the neural network are not the result of a coding activity performed by software engineers, more appropriately they can be considered as the result of the learning process of the network performed using a general purpose learning procedure. That represent a novelty with respect the software development practice in automotive today.

The software side of DNN development is a highly iterative activity composed by a stream of steps in an end-to-end fashion [9], as shown in Fig. 2.

Fig. 2. DNN development workflow.

Compared with the traditional approaches, deep learning development process needs the support of empirical design choices driven by heuristics. Development often start from well-known learning algorithms, which have been proven effective in comparable problems or domains, since the understanding of the result of learning process is difficult to be grasped and thus managed.

Expected DNN requirements also include performance demands, expressed in terms of statistical benchmarking of the DNN functional behavior (i.e. error rate), that are carefully targeted during the DNN validation phase.

Automotive software engineering, while welcomes innovation and outstanding functional performances, is strict in its request for a robust and predictable development cycle coming from the demand of compliance with standards as Automotive SPICE. The authors introduced in [1] a new lifecycle called W-model with the aim of providing a contribution in placing deep learning in a more controlled V-model perspective to address a lengthy list of challenges, such as requirements criteria for training, validation and test data sets, criteria for the training data pre-processing, management of very large sets of parameters and much more.

The introduction of a more structured conception of the deep learning lifecycle is instrumental to reach a controlled development approach that cannot be addressed by the mere functional benchmarking. However, deep learning intrinsically introduces its specific features (not completely fitting with the V-model) for software development.

The introduction of the W-model is essential because the central role played by data in this context (e.g. for DNN training and training validation). To support it we introduce the term "programming by example" to highlight the importance of data in developing systems based on deep learning technology.

The deep learning W-model is a framework lifecycle that conceptually integrates a V model for data development in the 'traditional' V-like perspective (Fig. 3).

The W model acknowledges that deep learning is driven by software development as well as data development. The design and the creation of training/validation/test data sets, together with their exploitation, are crucial development phases because the DNN's functional behavior is the combined result of its architectural structure and of its automatic adaptation through a training process.

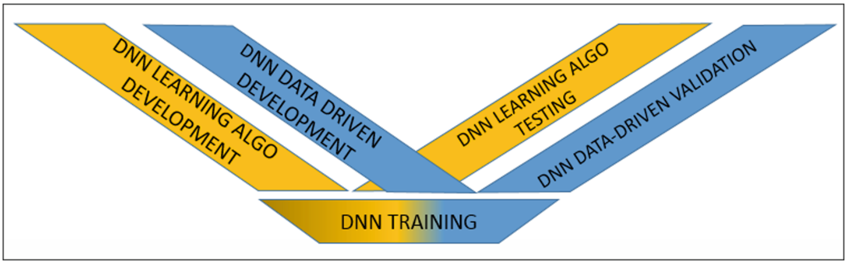

Fig. 3. The W-model for deep learning.

Deep learning moves away from feature engineering by definition and this element makes the W-model an appropriate and useful representation of such a sophisticated paradigm.

4 Deep Learning vs. Automotive SPICE 3.0

The software development process for on-board automotive ECUs is subject to proprietary OEMs norms as well as several international standards. Among them, the most relevant and influential standards for deep learning are Automotive SPICE [10] and ISO 26262 [11]. Needless to say that these standards are still far from addressing it with dedicated statements.

The Automotive SPICE standard - SPICE stands for Software Process Improvement and Capability dEtermination - provides a process framework that disciplines, at high level of abstraction, the software development activities and allows their capability assessment in matching pre-defined sets of numerous process requirements. ISO 26262 targets safety-related development and its scope expectedly includes system, hardware and software engineering. It is important to remark that the ISO 26262 standard already addresses configuration and calibration data, even though this aspect is an order of magnitude simpler and plainer than the development of DNN data sets.

Both standards, as far as the software is concerned, rely conceptually on the traditional development lifecycle: the V-model.

It is also very relevant for deep learning the ISO PAS "Safety of the Intended Functionality (SotIF)" [12] that is currently in advanced development stage. This ISO document addresses the fact that for some ADAS applications there can be safety violations with a system free from faults - for example a false-positive detection by a radar of an obstacle for the vehicle – because it is extremely problematic to develop systems able to address every possible scenarios.

In the following, we intend deepening the correlation between software engineering processes of Automotive SPICE and the W model for deep learning in order to contribute in the harmonization between deep learning development and the state-of-the-practice in automotive software development. To do that, Automotive SPICE processes related to software development are compared with the Deep Learning development phases with the aim of discussing the applicability of the Automotive SPICE process reference model in the case of Deep Learning as well as highlighting possible gaps and open issues between the two schemes.

The processes we consider are those related to software development that, in the Automotive SPICE standard, are identified by the prefix SWE:

SWE.1: Software Requirements Analysis process
SWE.2: Software Architectural Design
SWE.3: Software Detailed Design and Unit Construction
SWE.4: Software Unit Verification
SWE.5: Software Integration and Integration Test
SWE.6: Software Qualification Test

To additional details in terms of process purpose, base practices and work products, refer to [10]. According to the W-model discussed in Sect. 3, the development of software of Neural Networks (and in particular of Deep Neural Networks) can be seen as decomposed into two different branches:

1. the DNN Learning Algorithm Development that addresses the development of the code associated to the nodes belonging to the different layers the network is composed of;
2. the DNN Data Driven Development that addresses the data-driven developments consisting in the training of the network by data.

Figure 4 provides, in a graphical way, the description of DNN development phases divided into the two development branches (i.e. the DNN Learning Algorithm Development and the DNN Data Driven Development). It is to be noticed that the "Execution of DNN Training" and "DNN Validation" phases are common to both branches. In fact, their execution and the related effects affect both the learning algorithm and the training and validation data.

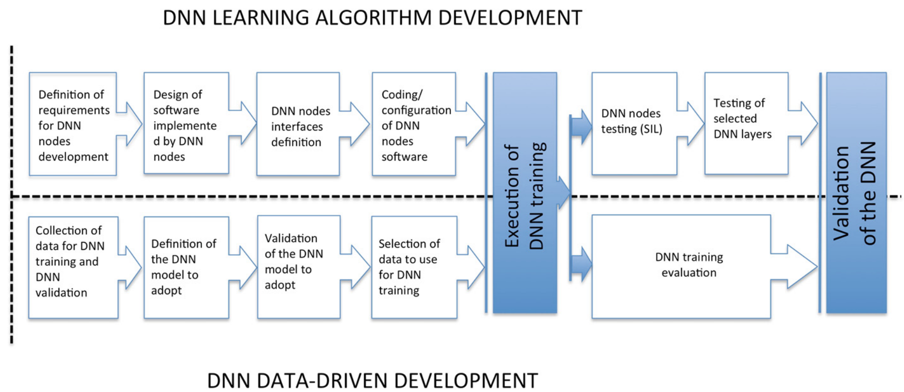

Fig. 4. DNN learning algorithm and DNN data-driven development phases.

In the following tables the singles phases the DNN Learning Algorithm Development and DNN Data Driven Development are composed of are mapped on the six SWE processes from Automotive SPICE.

Table 1. Mapping between automotive SPICE processes and DNN learning algorithm development with related addressability indication.

Automotive SPICE process	DNN learning algorithm development phase	Addressability
SWE.1	- Definition of requirements for DNN nodes development	Yes
SWE.2	- Selection or design of the algorithms implemented by DNN nodes	Yes
SWE.3	- Specification of DNN nodes Interfaces - Coding of nodes' software - Configuration of the DNN (i.e. setup of parameters/weights of the DNN) - Execution of DNN training	Yes Yes Yes No
SWE.4	- DNN nodes testing	Yes
SWE.5	- Testing of selected DNN layers	Yes
SWE.6	- Validation of DNN	Yes

In Table 1 the mapping between the SWE processes and the DNN learning algorithm development phases is provided along with a two-values column (Yes or No) indicating the applicability of the current contents of the SWE process (in terms of Base Practices and Work Products) to those phases. If the value on the column "Applicable" is "Yes" it means that the current version of the related process is applicable for DNN learning algorithm development without significant changes, if the value is "No" it means that the current version is not suitable to address the peculiarities of that phase.

In Table 2, the same is reproduced for the mapping between SWE process of Automotive SPICE and the Data Driven Development phases.

While it is possible the make a conceptual mapping between the W model and the SWE-class processes of Automotive SPICE (as shown above), the content, in terms of Base Practices and Work Products, of the current version Automotive SPICE (ver. 3.0) is far from being applicable in DNN development.

The Automotive SPICE model has been conceived taking into account the V-model. As a consequence of that the Base Practices and Work Products of the processes belonging to the SWE process group reflect the 'traditional' way to develop software.

To make Automotive SPICE applicable to projects developing Deep Learning applications, the current set of Base Practices and Work Products simply needs to be re-defined to make it meaningful and aligned with Deep Learning development.

While making the current version of Automotive SPICE applicable to the DNN Learning Algorithm Development can be considered a relatively light job, on the contrary, making it suitable for the DNN Data Driven Development needs a re-definition of the performance indicators (i.e. Base Practices and Work Products).

Table 2. Mapping between automotive SPICE processes and DNN data driven development with related addressability indication

Automotive SPICE process	DNN data driven development phase	Addressability
SWE.1	- Identification and storing of data to be used for the DNN training and DNN validation	No
SWE.2	- Selection or Design of the prediction model (i.e. definition of the DNN model to be adopted) - Validation of the DNN model to be adopted - Selection of data to be used for DNN training	No No No
SWE.3	- Execution of DNN training	No
SWE.4	- none	No
SWE.5	- Execution of DNN training - DNN training evaluation	No No
SWE.6	- Validation of DNN	Yes

5 Conclusions

The technology growth in computation capability and the availability of huge amount of data are pushing Artificial Intelligence applications, and in particular Deep Learning applications, to be massively adopted in automotive.

Automotive is a domain that in the last two decades witnessed impressive technological advancements accompanied by a progressive introduction of standards and norms that brought discipline and uniformity in development paradigms of software-intensive systems. Those paradigms have been, so far, based on the so-called V-model.

Automotive deep-learning-based applications and components are developed according to paradigms and models different that the V-model. That introduces new challenges in automotive in terms of applicability of the existing standards.

The penetration of Electronic Control Unit (ECU) hosting artificial intelligence is supposed to grow in a steady and substantial way according to endorsed market researches. The expected volumes are so important to reinforce the need to analyze the peculiarities and to integrate deep learning in the development lifecycle of automotive electronics systems.

This paper addressed the issue of the applicability of Automotive SPICE standard to deep-learning-based developments. The authors defined in a previous work the so-called W-model, a reference development model for deep learning able to represent that deep learning development is driven by software development as well as data development. A conceptual mapping between the W-model and Automotive SPICE is provided in this paper, with the aim of systematically highlight possible inconsistencies. The evidences of such a conceptual mapping show that Automotive SPICE can be considered generally applicable only for the part of deep learning development dealing with the algorithmic implementation of nodes. The part dealing with data driven development is far from being mapped to the Automotive SPICE process model.

From such a situation we can argue that, because of a large amount of deep-learning-based applications are expected to be part of vehicles in the next few years, a strict demand of new standards (or extensions of existing ones) exists. To respond to such a demand the automotive community should start to address this challenge in a similar way it has been able to face other technological and methodological challenges in the recent past.

References

1. Falcini, F., Lami, G., Costanza, A.M.: Deep learning in automotive software. IEEE Softw. **34**(3), 56–63 (2017). doi:10.1109/MS.2017.79
2. Levinson, J., et al.: Towards fully autonomous driving: systems and algorithms. In: Proceedings of 2011 IEEE Intelligent Vehicles Symposium (IV11) (2011). ieeexplore.ieee.org/document/5940562
3. Parlak, A., et al.: Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a diesel engine. Appl. Therm. Eng. **26**(8–9), 824–828 (2006)
4. Kang, M.-J., Kang, J.-W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS ONE, **11**(6), e0155781 (2016). journals.plos.org/plosone/article?id=10.1371/journal.pone.0155781
5. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. J. **61**, 85–117 (2015)
6. Haykin, S.: Neural Networks and Learning Machines. Prentice-Hall, New York (2009)
7. Credi, J.: Traffic sign classification with deep convolutional neural networks. Master's thesis, Department of Applied Mechanics, Chalmers University of Technology (2016)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
9. Miao, H., et al.: ModelHub: lifecycle management for deep learning. University of Maryland (2015). www.cs.umd.edu/class/spring2016/cmsc396h/downloads/modelhub.pdf
10. Automotive SPICE Process Assessment/Reference Model, ver. 3.0, Verband der Automobilindustrie (2015). www.automotivespice.com/leadmin/software-download/Automotive_SPICE_PAM_30.pdf
11. ISO 26262—Road Vehicles—Functional Safety—Part 1: Vocabulary, Int'l Standard Org. (2011)
12. ISO/AWI PAS 21448—Road Vehicles—Safety of the Intended Functionality, Int'l Standard Org. (2016)
13. Pressman, M.: Inside Nvidia's New Self-Driving Supercomputer Powering Tesla's Autopilot. CleanTechnica, 25 October 2016. cleantechnica.com/2016/10/25/inside-nvidias-new-self-driving-supercomputer-powering-teslas-autopilot
14. CEVA and AdasWorks to Demonstrate Free Space Detection for Autonomous Driving at AutoSens Conference 2016, CEVA, 15 September 2016. ceva-dsp.mediaroom.com/2016-09-15-CEVA-and-AdasWorks-to-Demonstrate-Free-Space-Detection-for-Autonomous-Driving-at-AutoSens-Conference-2016
15. De Ambroggi, L.: Artificial Intelligence Systems for Autonomous Driving on the Rise, IHS Says, IHS, 13 June 2016. technology.ihs.com/579746artficial-intelligence-systems-for-autonomous-drivingon-the-rise-ihs-says

A Proposed Approach to the Revision of IEC 80001-1 Following Annex SL

Silvana Togneri MacMahon[1(✉)], Todd Cooper[2], and Fergal McCaffery[1]

[1] Department of Visual and Human Centred Computing, Regulated Software Research Centre, Dundalk Institute of Technology and Lero, Dundalk, Co. Louth, Ireland
{Silvana.MacMahon, Fergal.McCaffery}@dkit.ie
[2] Trusted Solutions Foundry, San Diego, CA, USA
Todd@trustedsolutionsfoundry.com

Abstract. IEC 80001-1 was published in 2010 and is now undergoing revision. Feedback gathered on the adoption of the standard has revealed a number of barriers that have impacted its adoption. The standard provides requirements related to the roles, responsibilities and activities that need to be performed for the risk management of medical IT networks. One reported barrier is a lack of drivers to motivate Top Management to implement the standard. In addition, there is a lack of alignment between IT and biomedical engineering departments within hopitals. Finally, the IEC 80001-1 standard was considered to be too complicated and complex to implement. This paper presents the barriers identified in the feedback and presents an approach to the revision of the standard as a process based management system standard in accordance with ISO/IEC Directives Annex SL as a means to overcome these barriers.

Keywords: IEC 80001-1 · Annex SL · Management System Standard · Process assessment · BS EN 15224

1 Introduction

There is an increased focus on ensuring that a high standard of care is provided to the patient while reducing the cost of care. This focus is due to the recent downturn in the global economy. One potential approach to achieving this goal is through the use of interoperable medical devices [1–3]. Governments recognising this potential have provided incentives to promote the meaningful use of interoperable medical devices and Health Information Technology (HIT), such as Electronic Health Records (EHRs) [4–6]. The increased prevalence of chronic conditions such as diabetes, which has resulted in a move away from acute episodic care, has led to increased use of interoperable medical devices. The management of chronic disease requires the establishment of an ongoing relationship between the patient and their care team facilitated by carefully designed care processes and requiring the support of information technology [7–10]. The number of networked medical devices in use continues to increase as a result of this change, [11–13].

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 289–301, 2017.
DOI: 10.1007/978-3-319-67383-7_22

Benefits to patients identified through the use of networked medical devices include reducing the instances of adverse events improving patient safety, reducing the time spent by clinicians manually entering information, reducing redundant testing due to inaccessible information, improving patient care, reducing healthcare costs and ensuring comprehensive and secure management of health information [14, 15] resulting in medical IT networks becoming a critical, integral component of the medical system [16].

However, as medical devices increasingly interface with other equipment and hospital information systems the integration complexity of the systems is increased and this presents additional operational risks [13, 17–19]. Traditionally, when devices were placed onto a network, proprietary networks were used. Increasingly, medical devices are being designed to be placed onto the hospital's general IT network. There has been a move away from the use of proprietary networks as their use may limit the communication of the devices and therefore the potential benefits of connecting devices. This means that medical device manufacturers no longer exercise full control over the configuration of the network [20] with hospitals sourcing network components and devices from different manufacturers. This lack of control can lead to risks which result in unintended consequences outside the control of the medical device manufacturer as the placement of the device onto the hospital network creates a new system in which the device has not been validated [21]. These risks can result in the incorrect and degraded performance of the medical device [22, 23] compromising patient safety, effectiveness and the security of the IT network [24–27].

IEC 80001-1: *Application of risk management for IT-networks incorporating medical devices* [28] was published in 2010 to address the risks associated with the incorporation of a medical device into an IT network. This standard is now scheduled for revision. The revision of the standard will take into account feedback which has identified barriers to the adoption of the standard as well as the need to broaden the scope of the standard. This paper presents a proposed approach to the revision of the IEC standard and is structured as follows. Section 2 presents the results of the feedback gathered which identifies barriers to the adoption of the 2010 version of the IEC 80001-1 standard. Section 3 presents the proposed approach to the revision of the standard. Section 4 examines how the proposed approach addresses the identified barriers to adoption of the current version of the standard and, finally, Sect. 5 presents the conclusions of the work and outlines future work in this area.

2 Barriers to the Adoption of IEC 80001-1: 2010

Prior to commencing work on the revision of the standard, feedback was gathered by the developers of the standard, International Electrotechnical Commission (IEC) Sub-Committee (SC) 62A – International Organization for Standardization (ISO) Technical Committee (TC) 215 Joint Working Group 7 (JWG7), to identify any barriers to its adoption in its current form. This feedback was gathered for use in identifying an approach to the revision of the standard. The feedback was gathered through three case studies. These case studies examined the lessons learned from a number of projects which were conducted in which a Healthcare Delivery Organisation (HDO) attempted to implement IEC 80001-1. The projects were carried out in HDOs of varying sizes and in different geographical locations. The first case study was carried out in a large HDO based in the US who performed a pilot implementation of IEC 80001-1. The second was performed in a Cancer Therapy unit based in Austria targeting a full implementation of IEC 80001-1. The final case study was based on the experiences of a Health Service in Australia and it's experience in the implementation of IEC 80001-1 across a number of HDOs. Based on these case studies, the report on the feedback was compiled by JWG7 and identified 3 barriers to the adoption of the standards as follows:

- Lack of drivers to motivate Top Management to implement the IEC 80001-1 standard
- HDO Organizational challenges: Information Technology (IT) and Biomedical Engineering (BME) departments are not aligned
- The IEC 80001-1 standard is too complicated and complex to implement

Each of these barriers to adoption of the standard is discussed in the remainder of this section.

2.1 Lack of Drivers to Motivate Top Management

An issue which was identified during the case studies was that Top Management do not see the return on investment of implementing IEC 80001-1. This can cause issues in the adoption of the IEC 80001-1 standard as Top Management may be reluctant to provide the support and resources which are required in order to implement the standard. Comments from task group participants revealed that while participants felt that an argument can be made to say that implementing the standard increases patient safety, participants also felt that this benefit has not as yet been quantified and so may be thought of as too abstract by Top Management. It was also reported that there is a correlation between a hospitals experience and their desire to perform risk management activities required under IEC 80001-1. Hospitals that have experienced incidents such as lost patient records or viruses are more likely to implement the requirements of the standard. This can lead to "fragmented motivations" for implementing the standard.

It was also reported that IT management lack knowledge of basic risk management concepts such as safety and reliability engineering and Failure Mode Cause and Effect Analysis (FMCEA). It was stated that "this resonates as the single largest impediment to 80001 adoption and needs clear and concise focus in the revision". This issue is also discussed in the context of HDO Organisational Challenges section of this paper. Additional perceived barriers to the implementation of the standard were the cost, complexity, lack of resources and/or skillsets. These barriers are directly related to the lack of information concerning the return on investment on implementation of the standard. This lack of information has recently been partially addressed by the publication of a white paper by the Association for the Advancement of Medical Instrumentation (AAMI) which provides quantitative information regarding the return on investment of implementing IEC 80001-1 [29].

2.2 HDO Organisational Challenges

In addition to the challenges identified regarding Top Management support for adoption of the standard, an additional barrier was identified. There has been a move within hospitals to promote greater levels of communication between the clinical departments, which include clinicians, Management and BME, and the IT departments, which includes network administrators and network engineers. However, these departments still tend to operate in silos often leading to communication breakdowns between the two departments. The feedback indicated that, in general, IT do not understand clinical workflows or that network connectivity has become a crucial element of patient care. It is also reported that BMEs do not understand complex networking concepts. They “do not speak the same language”.

IEC 80001-1: 2010 references a risk management standard for medical devices - ISO 14971 [30]. It should be noted that based on this feedback, the revised standard will now also reference ISO 31000 [31], a generic risk management standard. IT departments while familiar with the definition of risk within ISO 31000 are not familiar with the requirements of ISO 14971. Expanding the reference to include ISO 31000 will provide understanding of how to integrate the requirements of IEC 80001-1 within a HDOs larger risk management framework which includes many more objectives than safety, effectiveness and security, the key properties defined in IEC 80001-1.

2.3 IEC 80001-1: Too Complex and Complicated to Implement

The IEC 80001-1 standard was reported to be too complex and complicated to implement. Organisations reported that, the standard was too abstract and did not provide a means to tailor it to their needs, also it lacked guidance on how a stepwise approach may be taken to the implementation of the standard. While there is a technical report, ISO TR 80001-2-7 [32], which provides guidance on how to assess against the requirements of the standard and provides information on tailoring the assessment to a specific HDO context, it was reported that the top-level standard, IEC 80001-1, is dependent on the associated technical reports to provide guidance on various aspect of implementation of the standard. However, often the technical reports are either not available due to lack of awareness or do not provide sufficient guidance on implementation.

2.4 Conclusions from the Case Studies/Lessons Learned Report

In order to address the barriers identified during the case studies, it was agreed that a process approach similar to that taken in ISO/TR 80001-2-7 should be taken in the revision of the standard. While this would provide an approach it does not fully address the barriers to adoption identified during the lessons learned report. This research has focused of developing a proposed approach to the revision of the IEC 80001-1 as a process standard while addressing the identified barriers. This proposed approach is discussed in the remainder of this paper.

3 Proposed Approach for the Revision of IEC 80001-1

In determining the proposed approach for the revision of the standard, a review of the lessons learned was conducted to ensure that the approach to the revision would address these lessons and identified barriers to adoption. In addition, during the revision of the standard the scope of the standards is to be broadened. IEC 80001-1 focused on risk management of medical IT networks which were defined as an IT network that contained at least one medical device. However, this scope is to be broadened to include health software and health IT systems. This is consistent with the approach taken in IEC/CD 62304 [33] and IEC 82304: 2016 [34]. This revised scope was considered in determining the approach to the revision of the standard.

3.1 Determining the Approach to the Revision

In determining the approach to the revision of the standard, a number of standards and ISO directives were examined to assess their ability to address the lessons learned. These standards are examined in the remainder of this section prior to presenting the proposed approach to the revision of the standard.

BS EN 15224:2016
BS EN 15224: 2016 [35] is a sector specific quality management system standard for healthcare. The standard incorporates requirements from "EN ISO 9001:2015" with additional requirements, specifications and interpretations for healthcare. This standard is based on Annex SL of the ISO/IEC Directives, Part 1 – Consolidation ISO Supplement – Procedures specific to ISO [36].

Annex SL
Annex SL of the ISO Directives outlines requirements for the development of Management System Standards. The directive defines a management system standard as: "a set of interrelated or interacting elements of an organisation to establish policies and objectives and processes". Section S.9 of the Annex outlines the High level structure, identical core text and common terms and core definitions for use in Management Systems Standards. This high level structure is shown in Table 1.

Table 1. Annex SL high level structure

Clause	Title
Clause 1	Scope
Clause 2	Normative References
Clause 3	Terms and Definitions
Clause 4	Context of the organisation
Clause 5	Leadership
Clause 6	Planning
Clause 7	Support
Clause 8	Operation
Clause 9	Performance evaluation
Clause 10	Improvement

ISO/TR 80001-2-7: Guidance for healthcare delivery organizations (HDOs) on how to self-assess their conformance with IEC 80001-1

This technical report provides guidance to HDOs on how to assess conformance with the requirements of IEC 80001-1. ISO/TR 80001-2-7 uses a process approach and outlines the requirements of IEC 80001-1 in the form of a Process Reference Model (PRM), Process Assessment Model (PAM) and assessment method which can be used by HDOs to assess the capability of their risk management processes in relation to medical IT networks. The PRM and PAM within the technical report were developed in compliance with the requirements of ISO/IEC 15504-2:2003 Software engineering — Process assessment — Part 2: Performing an assessment [37, 38]. This standard outlines the requirements for the development of PRMs and PAMs.

The PRM and PAM were developed using the TIPA transformation process [39] which is a goal oriented requirements engineering technique which allows a set of requirements to be transformed into a PRM and PAM which is compliant with the requirements of ISO/IEC 15504-2. The transformation process was used firstly, as it had been used in the development of similar PRM and PAMs for service management standards [40]. These service management standards were identified as being similar to the IEC 80001-1 standard [41]. Secondly, this approach was used for developing a PRM and PAM that are compliant with the requirements of ISO/IEC 15504-2 allows for an assessment to be performed regardless of the regulatory requirements of the geographical location in which the HDO provides care and also allows for the assessment to be tailored to take into account the context of the specific HDO in which the assessment is being conducted.

3.2 The Proposed Approach to the Revision of IEC 80001-1

Having reviewed the standards above the following approach to the revision of the standard has been proposed. It is proposed that the standard should be revised in the form of a management system standard in accordance with the requirements of Annex SL.

IEC 80001-1 as a Management System Standard

In order to determine if this approach would be possible, a high level mapping of the 14 processes from ISO/TR 80001-2-7 was performed against the structure for management system standards described in Annex SL (Table 1). The result of this mapping is shown in Table 2. It should be noted that as clauses 1 to 3 address Scope, Normative References, and Terms and Definitions respectively they have not been included in the mapping. All 14 processes within ISO/TR 80001-2-7 have been mapped to clauses 5 through 9 of Annex SL. As Clause 4 of annex SL addresses the context of the Organisation, it is expected that this section of the revised standard would provide guidance in terms of the context in which the HDO provides care. This section will incorporate wording from or reference to IEC/TR 80001-2-4 which provides guidance on implementing the requirements of IEC 80001-1 in large and small responsible organisations. This section will also provide information in relation to how a stepwise approach may be used in order to implement the requirements of the standard. The stepwise approach would be facilitated by the development of an Maturity Model

Table 2. Mapping of ISO/TR 80001-2-7 processes to Annex SL high level structure

Clause	Title	Notes
Clause 4	Context of the organisation	Advice on understanding the context and tailoring
		Maturity model – Stepwise approach
Clause 5	Leadership	**Organizational risk management process**
Clause 6	Planning	**Medical IT-Network risk management process**
		Medical IT-Network planning process
Clause 7	Support	**Medical IT-Network risk management process**
		Risk management policy process
		Medical IT-Network documentation process
		Responsibility agreements process
Clause 8	Operation	**Risk analysis and evaluation**
		Risk control process
		Residual risk process
		Change release and configuration
		Management process
		Decision on how to apply risk management
		Go-Live
Clause 9	Performance evaluation	**Monitoring process**
		Event management process
Clause 10	Improvement	The organisation shall continuously improve ….
		Refer back to MM and stepwise approach

(MM) that would allow HDOs to take a stepwise approach to the implementation of the standard. Clause 10 of the revised standard would provide further guidance on the implementation of a stepwise approach to the implementation of the standard and would address this specifically in the context of improvement and movement to the next level of the maturity model in terms of implementing the standard.

In addition to structuring the revised standard in a manner that is consistent with the structure of annex SL, the proposed approach will also incorporate the development of a PRM, PAM, documented assessment process and MM. The development of these models would be facilitated through the use of the TIPA Transformation Process for Management System Standards. This transformation process is discussed in the following section.

IEC 80001-1 as a Process Standard

While it is proposed that the standard will be developed as a Management System Standard, feedback gathered has also identified the value of adopting a process approach in the revision of the standard similar to the approach adopted in ISO/TR 80001-2-7. As previously discussed, the TIPA transformation process was used in the development of ISO/TR 80001-2-7 to ensure that the requirements of IEC 80001-1 could be transformed into a PRM and PAM that were compliant with the requirements of ISO/IEC 15504-2 (and ISO/IEC TR 24774 [42]). The developers of the TIPA transformation process have shown that the TIPA transformation process can be used in the development of PRMs and PAMs for Management System Standards [43]. The transformation process allows for compliance with ISO/IEC 33004:2015 [44]. ISO/IEC 33004:2015 sets out the requirements for process reference models, process assessment models, and maturity models and replaces ISO/IEC 15504-2 which has now been withdrawn.

Summary of the Proposed Approach to the Revision of the Standard
In order to revise the IEC 80001-1:2010 standard according to the proposed approach a number of steps are needed as follows:

- Firstly, using ISO/TR 80001-2-7 as a baseline (which contains the requirements of IEC 80001-1:2010 in the form of a ISO/IEC 15504-2 compliant PRM and PAM) the requirements expressed in the TR are reviewed in the context of the extended scope of IEC 80001-1.
- Additional requirements are incorporated into the draft revised standard to take account of the revised scope as required.
- All other existing technical reports aligned with IEC 80001-1 (IEC 80001-2-X) are reviewed for inclusion in the revised draft standard.
- Additional requirements form the technical reports are incorporated into the draft revised standard as appropriate. In some cases, it may be appropriate to make reference to the technical report rather than incorporating the requirements into the draft revised standard.
- Once all requirements have been identified for inclusion in the draft revised standard, the requirements should be structured according to the requirements of Annex SL, initially according to the high level mapping of ISO/TR 80001-2-7 processes to Annex SL.
- Using the TIPA transformation Process for Management System Standards a ISO/IEC 33004 compliant PRM, PAM and MM for the revised draft standard are developed.

The proposed approach to the revision of the IEC 80001-1 standard was presented to ISO TC215 JWG7 at a recent meeting in Hangzhou, China. While no concerns were raised regarding the proposed approach, the approach is to be sent to JWG7 members for feedback and for ballot.

4 How the Proposed Approach Addresses the Identified Barriers to Adoption

Section 2 outlined the barriers to adoption of the IEC 80001-1:2010 standard. In summary, the barriers identified were as follows:

- Lack of drivers to motivate Top Management to implement the IEC 80001-1 standard;
- HDO Organizational challenges: Information Technology (IT) and Biomedical Engineering (BME) departments are not aligned;
- The IEC 80001-1 standard is too complicated and complex to implement

This section reviews each of the identified barriers and examines how the proposed approach addresses each of the barriers.

4.1 Lack of Drivers to Motivate Top Management

Adoption of IEC 80001-1 requires sponsorship by Top Management by allocation of Budgets and resources to support the implementation of the standard. The proposed approach is revising the standard as a management system standard following the structure of Annex SL. Other management system standards such as ISO 9001:2015 [45] and BS EN 15224:2016 allow for certification against the requirements of the standard. Standards development organisations have published statistics regarding the return on investment of implementing standards such as ISO 9001 [46]. By revising IEC 80001-1 as a management system standard, a similar approach may be taken to determine the return on investment of implementing IEC 80001. This will involve leveraging Top Managements familiarity with the return on investment of implementing other management system standards such as ISO 9001 and may allow a path to certification against IEC 80001-1 in the future. This approach would also facilitate the integration of the requirements of the revised IEC 80001-1 standard with existing ISO 9001 processes (if previously implemented).

4.2 HDO Organisational Challenges

The second barrier to adoption which was identified was that IT and BME departments often operate in silos. IT do not understand clinical workflows and BME do not understand complex networking concepts. This issue was also identified during pilot implementations of ISO TR 80001-2-7 [47]. Using a structure based on Annex SL and ISO 9001 may aid in providing a common language between BME and IT. By basing the revision of the standard on a structure that both BME and IT may be familiar with, through implementation of BS EN 15224:2016 and ISO 9001 respectively, this may allow BMEs to discuss clinical aspects of networked medical devices in a way that is more understandable to IT and vice versa. By incorporating requirements from the technical reports into the revision of the standard, this will ensure that visibility of technical reports is provided. This will provide guidance on the implementation of the requirements and ensure that the standard is not “high level”, a criticism which is sometimes made regarding BS EN 15224:2016.

4.3 IEC 80001-1: Too Complex and Complicated to Implement

Feedback also revealed that the IEC 80001-1:2010 standard was felt to be both too complex and complicated to implement. Another barrier to adoption which was identified was the lack of a stepwise approach to implementation of the standard. These barriers are addressed in the proposed approach. Firstly, Annex SL simplifies the overall structure of the standard. This is illustrated in Table 2 which shows how 14 processes from ISO/TR 80001-2-7 can be mapped to the five clauses of Annex SL. Secondly, Annex SL allows information to be provided in relation to understanding the organisational context of the HDO and to provide guidance as to how this should be considered in the implementation of the standard. Thirdly, using the TIPA transformation process for management system standards allows for the development of a PRM, PAM and MM for the revised IEC 80001-1 standard. The PRM and PAM can be used to facilitate an assessment of the capability of a HDOs risk management process against the requirements of the revised standard while the OOM can facilitate a stepwise approach to the implementation of those requirements.

5 Conclusions and Future Work

This paper outlines a proposed approach to the revision of the IEC 80001-1:2010 standard. The proposed approach focuses on the revision of IEC 80001-1 as a Management System Standard following the structure as defined in Annex SL of the ISO Directives. In addition, the proposed approach will allow for the revision of IEC 80001-1 as a process standard by using the TIPA transformation process for Management System Standards to develop an ISO/IEC 33004 compliant PRM, PAM and MM which will allow for an assessment of capability of risk management processes related to health software and health IT systems to be performed. The development of the MM will allow for a stepwise approach to the implementation of the requirements of the standard which takes into account the context of the HDO in which care is being provided.

This paper also examined the barriers to adoption of the IEC 80001-1 standard and examined how the proposed approach to the revision of the standard addresses these barriers. The proposed approach may improve Top Management sponsorship of the implementation of the standard and increase management willingness to allocate the necessary budgets and resources to allow for implementation of the standard. Top Management understand the return on investment of implementing ISO 9001, which also follows Annex SL structure. Structuring the revised standard according Annex SL may allow for similar measures of return on investment to be developed. Using the Annex SL structure may address issues around the use of different language in the discussion of risk allowing for greater communication around the area by leveraging an understanding of and integration with existing standards. Revising the standard as a management system standard will simplify the structure of the standard and make implementation of the standard less complex. In addition, through inclusion of text from or reference to the associated technical reports will provide additional guidance on the implementation of the standard. Implementation will also be simplified by using a stepwise approach to implementation as defined in the proposed MM.

Following feedback from JWG7, the suitability of this approach will be determined and work will commence on the revision of the standard in line with the agreed approach. The revision will focus on addressing the extended scope of the standard and the identified barriers to adoption. All research outputs will be validated through the standard community within JWG7 and also through pilot implementation within a HDO.

Acknowledgments. This research is supported by Science Foundation Ireland through Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 13/RC/2094.

References

1. West Health Institute: The Value of Medical Device Interoperability - Improving patient care with more than $30 billion in annual health care savings (2013)
2. Hamilton, A., Nau, R., Burke, R., Weinstein, S., Dlatt, C.K.B., Fiore, S., Conyers, J.L.: Summary of the August 2011 Symposium on the Role and Future of Health Information Technology in an Era of Health Care Transformation. The George Washington University (2011)
3. Lee, I., Pappas, G.J., Cleaveland, R., Hatcliff, J., Krogh, B.H., Lee, P., Rubin, H., Sha, L.: High-confidence medical device software and systems. Comput. (Long. Beach. Calif.) **39**, 33–38 (2006)
4. Milenkovich, N.: OCR issues new HITECH regulations. http://drugtopics.modernmedicine.com/drug-topics/news/drug-topics/health-system-news/ocr-issues-new-hitech-regulations
5. Centers for Medicare & Medicaid Services: 42 CFR Parts 412, 413, 422 et al. Medicare and Medicaid Programs; Electronic Health Record Incentive Program; Final Rule (2010). http://www.gpo.gov/fdsys/pkg/FR-2010-07-28/pdf/2010-17207.pdf
6. Centers for Medicare & Medicaid Services: EHR Incentive Programs. http://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/index.html?redirect=/ehrincentiveprograms
7. Institute of Medicine: Crossing the Quality Chasm: A New Health System for the 21st Century (2001). https://download.nap.edu/catalog.php?record_id=10027
8. Wagner, E.H.: The role of patient care teams in chronic disease management. BMJ Br. Med. J. **320**, 569 (2000)
9. Wagner, E.H., Austin, B.T., Davis, C., Hindmarsh, M., Schaefer, J., Bonomi, A.: Improving chronic illness care: translating evidence into action. Health Aff. **20**, 64–78 (2001)
10. Hoffman, C., Rice, D.: Chronic Care in America: A 21st Century Challenge. Robert Wood Johnson Found, Princeton (1996)
11. Comstock, J.: 14M networked medical devices to ship by 2018. http://mobihealthnews.com/28295/14m-networked-medical-devices-to-ship-by-2018/
12. Agency for Healthcare Research and Quality (AHRQ): Health IT for Improved Chronic Disease Management (2013). http://healthit.ahrq.gov/ahrq-funded-projects/emerging-lessons/health-it-improved-chronic-disease-management
13. Castañeda, M.: Connecting devices and data on the healthcare network. Biomed. Instrum. Technol. **44**, 18–25 (2010)
14. Whitehead, S.F., Goldman, J.M.: Getting connected for patient safety how medical device "Plug-and-Play" interoperability can make a difference. Patient Saf. Qual. Healthc. 1–5 (2008)
15. Venkatasubramanian, K.K., Gupta, S.K.S., Jetley, R.P., Jones, P.L.: Interoperable Medical Devices - Communication Security Issues. IEEE Pulse, 2 September/October 2010
16. Hampton, R., Schrenker, R.: What does IEC 80001-1 mean to you? (2011). http://www.24x7mag.com/2011/01/what-does-iec-80001-1-mean-to-you/
17. Rakitin, S.R.: Networked Medical Devices: Essential Collaboration for Improved Safety. AAMI.org. (2009)
18. Loughlin, S., Williams, J.S.: The top 10 medical device challenges. Biomed. Instrum. Technol. **45**, 98–104 (2011)
19. Mehta, T., Mah, C.: Auto-provisioning of biomedical devices on a converged IP network. Biomed. Instrum. Technol. **43**, 463–467 (2009)
20. Gee, T.: Medical Device Networks Trouble Industry. http://medicalconnectivity.com/2008/12/18/medical-device-networks-trouble-industry/

21. Eagles, S.: An Introduction to IEC 80001: Aiming for Patient Safety in the Networked Healthcare Environment. In: IT Horizons, vol. 2008 (2008)
22. National Cybersecurity and Communications Integration Center: Attack Surface: Healthcare and Public Health Sector (2012)
23. Talbot, D.: Computer Viruses Are “Rampant” on Medical Devices in Hospitals (2012). http://www.technologyreview.com/news/429616/computer-viruses-are-rampant-on-medical-devices-in-hospitals/
24. Graham, J., Dizikes, C.: Baby’s death spotlights safety risks linked to computerized systems (2011). http://articles.chicagotribune.com/2011-06-27/news/ct-met-technology-errors-20110627_1_electronic-medical-records-physicians-systems
25. Shuren, J.: Health Information Technology (HIT) Policy Committee Adoption/Certification Workgroup - Testimony of Jeffrey Shuren, Director of FDA’s Centre for Devices and Radiological Health (2010). http://www.cchfreedom.org/pr/Health, IT Deaths - FDA jeffrey Shuren.pdf
26. Eagles, S.: IEC 80001: An Introduction. 80001-1 Experts (2012)
27. Cooper, T., David, Y., Eagles, S.: Getting Started with IEC 80001: Essential Information for Healthcare Providers Managing Medical IT-Networks. AAMI (2011)
28. IEC: IEC 80001-1 - Application of Risk Management for IT-Networks incorporating Medical Devices - Part 1: Roles, responsibilities and activities (2010)
29. Association for the Advancement of Medical Instrumentation: Health IT risk management, Arlington, Virginia (2017)
30. ISO: ISO 14971:2007 - Medical Devices - Application of Risk to Medical Devices (2007)
31. ISO: ISO 31000:2009 Risk management – Principles and guidelines (2009)
32. ISO: ISO/TR 80001-2-7: 2015 - Application of risk management for IT-networks incorporating medical devices – Application guidance – Part 2-7: Guidance for healthcare delivery organizations (HDOs) on how to self-assess their conformance with IEC 80001-1 (2015)
33. IEC: IEC/CD 62304 Health software – Software life cycle processes. https://www.iso.org/standard/71604.html?browse=tc
34. IEC: IEC 82304-1:2016 Health software – Part 1: General requirements for product safety (2016)
35. British Standards Institute: BS EN 15224:2016 Quality management systems. EN ISO 9001:2015 for healthcare (2016)
36. ISO/IEC: ISO/IEC Directives, Part 1 Consolidated ISO Supplement — Procedures specific to ISO - Annex SL (2015)
37. ISO/IEC: ISO/IEC 15504-2:2003 - Software engineering — Process assessment — Part 2: Performing an assessment (2003)
38. MacMahon, S.T., McCaffery, F., Keenan, F.: Transforming Requirements of IEC 80001-1 into an ISO/IEC 15504-2 Compliant Process Reference Model and Process Assessment Model (2013)
39. Barafort, B., Betry, V., Cortina, S., Picard, M., St Jean, M., Renault, A., Valdés, O., Tudor, P.R.C.H.: ITSM Process Assessment Supporting ITIL: Using TIPA to Assess and Improve your Processes with ISO 15504 and Prepare for ISO 20000 Certification. Van Haren, Zaltbommel (2009)
40. Barafort, B., Renault, A., Picard, M., Cortina, S.: A transformation process for building PRMs and PAMs based on a collection of requirements – Example with ISO/IEC 20000 (2008)
41. MacMahon, S.T., McCaffery, F., Eagles, S., Keenan, F., Lepmets, M., Renault, A.: Development of a Process Assessment Model for assessing Medical IT Networks against IEC 80001-1 (2012)

42. ISO/IEC: ISO/IEC TR 24774:2010 - Systems and software engineering — Life cycle management — Guidelines for process description (2010)
43. Cortina, S., Mayer, N., Renault, A., Barafort, B.: Towards a process assessment model for management system standards. Commun. Comput. Inf. Sci. **477**, 36–47 (2014)
44. ISO/IEC: ISO/IEC 33004:2015 Information technology – Process assessment – Requirements for process reference, process assessment and maturity models (2015)
45. ISO: ISO 9001: 2015 Quality management systems – Requirements (2015)
46. British Standards Institute: BSI's ROI tool - Calculate your Return On Investment with ISO 9001. http://roi.bsigroup.com/
47. Hegarty, F.J., MacMahon, S.T., Byrne, P., McCaffery, F.: Assessing a hospital's medical IT network risk management practice with 80001-1. Biomed. Instrum. Technol. **48**, 64–71 (2014)

SPI in Various Settings

Enterprise SPICE Extension for Smart Specialization Based Regional Innovation Strategy

Michael Boronowsky[1], Ieva Mitasiunaite-Besson[2], Antanas Mitasiunas[2](✉), David Wewetzer[1], and Tanja Woronowicz[1]

[1] TZI, Bremen University, Am Fallturm 1, 28359 Bremen, Germany
{mb, wewetzer, worono}@tzi.de
[2] Vilnius University, Universiteto Str. 24, 01513 Vilnius, Lithuania
{ieva.mitasiunaite-besson, antanas.mitasiunas}@mif.vu.lt

Abstract. Process capability modeling became a tool for the systematization and codification of knowledge for process oriented activities in various areas. Enterprise SPICE defines a domain independent integrated model for enterprise-wide assessment and continuous process improvement. This paper presents the use of a SPICE conformant application dependent process modeling to support a smart specialization based regional innovation strategy process. Smart specialization is the main approach for the development and implementation of innovation strategies to improve of European regions within the programming period 2014–2020 driven by EU structural funds. The work presented in this paper provides the details of the regional innovation strategy process capability assessment model that is designed as an extension of the Enterprise SPICE Model.

Keywords: Smart specialization · Entrepreneurial discovery · Process capability model · Innovation strategy

1 Introduction

Smart specialization is the main approach for the development and implementation of innovation strategies to develop European regional innovation systems within the programming period 2014–2020 driven by EU structural funds. As the EU's new policy instrument, regional smart specialization strategies (RIS3) is to be considered a systematic process and its development the prerequisite of structural funds allocation by the Commission for regional development. The concept of smart specialization acts as knowledge base in defining and implementing regional RIS3 [10]. In this context, there are hundreds of regions all across the European Community that have necessity to establish their own smart specialization strategy. The idea of regional smart specialization strategy, originated by Rodrik [6], was elaborated by Foray [7, 11] and other authors [9, 10, 12] and is supported by European Commission [8].

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 305–321, 2017.
DOI: 10.1007/978-3-319-67383-7_23

In this paper, a regional smart specialization strategy with focus on place-based economic transformation is understood as a process for strategy creation, implementation and monitoring. The strategy creation and implementation, strategic change management and leadership is considered being crucial success factors.

Once a RIS3 is a process oriented activity, a methodology for process oriented activity modeling, assessment and improvement can be applied. The *process capability* modeling became a tool for systematization and codification of knowledge for process oriented activities. The introduced concept of *process capability* enables to assess the predictability of activity and to improve the quality of its results.

Enterprise SPICE (ISO/IEC 33071) [2, 3] defines a domain independent integrated model for enterprise-wide assessment and continuous improvement of process capabilities. Particular application domains contain application specific knowledge that cannot be covered in width and depth needed by domain independent process model. To address improvement issues of RIS3, this paper presents employment of SPICE conformant application dependent process modeling to regional smart specialization strategy creation and implementation by development of RIS3 process assessment model designed as Enterprise SPICE extension. The approach discussed in this paper was introduced in [1] as a methodical approach of the INTERREG project P2L2. The P2L2 project applies interregional policy learning and exchange of experiences on aspects influencing the regional innovation ecosystems in the field of advanced materials. The activities related to the definition, implementation and evaluation of the RIS3 and smart specialization strategies are described in terms of an ISO/IEC 33004 conformant Process Reference and Assessment Model (PRM/PAM). However, in [1] only a general introduction to the developed approach was given. The work presented here in this paper is providing the technical details of the methodology.

2 Design Approach

The methodology for SPICE conformant application domain dependent process capability modeling based on the ISO/IEC 33002 [13] and ISO/IEC 33020 [15] capability framework and Enterprise SPICE domain independent external process model has been proposed in [4] for the construction of the PRM and PAM innoSPICE [5].

The purpose of the *regional smart specialization strategy process category* introduced here is to reflect directly the body of knowledge in terms of essential processes and base practices of application satisfying the requirements of ISO/IEC 33002 [13] and ISO/IEC 33004 [14] for process reference models (PRM) and process assessment models (PAM). The official EU regional smart specialization strategy resources [8] and the extended background research material [6, 7, 9–12] had been taken as the main sources for the domain specific body of knowledge.

3 Smart Specialization Strategy as Modern Regional Innovation System

The goal of this work presented in this paper is to create a RIS3 process assessment model, preceded by creation of RIS3 process reference model. RIS3 PRM and PAM are ideal process models. RIS3 PAM as ideal process model can be used to indicate the direction where real RIS3 process model should be improved after assessment.

The authors of this work are providing a methodology and tools for a RIS3 process assessment and improvement and thus filling an important gap in the RIS3 methodology. The success of this approach depends on the creation and validation of a suitable RIS3 process capability assessment model. The first version of this RIS3 process reference model (PRM) and process assessment model (PAM) is provided in this paper.

The RIS3 strategy design and implementation process is an institutional process. According to the Guide to RIS3 [8], the RIS3 life-cycle consist of four sequential phases: (1) analysis of context and potential, (2) production of shared vision, (3) selection of priorities, (4) establishment of policy mix and three parallel actions: (a) governance, (b) monitoring and (c) evaluation. The owners of these processes are public administration institutions that are responsible to create conditions for involvement and collaboration of entrepreneurial agents and society.

The Entrepreneurial Discovery Process is completely different from the RIS3 process. It is external to RIS3 process and they are performed by different entities. An adapted version of the innoSPICE *knowledge and technology transfer driver process category* [5] could be suitable as an EDP process assessment model. Entrepreneurial agents represent enterprises, public and private research organizations and individual entrepreneurs. Entrepreneurial agents provide the key input for the RIS3 definition and are key contributors to its implementation.

The picture below provides the architecture and the context of a smart specialization based regional innovation strategy (Fig. 1).

Public administration institutions create conditions and facilitate entrepreneurial discoveries, produce shared vision for regional/national transformation; select priorities for regional/national transformation, establish policy instruments for strategy implementation, including resources allocation for transformation actions, monitor, evaluate and govern whole process of RIS3 definition and implementation including their iterations. The purpose of society involvement is to respect societal values for regional transformation. Smart specialization means a smart way to find solutions that fit best to a regional improvement based on particularities of a region and first of all on the state of the current regional development. Smart specialization means RIS3 bottom-up design as opposite to traditional top-down approaches where public authorities on their own behalf decide on priorities for regional transformation. Smart specialization is based on the knowledge generated by a regional entrepreneurial discovery process.

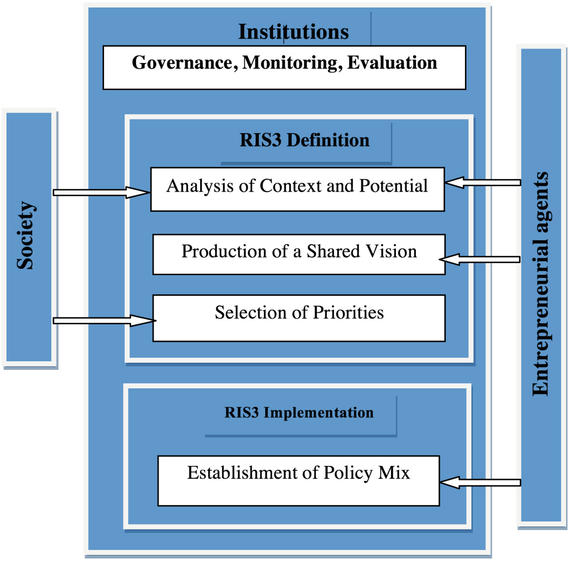

Fig. 1. RIS3 architecture and context.

4 Research and Innovation Smart Specialization Strategy Process Capability Model

All RIS3 related activities could be distributed into two sequential iterative stages: (a) RIS3 definition and (b) RIS3 implementation acting in RIS3 governance environment including monitoring and evaluation. Therefore, three corresponding processes subcategories have been introduced. RIS3 definition, RIS3 implementation and RIS3 governance processes subcategories are defined.

4.1 RIS3 Definition Process Subcategory

The description of RIS3 processes that satisfies ISO/IEC 15504-2 and ISO/IEC 33004 requirements for Process Assessment Model [13, 14] is developed. The description of base practices is provided as an example for process DEF.1. "Identification of EDP related knowledge" only. RIS3 definition process subcategory consists of 18 processes is provided in Table 1.

4.2 RIS3 Implementation Process Subcategory

Regional research and innovation smart specialization strategy implementation process subcategory from institutional point of view consists of 4 processes provided in Table 2.

Table 1. RIS3 definition subcategory processes' description.

DEF.1. Identification of EDP related knowledge	
Purpose	Outcomes
To identify preliminary knowledge potentially possessed by Entrepreneurial agents for transformation of the region on the basis of inclusive smart specialization	(1) Entrepreneurial knowledge related to science, technology, engineering and practical innovation is identified; (2) Entrepreneurial knowledge related to market growth potential, potential competitors, set of inputs and services required for launching a new activity is identified; (3) Knowledge related to aggregation of individual entrepreneurial knowledge and facilitation of EDP is identified; (4) Knowledge related to societal values and possible ways of their achievement is identified
Base practices	
DEF.1.BP1: Identify entrepreneurial knowledge related to science. Identify entrepreneurial knowledge related to science from the point of view what can be commercialized **DEF.1.BP2: Identify entrepreneurial knowledge related to technology.** Identify entrepreneurial knowledge related to technology from the point of view what can be commercialized **DEF.1.BP3: Identify entrepreneurial knowledge related to engineering.** Identify entrepreneurial knowledge related to engineering from the point of view what can be commercialized **DEF.1.BP4: Identify entrepreneurial knowledge related to practical innovation.** Identify entrepreneurial knowledge related to practical innovation from the point of view what can be commercialized **DEF.1.BP5: Identify entrepreneurial knowledge related to market.** Identify entrepreneurial knowledge related to market growth potential, potential competitors, set of inputs and services required for launching a new activity **DEF.1.BP6: Identify aggregation knowledge.** Identify knowledge related to aggregation of individual entrepreneurial knowledge and facilitation of EDP **DEF.1.BP7: Identify knowledge on societal values.** Identify knowledge related to societal values and possible ways of their achievement	
DEF.2. Identification of requirements to entrepreneurial agents	
Purpose	Outcomes
To identify requirements for EDP roles on the quadruple-helix-basis based on EDP related knowledge for agents who best understand the strengths, capabilities, constraints and limitations of a territory in order to identify regional assets, potential, target state and a way to reach it - that constitute the basis of inclusive smart specialization in the region	(1) Requirements for Entrepreneurial actors representing firms, higher education institutions, research institutions, independent innovators are identified; (2) Requirements for policy makers – public sector institutions as RIS3 driving force and EDP facilitator are identified;

(*continued*)

Table 1. (*continued*)

	(3) Requirements for representatives of society with focus to societal value are identified
DEF.3. Identification of policy makers	
Purpose	Outcomes
To activate and nominate institutional resources for entrepreneurial knowledge integration, synthesis and processing for the development of RIS3 strategy in the region	(1) RIS3 driving force and leaders are enlisted; (2) EDP facilitator and active participant is nominated; (3) The responsibility for aggregation of entrepreneurial knowledge embodied in and possessed by various relevant actors is assigned; (4) The responsibility for synthesis and processing of aggregated entrepreneurial knowledge is assigned
DEF.4. Facilitation of open consultation	
Purpose	Outcomes
To build a systematic understanding of the areas in the economy and society that have the greatest potential for future development or need to be encouraged and extracted	(1) Reactive selection of stakeholders is performed; (2) The stakeholders that are the most capable of providing societal value and entrepreneurial knowledge are chosen; (3) Proactive communication and collaboration of entrepreneurial agents is supported; (4) Wide understanding that no actor is omniscient and the more inclusive the process of knowledge collection, the more comprehensive the knowledge base at the disposal of policy makers is achieved; (5) An understanding that the integrated and aggregated entrepreneurial knowledge is greater than the sum of its individual parts is reached; (6) Societal engagement that contributes to the local ownership of RIS3 process and strategy is broadened
DEF.5. Identification of entrepreneurial actors	
Purpose	Outcomes
To capture the source of entrepreneurial knowledge used to inform and guide the development of RIS3 strategy in the region	(1) Inclusive identification of entrepreneurial actors is performed; (2) Engaged in EDP entrepreneurial actors are enlisted; (3) Entrepreneurial knowledge in the region possessed in principle by entrepreneurial actors enlisted is covered;

(*continued*)

Table 1. (*continued*)

	(4) Understanding of the commercial viability of activities and opportunities and market dynamics is enabled. (5) Related complementary entrepreneurial knowledge outside market centric activities is possessed; (6) Entrepreneurial knowledge related to development of a comprehensive knowledge base used to inform the smart specialization strategy is possessed
DEF.6. Shaping the context for entrepreneurial actors	
Purpose	Outcomes
To create environment within which entrepreneurial actors exist, emerge, function, interact and generate entrepreneurial knowledge	(1) The region's conduciveness to economic activity as the main precondition to the conduciveness of a region to supporting entrepreneurial actors is established; (2) A framework within which economic activity occurs is delineated; (3) Generic formal "rules of the game" in a society are established, understood and accepted; (4) Well-functioning and appropriately monitored, efficient and favourable for entrepreneurial actors institutional context is established; (5) Entrepreneurial actors exist, emerge, function, interact and generate entrepreneurial knowledge; (6) Markets characterized by economic and regulatory barriers to entry and by lengthy and punitive bankruptcy proceedings, and judicial and political contexts that militate against neutrality in the legal enforcement of private contracts as climates that are inimical to the translation of entrepreneurial discoveries of profit opportunity into bases for sustainable legal enterprise are removed
DEF.7. Engaging of entrepreneurial actors in exploratory behaviour	
Purpose	Outcomes
To create balanced conditions for EDP performance and to mitigate market failures.	(1) In order to resolve "incomplete appropriation" problem and feasibility to entrepreneurial actors to realize a sufficient private benefit relative to potential public benefit from exploratory practices and innovation is ensured; (2) Maximization of spill-overs within entrepreneurial discoveries is rewarded;

(*continued*)

Table 1. (*continued*)

	(3) Favourable conditions to entrants to discovery based new activity are created; (4) When exploratory activity requires a collective effort, in order to mitigate “coordination failures” problem by institutional measures, the efforts of individual actor are protected; (5) Subsidization and financial support mechanisms for resolution of market failures inhibiting exploratory practices are ensured
DEF.8. Facilitation the transmission of entrepreneurial knowledge	
Purpose	Outcomes
To enable EDP information to elicit, integrate, process, synthesize and use for RIS3 design	(1) Direct communication of policy makers with entrepreneurial actors to enable immediate channel between those capable of providing knowledge and those tasked with collecting it is ensured; (2) Intermediate distance between full autonomy and full embeddedness is established; (3) The trust, openness and transparency, the dialog, interaction and overall closeness that enables the effective communication of entrepreneurial knowledge by strong, well-functioning institution is facilitated; (4) The competence and productivity based on collaborative learning and participatory attitude is increased
DEF.9. Macro analysis of regional assets for innovation	
Purpose	Outcomes
To assess the existing assets, to evaluate major regional strength, to identify any bottlenecks of the innovation system and key challenges both for economy and for society	(1) Relevant stakeholders for regional self-assessment are identified; (2) In order to perform self-assessment relevant stakeholders are contacted, the guiding questions are distributed, necessary steps and milestones are organized, i.e. self-assessment is prepared; (3) The assessment of each dimension, i.e. enterprises, knowledge institutions, government, civil society – is performed; (4) Cross-dimensional assessment as mutual outside view is adopted; (5) First SWOT analysis is performed

(*continued*)

Table 1. (*continued*)

DEF.10. Positioning of the region	
Purpose	Outcomes
To identify competitive advantages through systematic comparison with other regions, mapping the national and international context to mark the differences and performing effective benchmarking	(1) Relevant linkages and flows of goods, services and knowledge of the region outside the administrative boundaries are identified; (2) Possible patterns of integration with partner regions are revealed; (3) The need to source know-how and technology from the rest of the world is identified; (4) Strategic positioning of the region relative to other regions of the Europe is performed
DEF.11. Prospect for a process of entrepreneurial discovery	
Purpose	Outcomes
To build a systematic understanding of current performance of EDP in the region, feasibility to generate entrepreneurial knowledge and the need to support of entrepreneurial discovery	(1) Statistics on entrepreneurial activities is developed; (2) A set of consultations and auditing tools is provided; (3) Direct discussion among entrepreneurial actors, management and governance bodies responsible for RIS3 is engaged; (4) Appreciation of entrepreneurial dynamics is achieved; (5) Existence of functioning entrepreneurial discovery process is assessed; (6) Feasibility to generate a significant flow of experiments, innovation ideas and entrepreneurial discoveries is evaluated; (7) The need to specifically support entrepreneurial discoveries, if necessary, is identified
DEF.12. Candidate priority definition	
Purpose	Outcomes
To provide policy makers with dominant characteristics and preliminary development priorities of their own administrative regions	(1) The combined place-specific features of a region are identified; (2) Individual administrative region within regional, EU and world-wide context is positioned; (3) Candidate priorities of the regional economy grounded by macro-analytical evidence are defined; (4) Specific development paths are identified

(*continued*)

Table 1. (*continued*)

DEF.13. Elaboration of an overall vision	
Purpose	Outcomes
To develop a shared and compelling vision on the economic development potential of the region, main direction for its international positioning and relevance in terms of meeting societal challenges	(1) Target stage for economic and societal transformation is defined and agreed; (2) Comprehensive scenario for transformation of the regional economy, society and environment by all stakeholders is shared; (3) Political endorsement for the vision and for subsequent steps for strategy definition and implementation is achieved; (4) Regional stakeholders' feeling they can contribute to and benefit from regional transformation being implemented is achieved; (5) The ground to reach the willingness to act towards the transformation of the region and support the regional consensus is build
DEF.14. Criteria definition for allocative rule	
Purpose	Outcomes
To define financial instrument to support RIS3 development and implementation by combining the macro-level analysis and the observation of micro-dynamics	(1) The vision of improved region and transformation scenario of a region as starting point for allocative rule criteria definition are elaborated; (2) Inclusive support by allocative rules is assumed; (3) The principle of EDP based transformation supported by allocative rules is respected, i.e. EDP discoveries, if present, if not - EDP establishment is supported; (4) The loop for EDP facilitation: observe and detect EDP or create conditions for EDP by EDP supporting allocative rule is closed; (5) The criteria of allocative rule as integration of shape based macro level analysis and local knowledge of micro level discoveries are formulated
DEF.15. Analysis and support of discoveries	
Purpose	Outcomes
To observe carefully bottom-up process, identify entrepreneurial discoveries, assess the outcomes, help the most promising projects to grow and increase probability that entrepreneurial discovery happens	(1) Direct dialog between institutions and entrepreneurial actors is established; (2) Commercially viable lines of business that are not likely to be able to completely appropriate the social benefits that flow from exploitation of such discoveries are identified for support;

(*continued*)

Table 1. (*continued*)

	(3) The impediments for discovery like incomplete appropriation are detected; (4) The incentives to compensate for the risky nature of entrepreneurial search and discovery activities without granting the discoveries monopolies in the rights are provided; (5) The diffusion of the knowledge regarding the value of a new activity for future specialization is assured; (6) Most suitable policy instruments from a wide range of instruments in support to discovery happens are defined
DEF.16. Critical mass assembly	
Purpose	Outcomes
To assist in the formation and effective alignment of the crucial connections, and to developing public sector capabilities and mechanisms that support the coordination of complementary early efforts by private agents in the sphere of experimentation and discovery	(1) The phase when a single discovery begins to be translated into a collective phenomenon is identified; (2) The number of agents and organisations willing or able to invest in particular type of discovery is estimated; (3) The need for intervention to discovery to avoid social value failure is identified; (4) The impediments to increase critical mass for discovery like insufficient spill-overs are detected; (5) Encouragement of imitative entry is promoted; (6) The resolution of coordination failure problem is facilitated
DEF.17. Discovery definition	
Purpose	Outcomes
To assess projects or domains as candidate discoveries based on defined selection criteria for inclusion into priorities	(1) Discovery characteristics like considered activity is new or it aims at experimenting and discovering opportunities are identified; (2) Discovery potential to generate valuable information and learning spill-overs (Information externalities) is identified; (3) Qualifying characteristics of discoveries as a proliferation of entrants into new activity is established; (4) The discovery's likelihood to initiate a desirable structural change (modernization, diversification) for the region is assessed; (5) The need for support for discovery implementation is defined;

(*continued*)

Table 1. (*continued*)

	(6) Discovery's key supply factors including human capital availability and accessibility is assessed; (7) A demand and main competitors are screened
DEF.18. Identification of priorities	
Purpose	Outcomes
To match between a top-down process of identification of broad objectives aligned with EU policies and bottom-up process of emergence of candidates niches for smart specialization, areas of experimentation and future development stemming from the discovery of entrepreneurial actors	(1) The priority's proximity to market is assessed; (2) The priority's richness in innovation and spill-overs is assessed; (3) Critical mass of priority's resources is accessed; (4) The need of priority for support and financial value is evaluated; (5) The significance of priority to the region is assessed; (6) The capacity of the region to digest priority is assessed; (7) The connectedness of priority to the regional economy is assessed; (8) Top-down priorities are revised; (9) Horizontal priorities are defined

Table 2. RIS3 Implementation category processes' description.

IMP.1. Definition of roadmaps/action lines	
Purpose	Outcomes
To provide design for implementation of priority on the basis of evidence on their effectiveness and relevance for the implementation of priorities identified, justified according to their contribution to the overall strategy goals, including tentative Allocative rules	(1) Existing programmes and policy instruments in a region on the basis of evidence on their relevance for the prioritised areas are identified; (2) Relevant existing programmes and policy instruments on the basis of evidence on their effectiveness are incorporated; (3) New policy instruments, justified according to their contribution to the overall strategy goals are included; (4) Action lines and policy instruments are accompanied by measurable indicators, that reflect outputs achieved and outcomes reached

(*continued*)

Table 2. (*continued*)

IMP.2. Development of action plan	
Purpose	Outcomes
To detail and organize all the rules and tools a region needs in order to reach prioritised goals and to provide for comprehensive and consistent information about strategic objectives, timeframes for implementation, identification of funding and tentative budget allocation	(1) Pilot projects are defined; (2) Target groups are defined; (3) Actors involved and their responsibilities are defined; (4) Baseline values, measurable targets to assess both outputs and outcomes are defined; (5) Timeframes are defined; (6) Funding sources, targeted to the several groups and projects, are identified; (7) Tentative budget is allocated
IMP.3. Balancing of targeted and horizontal measures	
Purpose	Outcomes
To define appropriate mix of measures based on identified technological and horizontal-type priorities to increase reuse and avoid multiplication	(1) Actions for support to technological (vertical) priorities implementation are identified; (2) Identical or similar actions of various vertical priorities, that can be substituted by single integrated action, are identified; (3) Integrated actions to substitute similar actions identified are defined; (4) Identified similar actions of vertical priorities are substituted by links to integrated actions; (5) Based on integrated actions defined horizontal actions groups are composed; (6) Based on horizontal actions groups horizontal-type priorities are identified
IMP.4. Creation of framework conditions	
Purpose	Outcomes
To provide tools, services and financial instruments to support policy experimentation and allow testing mixes of policy measures at a small scale, before deciding on implementation at a larger and more expensive scale	(1) Policy experimentation goals are defined; (2) Testing of policy mix measures at a small scale is defined; (3) Pilot projects to perform testing effectively are launched; (4) Effective evaluation mechanisms leading to sound appraisal of success and feasibility as mainstream RIS3 projects is coupled

4.3 Governance Process Subcategory

Regional research and innovation smart specialization strategy governance process subcategory consists of 5 processes provided in Table 3.

Table 3. Governance subcategory processes' description.

GOV.1. Ensuring collaborative leadership and ownership	
Purpose	Outcomes
To ensure that all stakeholders own and share the strategy to let each actor to have a role and take the lead in specific phases of RIS3 design according to actors' characteristics, knowledge and capacities	(1) The scope of regional RIS3 to align different expectations an agendas on the question at stake of different stakeholders, often restricted by own areas of action, is defined; (2) The goal of ensuring participation of the key actors and securing ownership of the approaches defined in the strategy is defined; (3) Potential actors relevant to the regional RIS3 process who can contribute to the benchmarking and peer review processes by open, simple and transparent procedure are invited; (4) A wide view of innovation and regional improvement is adopted; (5) Quadruple-helix representation is ensured; (6) Boundary spanners as intermediators and moderators are included; (7) Interactive, regionally-driven and consensus-based RIS3 process is ensured
GOV.2. Establishment of governance structure	
Purpose	Outcomes
To establish responsible body that ensures regional RIS3 strategy design and implementation for regional transformation, including analysis, experimentation, debates and decision-making, with a participation of actors and experts from within and outside the region	(1) A person at highest possible political level as regional RIS3 owner is denoted; (2) The quadruple-helix representatives are nominated; (3) The stakeholders and decision-makers from multi-level dimension policies of RIS3 context are included; (4) With respect to policy areas and organisations concerned are beyond the traditional science and technology and related ministries, inter-ministerial body is involved; (5) The regional knowledge ecology is represented
GOV.3. Verification	
Purpose	Outcomes
To assess whether the requirements to work products of RIS3 design and implementation are met	(1) Criteria for work product verification are identified; (2) Required verification activities are performed; (3) Defects are identified and recorded; (4) Results of verification are made available to involved parties

(*continued*)

Table 3. (*continued*)

GOV.4. Monitoring	
Purpose	Outcomes
To learn about actual transformation process and informing accordingly, to build and reinforce trust and cooperation with and among stakeholders and citizens, and to guarantee accountability of policy making	(1) Performance of activities against the plans is verified; (2) Correctness of funds use and spending on delivering planned outputs is checked; (3) Evolutions of outcomes indicators against target values is verified; (4) Pre-conditions for conducting evaluation are created; (5) Monitoring system and evaluation design are integrated; (6) Information on RIS3 implementation is gathered and systematized; (7) Learning on process failure is preceded being irreversible
GOV.5 Evaluation	
Purpose	Outcomes
To assess whether and how strategic objectives are achieved and assess the effects of the actions undertaken	(1) Target values of outcomes indicators defined by RIS3 strategy are identified; (2) Related monitoring data are identified; (3) Effect of public intervention is evaluated; (4) A causal link of public intervention, i.e. Does it work? is established if present; (5) The answer to the question: Why an intervention produces intended and/or unintended effects?, i.e. Why it works? is provided; (6) The answer to the question: How it works? is provided

5 Conclusions and Future Work

A SPICE conformant application dependent process modeling methodology has been applied for the modeling of regional research and innovation activities. The approach for Enterprise SPICE extension by the inclusion of application domain dependent PAMs has been proposed. The process assessment model for regional research and innovation smart specialization strategy processes introduced in this paper has been developed as Enterprise SPICE extension.

Within the framework of the INTERREG P2L2 Project several pilot implementations of a guided self-assessment had taken place based on an adaptation of the model proposed here [1]. The development of regional innovation strategies is an important topic with a global dimension, far beyond the European Union. The authors of this paper are confident, that the model presented here is addressing an urgent need for a more structured, transparent approach that allows continuous improvement also of the

RIS3 processes, that had not been conceptualized before. The model is not strictly bound to the European approach of RIS3, by the nature of a process reference model it is built with a certain level of abstraction. This abstraction allows to consider the model as a generalized approach to regional innovation strategy development, making it applicable also in regions outside of the European Union. However, more extensive trials in different regions for regional research and innovation smart specialization strategy process assessment and improvement are needed for an appropriate validation of the model.

A very promising approach for the development of regional innovation systems is expected to be found in the combination of the standard based Model innoSPICE [5] which is focusing on the processes of the individual and organizational level in knowledge- & technology transfer and innovation (micro-level). As the developed RIS3 model is focusing on the meso- to macro level, both models are covering the essential processes spanning a bow from the creation of new knowledge up to the implementation of a regional high level innovation strategy. This allows to assess the process capabilities of actors on very different level within a single regional innovation system. This approach will support a coherent "vertical" oriented management framework (connecting micro-, meso-, and macro- level) for regional development.

Acknowledgements. This document was supported within the framework of the P2L2 –Public Policy Living Lab Project, co-financed by the INTERREG Europe Programme, European Regional Development Fund.

References

1. Woronowitz, T., Boronowsky, M., Wewetzer, D., Mitasiunas, A., Seidel, K., Cotera, I.R.: Towards a capability maturity model for regional innovation strategies. Procedia Comput. Sci. **104**, 227–234 (2017). ICTE 2016, Riga, Latvia, December 2016
2. ISO/IEC 33071. Information technology – Process Assessment – An integrated process capability assessment model for Enterprise processes. First edition, 15 October 2016
3. Enterprise SPICE. An Integrated Model for Enterprise-wide Assessment and Improvement. Technical Report - Issue 1. The Enterprise SPICE Project Team, 184 p., September 2010. http://www.enterprisespice.com/page/publication-1
4. Boronowsky, M., Mitasiunas, A., Ragaisis, J., Woronowicz, T.: An approach to development of an application dependent SPICE conformant process capability model. In: Woronowicz, T., Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 61–72. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38833-0_6
5. Besson, J., Woronowicz, T., Mitasiunas, A., Boronowsky, M.: Innovation, knowledge- and technology transfer process capability model - InnoSPICE™. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 75–84. Springer, Heidelberg (2012)
6. Rodrik, D.: Industrial Policy for the Twenty-First Century. Harvard University, John F. Kennedy School of Government, Cambridge (2004)
7. Foray, D., David, P.A., Hall, B.H.: Smart specialization. From academic idea to political instrument. In: MTEI-Working paper (2011)
8. Foray, D., et al.: Guide to Research and Innovation Strategies for Smart Specializations (RIS3), May 2012

9. Rodriguez-Pose, A., Winkie, C.: Institutions and the entrepreneurial discovery process for smart specialization. In: Papers on Evolutionary Economic Geography (2015)
10. Smart Regions 2.0 Conference, June 2017. http://www.cvent.com/events/smart-regions-2017/custom-22-b3ab1cc6b3314d33a9b5b3d647f69d93.aspx
11. Foray, D.: On the policy space of smart specialization strategies. Eur. Planning Stud. **24**(8), 1428–1437 (2016). 10.1080/09654313.2016.1176126
12. Capello, R., Kroll, H.: From theory to practice in smart specialization strategy: emerging limits and possible future trajectories. Eur. Planning Stud. **24**(8), 1393–1406 (2016)
13. ISO/IEC 33002. Information technology – Process Assessment – Requirements for performing process assessment (2015)
14. ISO/IEC 33004. Information technology – Process Assessment – Requirements for process reference, process assessment and maturity models (2015)
15. ISO/IEC 33020. Information technology – Process Assessment – Process measurement framework for assessment of process capability (2015)

Developing an Integrated Risk Management Process Model for IT Settings in an ISO Multi-standards Context

Béatrix Barafort[1], Antoni-Lluís Mesquida[2(✉)], and Antònia Mas[2]

[1] Luxembourg Institute of Science and Technology,
5 Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
beatrix.barafort@list.lu

[2] Department of Mathematics and Computer Science,
University of the Balearic Islands, Cra. de Valldemossa,
km 7.5, Palma de Mallorca, Spain
{antoni.mesquida,antonia.mas}@uib.es

Abstract. With risk management as a key topic for most organizations, aligning and improving organisational and business processes is essential. Capability and Maturity Models can contribute to assess and then enable process improvement. With the need to integrate risk management in IT settings (IT department/organisation), ISO/IEC 15504-330xx process assessment approach combined with ISO 31000 for risk management can be the foundations for new process models. An integrated process-based approach with various market-demanded ISO standards (ISO 9001, ISO 21500, ISO/IEC 20000-1 and ISO/IEC 27001) is proposed in the paper; it explains how the Integrated Risk Management Process Model for IT settings in an ISO multi-standards context is developed with a Design Science research method.

Keywords: Integrated risk management · ISO · IT settings · ISO/IEC 15504-330xx · Process reference and assessment models engineering · Design science research method

1 Introduction

Nowadays, risk management is a key topic for most of the organizations. Qualitative and quantitative approaches of risk management can be deployed. Capability & Maturity Models (C&MM) contribute to the community of practice by providing instruments for measuring process capability throughout process assessment and enabling improvement. Many models tackle risk management and propose various ways and mechanisms for process improvement. Organizations wishing to improve risk management face the problematic of choosing and selecting the adequate approach aligned to their business challenges and market positioning. Related to the area of C&MM, the International Standardization Organization (ISO) have published many years ago the international standard series on Process assessment (ISO/IEC 15504 [1]), now revised and published in the ISO/IEC 330xx standard series [2]. The main normative documents of the series provides requirements for a very structured and systematic approach for process

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 322–336, 2017.
DOI: 10.1007/978-3-319-67383-7_24

assessment, process reference and process assessment models description, and some guidance related to process assessment and improvement. This provides a consensus and was the basis for various initiatives proposing Process Reference Models (PRM) and Process Assessment Models (PAM) on the one hand at ISO level [3–5], and on the other hand at market level [6–8]. Among these various ISO/IEC 15504-330xx process models, none is dedicated to risk management. On top of that, in many IT organizations, management systems are needed and or required by the market in terms of certifications such as ISO/IEC 27001 [9] for information security management, ISO/IEC 20000-1 [10] for IT service management and ISO 9001 [11] for quality management. Project management remains a key concern in IT settings; even if it does not lead to ISO certification, the project management standard ISO 21500 [12] relies on a management system for mastering projects, including managing project risks. According to companies feedback and author experiences, these topics are the most commonly addressed by many IT organizations, whatever their size and domain; we have selected them for being part of our research.

In this context, we had investigated *how to integrate risk management in IT settings within a management system context?* in previous works [13]. By IT settings, we mean any IT department or IT organisation needing to integrate risk management activities. The authors made the assumption that an integrated risk management approach for IT settings will benefit organizations by being based on ISO standards which represent international consensus. They are the ground material of our research. With this background, our current research is investigating the following research question: *how to improve risk management processes in IT settings, in an ISO-multiple standards context targeting quality management, project management, IT service management and information security management, from a management system perspective?* For doing so, some more previous works have already cleared the field in order to identify processes for a new Integrated Risk Management process model for IT Settings (IRMIS) [14] based on the ISO 31000 standard for Risk management [15]. It is the international reference in the domain. With ISO 31000 as our guideline, the integration is considered regarding ISO 9001, ISO 21500, ISO/IEC 20000-1, and ISO/IEC 27001.

According to our research question, we aim at supporting Risk management processes improvement in IT settings, with a structured, integrated, interoperable, assessable, effective and efficient way via a PRM and a PAM as artefacts enabling process assessment and improvement. These two artefacts extend the ISO 31000 standard which is already process-oriented, but not structured neither organised for rigorous process assessment. So this paper presents the first results achieved with the development of a PRM and a PAM for IRMIS, implementing a Transformation process [16] supporting the design of process models according to ISO/IEC 15504/330xx. In order to develop these innovative artefacts, a Design Science Research Method [17] is followed.

After this introduction, Sect. 2 presents Related works and ISO standards inputs, and Sect. 3 the Design Science Research Method. Section 4 details the Transformation Process applied to ISO 31000, with the other ISO standards targeted in the IT settings scope of our research. Finally, Sect. 5 concludes the paper and presents research perspectives.

2 Related Work and ISO Standards Inputs

A lot of works have targeted Risk management in various domains. Capability & Maturity Models (C&MM) are amongst them. A recent paper presenting the LEGO approach to achieve a meta-model on Risk Management merging various sources, includes a survey on Risk management C&MM which has shown and compared their respective approaches [18]. There were not all similar in structure neither in levels. In order to avoid this, to ensure integration and consistency, to align with market demands and pressures related to certifications, we made the deliberate choice to focus on PRMs and PAMs fulfilling ISO/IEC 15504/330xx requirements on Process assessment and encompassing management systems principles. The economic benefits of standards is not to be anymore demonstrated in the industry [19], in particular with ISO certifications such as the most popular one: ISO 9001 [20].

We have studied existing and available PRMs & PAMs related to Risk management in C&MM context, based on ISO/IEC 15504/330xx and publicly available. Table 1 lists them.

Table 1. List of Risk management processes in existing Process models fulfilling ISO/IEC 15504-330xx requirements for PRM & PAM

Process model	Name of the Risk management related process(es)
ISO/IEC 15504-5:2012 – Part 5: An exemplar software life cycle process assessment model	MAN.5 Risk management
ISO/IEC 15504-6:2013 – Part 6: An exemplar system life cycle process assessment model	PRJ.5 Risk management
ISO/IEC 15504-8:2012 – Part 8: An exemplar process assessment model for IT service management	SMS.6 Risk management
Enterprise SPICE (ISO/IEC 33071:2016 – An integrated process capability assessment model for Enterprise processes)	GVM.9 Risk management
ISO/IEC 33072:2016 – Process capability assessment model for information security management	COM.11 Risk and opportunity management
ISO/IEC 33073 (under development) – Process capability assessment model for quality management	COM.11 Risk and opportunity management
ISO/IEC 30105-2: 2016 – Information technology – IT Enabled Services-Business Process Outsourcing (ITES-BPO) lifecycle processes – Part 2: Process assessment model (PAM)	ENB1 Risk management
Automotive SPICE Process Assessment Model	MAN.5 Risk management
COBIT Process Assessment Model (PAM): Using COBIT 5	EDM03 Ensure risk optimisation Manage risk

According to these processes, the risk management process, as tackled by the ISO 31000 standard, is very general. There is little difference among these processes, where risk identification is performed, and then analysis and evaluation, from the risk assessment perspective, and then risk treatment. There is not much detail in each of these PAM.

In addition to Table 1, some closely related works have been performed in the medical IT networks domain with a PRM and PAM for improving risk management, in order to allow Healthcare Delivery Organisations to assess the capability of their risk management processes against the requirements of IEC 80000-1 (application of risk management to IT-networks incorporating medical devices) [21]. There are 14 processes for different aspects of the life cycle risk management. In this process model, there are 4 processes dedicated to the risk management itself: Medical IT Network Risk Management, Risk Analysis & Evaluation, Risk Control, Residual Risk. This approach is targeting the medical sector with a particular objective of contribution to ISO 80000-1 but with a common overall goal with our works for improving risk management processes. We nevertheless address management systems from various selected ISO standards perspectives in an IT settings mind-set, as indicated in the next paragraph.

In previous works, the authors explored risk management in IT settings from the angle of selected relevant ISO standards driven by market demand and authors expertise (targeting quality management, project management, IT service management and information security management), with ISO 31000 as main theme. Table 2 provides the full list with identification numbers and titles of each considered standard, with an additional standard bringing valuable insights on information security risk management: ISO/IEC 27005 [22].

In previous works, the authors had shown that management system standards mechanisms are present in all quoted standards in Table 2. These mechanisms help integrating processes, and proposing common core processes as well as risk management dedicated processes in a single model addressing mechanisms for several types of risks (project, process, information security, IT services).

Table 2. List of relevant ISO standards supporting IRMIS PRM and PAM

ISO Standard number	ISO Standard title
ISO 31000:2009 [15]	Principles and generic guidelines on risk management
ISO 9001:2015 [11]	Quality management systems - Requirements
ISO 21500:2012 [12]	Guidance on project management
ISO/IEC 20000-1:2011 [10]	Information Technology - Service management - Part 1: Service management systems requirements
ISO/IEC 27001:2013 [9]	Information Technology - Security techniques - Information security management systems - Requirements
ISO/IEC 27005:2011 [22]	Information Technology - Security techniques - Information security risk management

3 Research Method

This research is based on Design Science principles. According to Denning, Design science is a "*problem-solving paradigm and seeks to create innovations that define the ideas, practices, technical capabilities and products through which the analysis, design, implementation, management and use of Information Systems can be effectively and efficiently accomplished*" [23]. Design Science aims to "*create things that serve human purposes, and then to create new and innovative artifacts*" [24] such as constructs, models, methods, and instantiations. Each designed artefact is aiming at improving the environment and the way to measure this improvement is investigated. By applying design science principles, we aim to guarantee the value chain linking research and technological activities.

Peffers et al. proposes a model describing the Design Science Research Method (DSRM) with a set of six activities in a nominal sequence [17]. Table 3 details these activities for the creation of the PRM and PAM artefacts.

Table 3. Design activities of the IRMIS PRM & PAM

1. Problem identification and motivation *This activity aims at defining the specific research problem and justifying the value of a solution. The problem definition will be used to develop an artifact that can provide a solution. In order to motivate the value of a solution, this set of activities includes knowledge of the state of the problem and the importance of its solution.* IRMIS PRM & PAM: Companies are facing multiple certifications and regulations which are critical for competitive advantage; risk management plays a central part in this multiple frameworks landscape. In this context, business and market constraints have been identified via contacts of the authors, and via their experience in process assessment and improvement. It has led to the problem motivation related to ISO standards which are critical, not only for risk management, but also for management systems, information security management, IT service management and project management. The problems practitioners face in industry regarding risk management improvement are then manifold in the context of ISO standards in IT settings
2. Define the objectives for a solution *This activity aims at inferring the objectives of a solution from the problem definition and knowledge of what is possible and feasible.* IRMIS PRM & PAM: in our case, the targeted solution for managing risk and improving risk management with a process-based approach in IT settings is a PRM & PAM integrating risk management and based on ISO standards. The objectives for this solution are connected and limited to ISO standards, and the solution need a structured, integrated, interoperable, assessable, effective and efficient way. What is possible and feasible has to be aligned with the requirements of ISO/IEC 33004 (Requirements for process reference, process assessment and maturity models) and to follow recommendations of the ISO/IEC 24774 (Guidelines for process description) [25]
3. Design and development *This activity aims at creating the artefact(s). These artifacts can be "constructs, models, methods, or instantiations" or "new properties of technical, social, and/or informational resources".*

(*continued*)

Table 3. (*continued*)

IRMIS PRM & PAM: For creating the PRM & PAM, the Transformation process is applied; it is a goal oriented requirements engineering (GORE) technique which was developed by the Luxembourg Institute of Science and Technology to provide clear guidance on how to transform a set of domain requirements into PRMs and PAMs which are compliant with the requirements of ISO/IEC 33004 and follow ISO/IEC TR 24774 guidance. The Transformation process advocates identifying elementary requirements and organising these requirements into requirement trees. These requirement trees are then oriented around the business goals to which they are related to form goal trees. The requirement and goal trees representation help PRM & PAM developers to visualize and support validation by experts. More details about this Transformation process can be found in [16]. The Transformation process is composed of nine steps. These steps are: 1. Identify elementary statements in a collection of requirements. 2. Organise and structure the requirements. 3. Identify common purposes upon those requirements and organise them towards domain goals. 4. Identify and factorise outcomes from the common purposes and attach them to the related goals. 5. Group activities together under a practice and attach it to the related outcomes. 6. Allocate each practice to a specific capability level. 7. Phrase outcomes and process purpose. 8. Phrase the Base Practices attached to the Outcomes. 9. Determine Work Products among the inputs and outputs of the practices. This Transformation Process is used iteratively in order to refine the grouping and process descriptions. Section 3 of the paper provides details for each step, and a particular process for illustration purposes. In the case of these works, we use the term “statement” instead of “requirements”, because our main ISO standard thread is the ISO 31000: this standard is not a management system one and does not provide requirements such as “shall” statements, but “should” statements
4. Demonstration *Design Science approach: This activity aims at demonstrating the use of the artifact to solve one or more instances of the problem. This can be done via the experimentation of the artifact’s use.* IRMIS PRM & PAM: This activity has not been performed yet; experimentation(s) of the use of the artefact are planned with expert domain reviews and process assessment experimentations
5. Evaluation *Design Science Approach: This activity aims at observing and measuring how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration. It requires knowledge of relevant metrics and analysis techniques.* IRMIS PRM & PAM: Following experimentation(s) of the artefact, a careful observation and measurement of the experimentation will be performed
6. Communication *Design Science Approach: This activity aims at communicating the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate.* IRMIS PRM & PAM: this current paper is part of the communication. The participation to ISO meetings and commenting similar artifacts contribute to the confrontation of these works to practitioners

After describing the six activities of the DSRM of our research works, next section will focus on the design and development of the artifacts.

4 Design and Development of a PRM and a PAM for an Integrated Risk Management Process Model Dedicated to IT Settings: A First Proposition

According to the Transformation process mentioned in Sect. 3, the PRM and PAM development has been performed. The first three steps have already been presented in [14], and are reminded here in order to provide a full view of the approach. Figure 1 provides an overview of the Transformation process, with the positioning of the various steps.

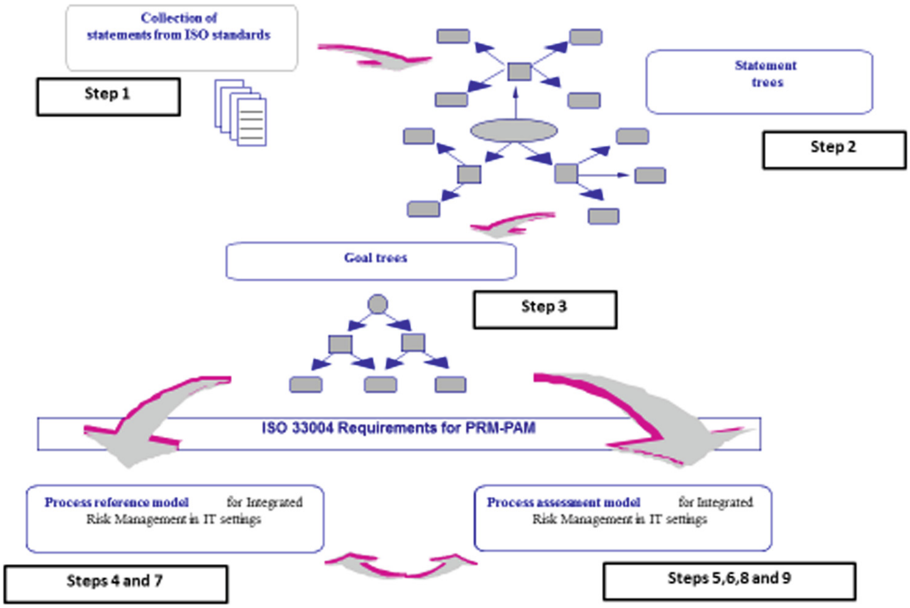

Fig. 1. Transformation process activities

In order to illustrate the Transformation process, this section shows the application of the Transformation process steps to one exemplar process of ISO 31000: the *Risk identification* process. This process belongs to the overall Risk management process, as stated in ISO 31000. The ISO 31000 standard is the main thread for the Transformation Process. Other standards are considered in a second time, once the structure of each identified process is determined. Our assumption is that the PAM will be contextualised to each targeted domain in an IT setting: for instance project management or information security management. The nature of the managed risks varies, but the mechanisms of the practices for managing risks in a management system environment does not.

Step 1: Identify elementary statements in a collection of statements

The first step consists in identifying all of the statements under the form of a collection of elementary statements. ISO 31000 provides, for each clause, a set of statements which are formulated mainly with "*should*" statements, also with "*may*", "*can*" or just information without any particular semantics format. The verbs in passive voice statements (revealing statements) were easily identified and split into elementary statements. Other sentences with a verb in present tense, clearly indicating an action to perform or a condition to be satisfied, were also considered elementary statements. When a sentence was composed of two parts separated by the coordination conjunction "and", it was divided into two elementary statements. If there was an enumeration, each element of the list was identified as an elementary statement. For the particular case of the *Risk identification* process, ten elementary statements were identified (Table 4).

Table 4. Elementary statements of the ISO 31000 for the Risk identification clause

The organization should identify sources of risk, areas of impacts, events (including changes in circumstances)
The organization should identify causes of risk
The organization should identify potential consequences of risk
Identification should include risks whether or not their source is under the control of the organization, even though the risk source or cause may not be evident
Risk identification should include examination of the knock-on effects of particular consequences, including cascade and cumulative effects
It should also consider a wide range of consequences even if the risk source or cause may not be evident
All significant causes and consequences should be considered
The organization should apply risk identification tools and techniques that are suited to its objectives and capabilities, and to the risks faced
This should include appropriate background information where possible
People with appropriate knowledge should be involved in identifying risks

Step 2: Organize, and structure the statements

During the second step, the elementary requirements were organized and gathered around the objects they are about in order to build a "statement tree" by applying mind mapping techniques. The elementary "should statements" were organized and structured under the form of a "mind map" for statement trees. A statement tree offers a graphical view of the connections between the components of each elementary statement. This "mind map" helped to have a graphical view of the elementary items having the same object (or component). A decision was made to distribute in various statement trees the set of statements; this was guided by the affiliation of statements within Clauses. These trees considered the Clauses and Sub-clauses titles, as well as the subject of each elementary item. This statement tree structuring was inspired by

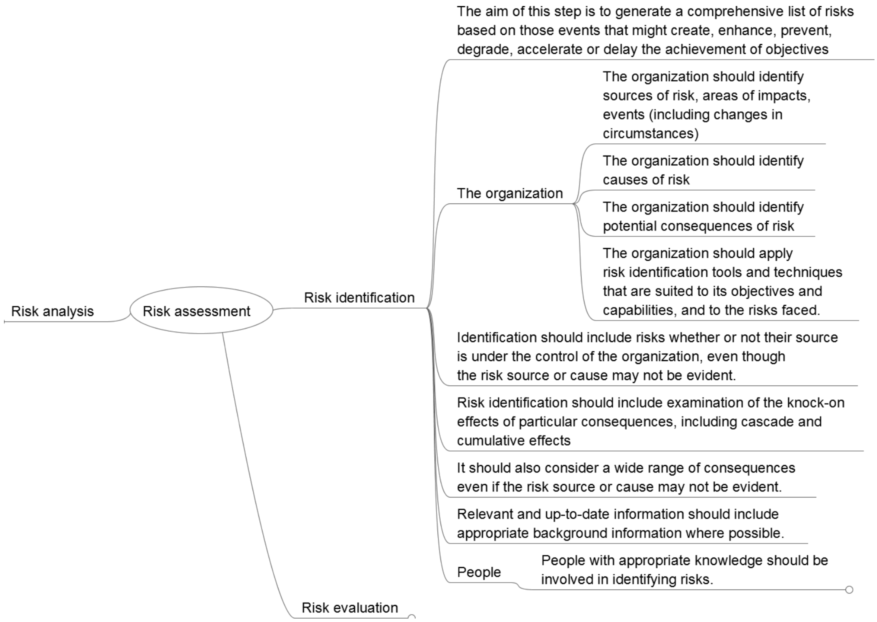

Fig. 2. Statement tree obtained for the Risk identification "object"

previous works where some groupings were similar. Risk identification was an "object" considered from the sub-clause (Fig. 2).

Step 3: Identify common purposes upon those statements and organize them towards domain goals

From the statements tree, some common purposes were identified and the elementary statements were organized accordingly, taking the original meaning of the ISO 31000 statements into account. A goal tree was then built for each common purpose, in which the inter-related activities were properly grouped. At this stage, we were able to identify processes, at least for a first proposal of a process list which may be refined according to the various iterations that are possible all along the Transformation Process. Common processes were identified from the management system mechanisms. In terms of Risk assessment, domain goals appeared with: Risk identification, Risk analysis and Risk evaluation, and then Risk Treatment. Sub-clauses in ISO 31000 guided these risk management dedicated processes.

At this stage of our research works, we identified four processes for the Risk management process group. From a process assessment practitioner point of view, this may be reviewed at the validation phase, with aggregation in two or event one single process for usability, efficiency and assessability reasons (Fig. 3).

Step 4: Identify and factorize outcomes from the common purposes and attach them to the related goals

An outcome is an observable result of (1) the production of an artefact, (2) a significant change of state, or (3) the meeting of specified constraints. The outcomes of each

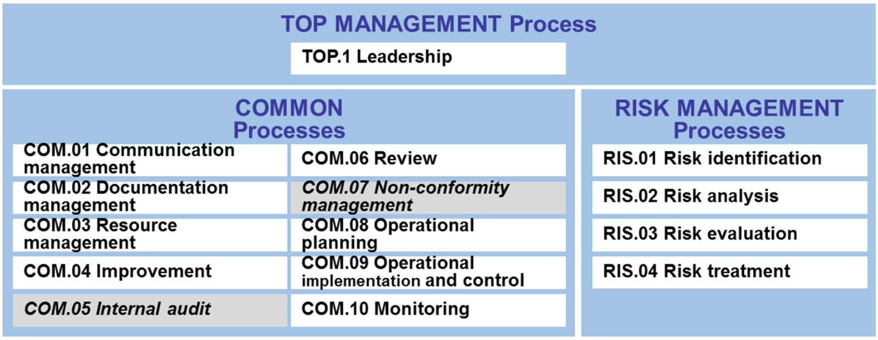

Fig. 3. IRMIS PRM proposed list of processes

process had to be factorized or merged, according to convenience and expert judgement, in order to define from 3 to 7 outcomes per process, and thus to follow the recommendations of ISO/IEC TR 24774 [25].

In some cases, the common purposes identified during step 3 were considered as the process outcomes and were attached to the related domain goals. In other cases, where a more detailed granularity level is wished, the common purpose supported the definition of a process purpose. Grouping of elementary statements then enable to identify outcomes.

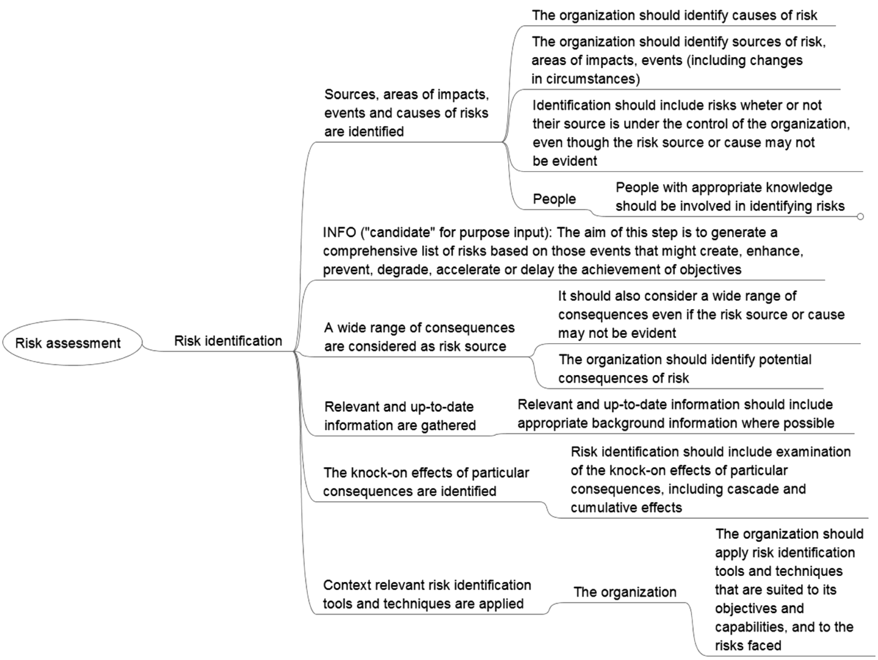

Fig. 4. Goal tree obtained for Risk identification

Table 5. The Risk identification process description in the IRMIS PRM

Process ID: RIS.1
Process Name: Risk identification
Process Purpose: The purpose of the Risk identification process is to generate a comprehensive list of risks based on those events that might create, enhance, prevent, degrade, accelerate or delay the achievement of objectives
Process Outcomes:
As a result of the successful implementation of the Risk identification process: 1. Context relevant risk identification tools and techniques are applied 2. Sources, areas of impacts, events and causes of risks are identified by personnel with appropriate knowledge 3. A wide range of consequences are examined and considered as risk source, including the examination of knock-on effects of particular consequences

The goal tree for the *Risk identification* process (Fig. 4) shows the resulting process outcomes before final proposition of 3 outcomes (Table 5).

Step 5: Group activities together under a practice and attach it to the related outcomes

The original input of the Transformation process (the statements from ISO 31000) contains information describing activities that should be conducted for implementing the processes. According to the number and level of detail of these activities, they were grouped as practices. Each practice represents a functional activity of the process. When implemented, a practice contributes to the achievement of at least one outcome of the performed process. During this step, we linked these activities or practices to the related outcomes and we kept traceability between each practice and the initial set of elementary statements. Indeed, it is possible that several elementary statements are related to (or hidden behind) only one practice of a process. The goal trees enable to keep that in mind for further activities, in particular, when questionnaires are being developed for supporting process assessment.

Step 6: Allocate each practice to a specific capability level

During this step and for each process, we review the practices and their linked outcomes in order to be sure that they contribute to the process performance attribute (capability level 1) of their associated process.

We ensured that our process descriptions are such that no aspects of the measurement framework beyond level 1 are contained or implied and thus, that the created process reference and process assessment models comply with ISO/IEC 33004.

Step 7: Phrase outcomes and process purpose

'In order to create a process reference model that follows the guidelines of ISO/IEC TR 24774, each outcome has to be phrased as a declarative sentence using verbs at the present tense. Then, the purpose is phrased or refined if phrased when the process is identified to state a high-level objective for performing the process and provide measurable and tangible benefits to the stakeholders through the expected outcomes (process assessment concern). We also check that the set of outcomes is necessary and sufficient to achieve the purpose of the process. For the *Risk identification* process, the process description for the PAM is as follows:

The resulting IRMIS PRM is suitable for use in process assessment performed in accordance with the requirements for a PRM described in Clause 6.2 of ISO/IEC 33004.

(a) The declaration of the domain is: Integrated Risk Management for IT settings.
(b) The description of the processes is provided in the IRMIS PRM.
(c) The IRMIS PRM describe at an abstract level the processes implied by ISO 31000. The purpose of the IRMIS PRM is to facilitate the development of a process assessment model for integrated risk management.
(d) A description of the relationship between the processes defined within the IRMIS PRM is supported by a figure collecting all the processes by process groups.

The process descriptions are unique. The identification is provided by unique names and by the identifier of each process of the IRMIS PRM. Processes are described in terms of its purpose and outcomes. For all processes, the set of process outcomes are necessary and sufficient to achieve the purpose of the process. No aspects of the ISO/IEC 33030 Measurement Framework beyond level 1 are contained in process descriptions.

Once the PRM determined, critical aspects of integration with other selected ISO standards were tackled. The selected relevant standards were ISO 21500 and ISO/IEC 27001 supported by ISO/IEC 27005. ISO 21500 has a dedicated process for Risk identification. ISO/IEC 27001 does not provide much detail, but ISO/IEC 27005 does. So we used these standards for a PAM providing multi-application views.

Step 8: Phrase the Base Practices attached to Outcomes

Once the purpose and outcomes of a process is phrased, the process reference model is considered stable enough to phrase the base practices. Base practices are phrased as actions, starting with a verb at the infinitive, according to ISO/IEC 24774. During steps 8 and 9, we pay a particular attention to choose a wording that suits and that is commonly used for dealing with risk management in organizations in order to ensure a good adoption of the models. The context for Risk management will target project management in ISO 21500 and information security in ISO/IEC 27001.

Step 9: Determine Work Products among the inputs and outputs of the practices

A work product is an artefact associated with the execution of a process. During the steps 1 and 5, work products can be identified as one goes along. It is very clear that the main output work product for Risk identification is a "comprehensive list of risks". It is mentioned as "Risk register" in ISO 21500.

Table 6 presents a proposal of the PAM with multiple views, illustrated for ISO 21500 and ISO/IEC 27001.

The idea to provide views is to extend the ISO 31000 to the context of the other selected ISO standards, but to keep the ISO 31000 structure as the main line. The management systems mechanisms help the integration, but the specifics need to remain as such. The assessor will then be able to collect data with the appropriate context.

Table 6. The Risk identification process description in the IRMIS PAM

PAM IRMIS view	ISO 21500 view	ISO/IEC 27001 view completed by ISO/IEC 27005
Generic Risk Management	Specifics: project	Specifics: information security
BP1. Gather relevant and up-to-date information for the identification of risks (appropriate background information where possible) (Outcome 1)	Information comes as the project progresses through its life cycle	Information comes from the information security risks associated with the loss of confidentiality, integrity and availability for information within the scope of the information security management system
BP2. Apply context relevant risk identification tools and techniques. (Outcome 1)		
BP3. Identify sources of risk, areas of impacts, events (including changes in circumstances, and whether or not their source is under the control of the organization) and causes of risks (Outcome 2)	Identification of risks with a potential negative impact (threats) Identification of risks with a potential positive impact (opportunities)	Identification of assets Identification of threats Identification of existing controls Identification of existing vulnerabilities
BP4. Identify potential consequences of risk from a wide perspective including the examination of the knock-on effects of particular consequences such as cascade and cumulative effects. (Outcome 3)		Identification of consequences
Input Work Products: Risk management plan	Project plans	Criteria for performing risk assessments
Output Work Products: Risk register	Risk register	Risk identification

5 Conclusion

This paper has presented the work performed in order to develop an ISO/IEC 33004 compliant Integrated risk management in IT settings PRM and PAM (IRMIS) by applying a Transformation process. The resulting IRMIS PRM & PAM is covering the risk management guidance recommended by the ISO 31000 International Standard for the high level objectives of the PRM, and detailed and context-based indicators within the PAM, for process assessment purposes. The next stage of our research will consist in following all the steps of the DSRM in order to evaluate the results, and communicate them. This will allow companies to assess the capability of their risk management processes from an ISO-many fold perspective and then, to use the results as a basis for process improvement. For doing so, the IRMIS PRM & PAM will be validated through risk management expert opinion by collecting feedback. Other R&D experts working in process models for other domains are planned to be consulted. Demonstration and evaluation will also be carried out in industry. Different Risk management officers in IT settings (including Security officers of Information Systems, IT Project Managers and IT Service Managers) will be consulted about the suitability of the structure and contents of the IRMIS PRM and PAM. They will be asked to use these models in order to evaluate their effectiveness. Statement and goal trees could be used as a tool supporting validation of the models. All changes requested and comments obtained from the validation process will be incorporated into the final version of the IRMIS Framework.

Acknowledgements. This work has been supported by the Spanish Ministry of Science and Technology with ERDF funds under grants TIN2016-76956-C3-3-R and TIN2013-46928-C3-2-R.

References

1. ISO/IEC ISO/IEC 15504: Information technology – Process assessment, Parts 1-10. International Organization for Standardization, Geneva (2003, 2012)
2. ISO/IEC 330xx: Information Technology - Process assessment. International Organization for Standardization, Geneva (2013, 2017)
3. ISO/IEC 15504-5: Information Technology – Process assessment – An exemplar software life cycle process assessment model. International Organization for Standardization, Geneva (2012)
4. ISO/IEC 15504-8: Information Technology – Process assessment – An exemplar process assessment model for IT service management. International Organization for Standardization, Geneva (2012)
5. ISO/IEC 33072: TS Information Technology – Process Assessment – Process capability assessment model for information security management. International Organization for Standardization, Geneva (2016)
6. Automotive Spice, https://goo.gl/BNu8c2
7. TIPA for ITIL, https://goo.gl/EA9NMh
8. Lepmets, M., McCaffery, F., Clarke, P.: Development and benefits of MDevSPICE®, the medical device software process assessment framework. J. Softw. Evol. Process **28**(9), 800–816 (2016)
9. ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements. International Organization for Standardization, Geneva (2013)
10. ISO/IEC 20000-1: Information Technology – Service management – Part 1: Service management system requirements. International Organization for Standardization, Geneva (2011)
11. ISO 9001: Quality management systems – Requirements. International Organization for Standardization, Geneva (2015)
12. ISO/IEC ISO 21500: Guidance on project management. International Organization for Standardization, Geneva (2012)

13. Barafort, B., Mesquida, A.L., Mas, A.: Integrating risk management in IT settings from ISO standards and management systems perspectives. Comput. Stand. Interfaces (2016)
14. Barafort, B., Mesquida, A.L., Mas, A.: How to elicit Processes for an ISO-based Integrated Risk Management Process Reference Model in IT Settings? In: To be published in Proceedings of the 24th European System & Software Process Improvement and Innovation Conference 2017, Ostrava (2017)
15. ISO 31000: Risk management – Principles and guidelines (2009)
16. Barafort, B., Renault, A., Picard, M., Cortina, S.: A transformation process for building PRMs and PAMs based on a Collection of Requirements – Example with ISO/IEC 20000. In: 8th International SPICE 2008 Conference, Nuremberg (2008)
17. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. J. Manage. Inf. Syst. **24**(3) (2008)
18. Buglione, L., Abran, A., von Wangenheim, C.G., McCaffery, F., Hauck, J.C.R.: Risk management: achieving higher maturity & capability levels through the LEGO approach. In: 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 131–138. IEEE, October 2016
19. ISO, Economic benefits of standards – International case studies. ISBN 978-92-10556-7
20. ISO Survey (2015). https://goo.gl/lrkvkQ
21. MacMahon, S.T., McCaffery, F., Keenan, F.: The MedITNet assessment framework: development and validation of a framework for improving risk management of medical IT networks. J. Softw. Evol. Process **28**(9), 817–834 (2016)
22. ISO/IEC 27005: Information technology – Security techniques – Information security risk management – Requirements. International Organization for Standardization, Geneva (2011)
23. Denning, P.J.: A new social contract for research. Commun. ACM **40**(2), 132–134 (1997)
24. March, S., Smith, G.: Design and natural science research on information technology. Decis. Support Syst. **15**(4), 251–266 (1995)
25. ISO/IEC TR 24774: Software and systems engineering – Life cycle management – Guidelines for process description. International Organization for Standardization, Geneva (2010)

A Framework for Assessing Organisational IT Governance, Risk and Compliance

Mikhel Vunk[1], Nicolas Mayer[2(✉)], and Raimundas Matulevičius[1]

[1] Institute of Computer Science, University of Tartu, Tartu, Estonia
mihkel.vunk@gmail.com, rma@ut.ee
[2] Luxembourg Institute of Science and Technology,
5 Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
nicolas.mayer@list.lu

Abstract. Enterprises have reached to understanding that information technology (IT) is more than just a technical issue. Domains such as IT governance, risk management and compliance (GRC) have been established to steer it. Though there has been some improvements, these domains are usually considered separately, thus less business value is created due to complexity of the process flows. There has been little attempts to integrate all three aspects, however this was done using domain specific standard and not taking into account the existing state of the art. In this paper, we conduct a systematic literature review to understand the processes, roles, strategies, and technologies of IT GRC as well as their integration. Based on the results of the review, we propose an assessment framework, which could guide evaluation of the enterprise's IT GRC concerns.

Keywords: Governance · Risk management · Compliance · IT GRC · Systematic review

1 Introduction

Enterprises are facing challenges while governing their Information Technology (IT) resources and needs. Due especially to instability of the markets in the global financial system, competition pressure and corporate disasters in last decades, all corporations need to have focused on their governance, risk and compliance (Corporate GRC) activities. Basically, according to Racz *et al.*, GRC can be defined as "an integrated, holistic approach to organization-wide governance, risk and compliance ensuring that an organization acts ethically correct and in accordance with its risk appetite, internal policies and external regulations through the alignment of strategy, processes, technology and people, thereby improving efficiency and effectiveness" [1]. Therefore, ensuring that their IT supports their current and future GRC-needs, IT GRC has been derived. IT GRC is not new but it is still a subject of research. The main challenge of IT GRC is to have an approach as integrated as possible to IT governance, IT risk management and IT compliance. The aim is to improve effectiveness and efficiency of the three disciplines, mainly compared to the traditional silo approach generally performed within organizations.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 337–350, 2017.
DOI: 10.1007/978-3-319-67383-7_25

The scope of this study is to define a framework for IT GRC. Although there exist a number of studies that separately consider the IT governance, IT risk management and IT compliance challenges [2–4], little is done to integrate these domains together [5]. In this paper, the research question considered is *how IT governance, IT risk management and IT compliance could be integrated.*

To answer this research question, we have performed a systematic literature review, aiming at answering the following sub-questions: which processes have been defined for IT GRC, what roles of people are involved for IT GRC, what strategy is used for IT GRC, and what is considered as technology for IT GRC. Based on the review results, we proposed an integrated framework for assessing organisational IT GRC. The framework is supported by a web application, which could be used by organisations to assess their IT GRC practices.

The rest of the paper is organized as follows. In Sect. 2, we present the systematic literature review. Section 3 overviews the integrated framework for IT governance, IT risk management and IT compliance, including its implementation and validation aspects. Finally, Sect. 4 presents the concluding remarks and highlights the directions for future work.

2 Systematic Review of IT GRC

In this chapter, we present a systematic literature review and its components regarding IT governance, IT risk and IT compliance. Firstly, we describe the research method. Next we discuss the review protocol. Finally, we present the review results, thus constituting the state of the art for the integrated IT GRC framework.

2.1 Systematic Review Method

We have applied a systematic literature review method [6] to determine what is the state of the art in the IT GRC domain. The goal of our study is to understand how IT governance, IT risk management and IT compliance could be integrated. The review is executed through three stages – plan, conduct and report, as illustrated in Fig. 1. During the plan stage, we have specified the research question, developed and validated the review protocol. Second stage consists of the activities to conduct the research protocol. This included research identification, selection of the primary studies and assessment of their quality, and extraction and synthesis of the data. The final stage included preparation and validation of the report.

2.2 Review Protocol

Background: Enterprise processes are complex, involving IT not only as the technical issue but also including governance, risk management and compliance. However, IT governance, IT risk management and IT compliance are commonly dealt separately in silos. Hence the challenge is to integrate them to improve enterprises efficiency and effectiveness [7]. Typically, the integration of the three domains is referred as IT GRC, covering all the three disciplines. The literature review is conducted to find the state of

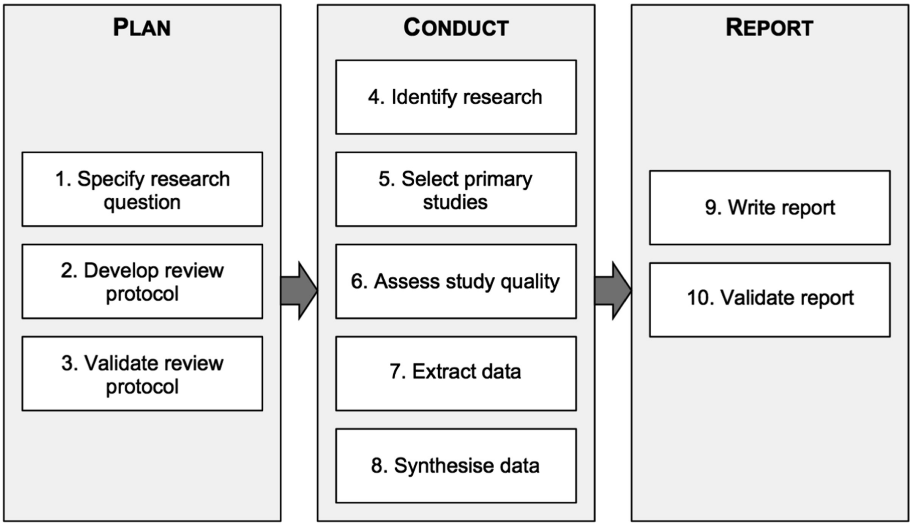

Fig. 1. Major steps for taking a systematic literature review. Three phases are expanded into tasks [6].

the art of IT GRC based on scientific literature. It is worth to note that, in terms of scope, we clearly distinguish here (Corporate) GRC from IT GRC, the latter being the subset of Corporate GRC dealing with IT [5].
Before defining the research question, we have conducted a small exploration over the secondary studies. It revealed a framework [7] which integrates IT governance, IT risk management and IT compliance based on ISO standards. However we did not identify any other integrated framework, for instance, resulting from the literature review.

Research questions. Task 1 of the systematic literature review process is about specifying the research questions (see Fig. 1). For this review, we used PICOC method (i.e., population, intervention, comparison, outcome and context) to create a frame for formulating research questions [6]. For population we chose “Enterprises relying their processes on IT, tangling in complexity for IT governance, IT risk management and IT compliance”. Intervention to improve them would be “Integration of IT GRC”. For comparison we are “Comparing IT GRC state of the art studies done so far”. The outcome of this paper is ought to be “Integrated framework for IT GRC, leading to a better effectiveness and efficiency of these domains in organisations”. Context for the research are: Proceedings and Journals.

The main research question is *how IT governance, IT risk management and IT compliance could be integrated?* Based on the frame of reference for GRC research [10], we have broken it into four sub-questions:

SQ1. Which processes have been defined for IT GRC?
SQ2. What roles of people are involved for IT GRC?
SQ3. What strategy is used for IT GRC?
SQ4. What is considered as technology for IT GRC?

The review protocol has been designed as follows (task 2):

Search strategy. The search was performed over three libraries – *ACM Digital Library*[1], *IEEExplore*[2] and *SpringerLink*[3]. Search queries for these libraries were based on an initial pseudo-query, which was formed from the main research question: "(*IT or information technology*) *and* ((*governance and risk and compliance*) *or GRC*)". This query, however, was modified for each library according to its search capabilities.

Selection Criteria and Procedures. The search query is constructed so that the main emphasis is on IT GRC variants either in title, abstract (e.g., ACM Digital Library) or without context constraint (i.e., IEEExplore and SpringerLink). To decide which studies to include (or exclude), inclusion (and exclusion) criteria are applied. Regarding inclusion criteria, we have included the study if the study is reported as a journal, proceeding or book chapter publication, and if its title or abstract contained *GRC* (or *governance, risk* and *compliance*). At the opposite, we excluded studies that contained discussions over only one or two domains (e.g., COBIT [2], De Smet and Mayer [8], etc.) as they are not directly comparable and because our objective is to survey the specific topic of IT GRC as a whole. However, we acknowledge that some relevant input can be found in domain-specific studies. Papers with different meanings for the GRC acronym (e.g., ground response curve) were obviously not included in the study, as well as studies already included earlier. Finally, we excluded the studies, which were relevant to the IT GRC domain, but that does not contain the information needed to answer our research questions.

Quality checklists. To measure the quality, the resulting studies are divided into two groups – (1) method, approach or framework presentation and (2) empirical study, such as survey, case study or experiment. Following guidelines of the systematic research method, we have applied a list of quality evaluation criteria, which help us to assess quality of the selected studies. Sample of the evaluation criteria includes presence of (*i*) the problem statement, (*ii*) the research questions, (*iii*) the research method description, (*iv*) illustrative example or related work, (*v*) discussion, (*vi*) conclusion and similar.

Data extraction strategy. Data is extracted using extraction forms. The initial forms were built using four initial studies, out of which one turned out to use another's results for the basis of integration standard. Thereby current forms are based on three studies [9–11]. The data extraction form consists of two parts. Firstly, we gather factual information about the paper (e.g., date of extraction, extractor, paper title, authors, short overview and quality score). Secondly we extract the contextual information regarding (*i*) the processes defined for the IT GRC, (*ii*) roles of people involved in IT GRC, (*iii*) strategy used for IT GRC, and (*iv*) technology applied for IT GRC.

Finally, regarding protocol validation (task 3), the review protocol was basically created by the first author of this paper and validated in the iterative discussion among all the authors (i.e., in the manner of student and supervisor discussion as mentioned in [6]).

[1] http://dl.acm.org/.

[2] http://ieeexplore.ieee.org/.

[3] http://link.springer.com/.

2.3 Systematic Literature Review Result

Task 4 from Fig. 1, *identify research*, results in search queries returning a total of 1444 results out of which were 168 from ACM, 105 from IEEE and 1171 from SpringerLink. After applying inclusion/exclusion criteria – task 5 – *select primary studies* to these results, 36 were included out of which 27 unique studies were left for *quality assessment (task 6)* and *data extraction (task 7)*. Main reasons for excluding the papers were: wrong acronym of GRC, not all domains were present or the scope of paper did not match with our corporate/IT GRC scope, or the quality indicators did not capture any required aspects.

After *quality assessment*, we have selected 7 primary studies. Due to small amount of studies found, the quality measure does not give an advantage in choosing sources of better quality amongst the seven included primary studies any more. Papers found suitable for the review are listed below:

- N. Racz, E. Weippl and A. Seufert, "Integrating IT Governance, Risk and Compliance Management Processes" [12].
- N. Racz, E. Weippl and A. Seufert, "Governance, Risk & Compliance (GRC) Software - An Exploratory Study of Software Vendor and Market Research Perspectives" [9].
- P. Vicente and M.M. da Silva, "A Conceptual Model for Integrated Governance, Risk and Compliance" [13].
- P. Vicente and M.M. da Silva, "A Business Viewpoint for Integrated IT Governance, Risk and Compliance" [10].
- M. Krey, "Information Technology Governance, Risk and Compliance in Health Care - A Management Approach" [11].
- D. Puspasari, M. Kasfu Hammi, M. Sattar and R. Nusa, "Designing a tool for IT Governance Risk Compliance: A case study" [14].
- A. Shahim, R. Batenburg and G. Vermunt, "Governance, Risk and Compliance: A Strategic Alignment Perspective Applied to Two Case Studies" [15].

Information is extracted from the studies into 4 categories: *processes, roles, strategies* and *technologies*. During extraction, we excluded from the results Mayer *et al.* [7] study. Although relevant, this study was firstly also reported as the secondary source in background study (see Sect. 2.2). Also we use this study to validate result of the current literature study (see Sect. 3.5). We have also excluded the paper by Racz *et al*. [1], since its results were recaptured in other two later papers by the same authors (see the list of selected papers).

The following sections present an overview of the extracted data from the included studies.

"Integrating IT Governance, Risk and Compliance Management Processes" [12], "Governance, Risk & Compliance (GRC) Software - An Exploratory Study of Software Vendor and Market Research Perspectives" [9]

The first paper introduces a high-level model from individual domain components as an artefact for IT GRC research knowledge base. IT governance process model is based on the ISO/IEC 38500:2008 standard for the corporate governance of IT.

Its IT risk process model is derived from the COSO ERM framework. The IT Compliance is covered by the process model suggested by Rath and Sponholz [16]. This way the developed model helps answering the *SQ1*.

In the second publication the author's study presents a survey from GRC software vendors on their perceptions of state-of-the-art IT GRC software. The survey potentially contributes with some description on the technology aspects, thus contributing to the *SQ4*.

Processes: The proposed process model is vertically split into three separate GRC domains, where the processes and their flow have been captured. Main flows are going from compliance to risk and from risk to governance. *IT Governance* tasks are evaluating, directing, reporting and monitoring. *IT Risk* domain holds internal environment, objective setting, risk assessment, risk response, control activities, information and communication, and monitoring. *IT Compliance* starts with requirements analysis, and continues with deviation analysis, deficiency management, reporting/documentation, and deviation analysis.

Technology: GRC software vendors have different perspectives on which functionality should be delivered by GRC software. The paper did not specify technology or tools, but listed their functionalities without domain affiliation. We extracted the functionalities proposed from survey as following: (*i*) *governance* should be supported with surveys, reporting, dashboards, analytics, conducting controls testing and management, and workflow management; (*ii*) *risk management* should be performed through case, issue, event, remediation, loss management, and operational risk management; finally (*iii*) *compliance* should be supported by functions for policy, audit, and compliance management.

"A Conceptual Model for Integrated Governance, Risk and Compliance" [13], **"A Business Viewpoint for Integrated IT Governance, Risk and Compliance"** [10]

The first paper presents conceptual models for governance, risk and compliance. The proposed model is assessed against the OCEG Capability Model. The newly developed model is rather extensive but basically it contributes to answering the *SQ1*.

In the second paper authors continue developing the integrated model. Thus they align it with the GRC state of the art and enforce it with the approach introduced by Racz *et al.* [9, 12] (see above). The new contribution is focussed on the *business viewpoint*. The study concludes that there exists a strong relation between the IT GRC and enterprise/corporate GRC, where the high level processes can be executed in both domains. The second paper contributes with the GRC role description, thus potentially gives an answer to the *SQ2*.

Processes: The major functionalities of the integrated GRC model are *audit management*, *policy management*, *issues management* and *risk management*.

Roles: In the study, a sample of actors, their roles and categories are presented. This includes: (*i*) *leadership and champions*, (*ii*) *oversight personnel* (e.g., board of directors), (*iii*) *strategic personnel*, like C-suite (e.g., chief information officer, chief compliance officer, chief audit executive, chief financial officer, chief risk officer, chief operations officer), information systems and system owners, process owners, and (*iv*) *operational personnel* (e.g., key-users, governance, risk, audit, controls, legal and compliance managers).

"Information Technology Governance, Risk and Compliance in Health Care - A Management Approach" [11]

This paper presents results of a survey where Swiss hospitals' environment was assessed using the CobiT Maturity Model. Here, however, the risk and compliance processes are not explicitly described and only activities regarding governance are explicitly extracted as processes. The study contributes with some generic recommendations to achieve compliance, thus also contributing to the answer of *SQ1*.

Processes. IT governance is described through strategic alignment, value delivery, resource management, and performance measurement. *Strategic alignment* (Business-IT-Alignment) ensures the linkage of business and IT plans (aligns operations between IT and enterprise). It defines, maintains and validates the IT value propositions. *Value delivery* guarantees that the value proposition is executed throughout the delivery cycle to ensure that IT delivers the promised benefits, concentrating on cost optimization. *Resource management* ensures the proper investment in and management of critical IT resources such as information, infrastructure, applications and people. *Performance measurement* tracks strategy implementation, process performance, resource usage, etc.

Compliance is initiated (not covered) by three steps: (*i*) identifying good practices of dealing with laws and regulations, (*ii*) improving personnel awareness in regulatory requirements and, thereby, (*iii*) increasing process performance of an enterprise and compliance with laws and regulations.

"Designing a tool for IT Governance Risk Compliance: A case study" [14]

This paper defines the IT GRC domain and reviews studies about IT GRC frameworks. The results of the review are used to develop some GRC application used in the bank domain. The paper contributes with few data to answer the *SQ1*.

Processes. Firstly, some functionalities regarding GRC management are presented such as policy and controls library, IT control self-assessment and measurement, IT asset repository, remediation and control management, basic compliance reporting, IT compliance dashboard, IT risk assessment and controls, and policy mapping. Secondly, a high level top-down perspective is presented from the senior management point of view.

"Governance, Risk and Compliance: A Strategic Alignment Perspective Applied to Two Case Studies" [15]

This study defines an integrated GRC approach, where it positions GRC to the integrated strategic perspective. This allows assessing the GRC maturity and its alignment paths. Two case studies are presented to explain the drivers to measure the effect of business-IT alignment on performance. Those examples reveal that the companies, which align their business with the IT strategies, have an advantage over other companies. The authors provide guidelines to assess company GRC-maturity and define paths to achieve strategic alignment. This study contributes to the answers of the *SQ3* question.

Strategy: The strategic alignment model is divided into external and internal domains, which both are split to the business and IT domains. While strategic fit integrates the external and internal domains, the functional integration connects business and IT domains.

Authors also define four paths to reach strategic alignment in GRC. For instance, the *strategy execution* indicates that GRC organisational strategy and infrastructure (in business domain) are the basis for choosing the IT domain infrastructure. Another path describes *technology transformation*, which shows scenarios to develop GRC strategy in the business domain and GRC solutions in the IT domain. The *competitive potential* path lets the GRC solution lead the GRC strategy and infrastructure in the business domain. Finally, the *service level* path describes how the GRC strategy is adopted to the GRC solution and then integrated in the GRC organizational infrastructure.

2.4 Summary

First to notice, there was quite small amount of studies qualified for the review at hand. Although we planned to identify the state of the art in four categories (processes, roles, technology and strategy), the main emphasis was found on the *process* category – four studies address process aspects while roles, technology and strategy are each addressed by only one study. The answer to systematic review protocol's main research question, a driver for this research, will be addressed in the next section as the literature review part captured answers regarding state of the art of IT GRC.

3 A Framework for Integrated IT GRC

In this section, we aim to define a framework for integrated IT GRC based on the state of the art performed. The proposed framework shall be an instrument to adopt the IT GRC activities within a company. It is meant to help in establishing the needed processes and to assess the maturity of IT GRC activities in a company that already has some. The main target group for this framework would be companies, which need integrated IT GRC approach.

3.1 Integrated IT GRC Model

To structure our proposed IT GRC framework, the approach is to synthesize data obtained during the systematic literature review into one model. As a base, we use the **frame of reference for integrated GRC** by Racz *et al*. [1] that is largely adopted according to the state of the art. In literature review, we tried to extract all four basic components of this frame of reference, i.e., strategy, processes, technology and people/roles. Since the review yielded results mostly in processes and extremely vaguely other components, we decided to use others as much as possible but main emphasis is on aligning processes to this triangle. As a consequence, we put the focus rather on GRC **main functionalities** as used by Vicente *et al*. [13] as the starting point for their conceptual model. These GRC main functionalities – *audit*, *policy*, *issue* and *risk management* – have been placed in the aforementioned GRC triangle. Finally, each main functionality is organized in our model around the **IT governance process flows** – *direct*, *evaluate*, *monitor* and *report* established by Racz *et al.* [17]. According to Racz *et al.*, "IT governance provides the frame for IT risk management and IT compliance decisions". To remove noise we left out groups which did not have any processes in (e.g. no *Direct* activities are related to *Audit management*). The resulting model is presented in Fig. 2.

3.2 Management Processes of the Four GRC Main Functionalities

Our main task to build our integrated IT GRC model is then to map the main findings obtained during the systematic review to the functionalities and process flows adopted.

Fig. 2. The Integrated IT GRC model

For each of the four GRC main functionalities (Audit, Policy, Issue and Risk management), we identify based on the systematic review the involved processes, associated roles and possible subprocesses. These processes are classified according to the process flow: *direct*, *evaluate*, *monitor* and *report*. Because of space limitation, only *Audit management* is detailed in this paper. The other GRC main functionalities are detailed in a technical report [18].

Audit management. Audit management consists in evaluating, reporting and monitoring tasks, since from the review results, its main tasks are focused on overseeing whether the compliance is obeyed. Following is the list of audit management processes and their definitions, as found in the literature. Audit management proposed processes and roles are presented in Fig. 3.

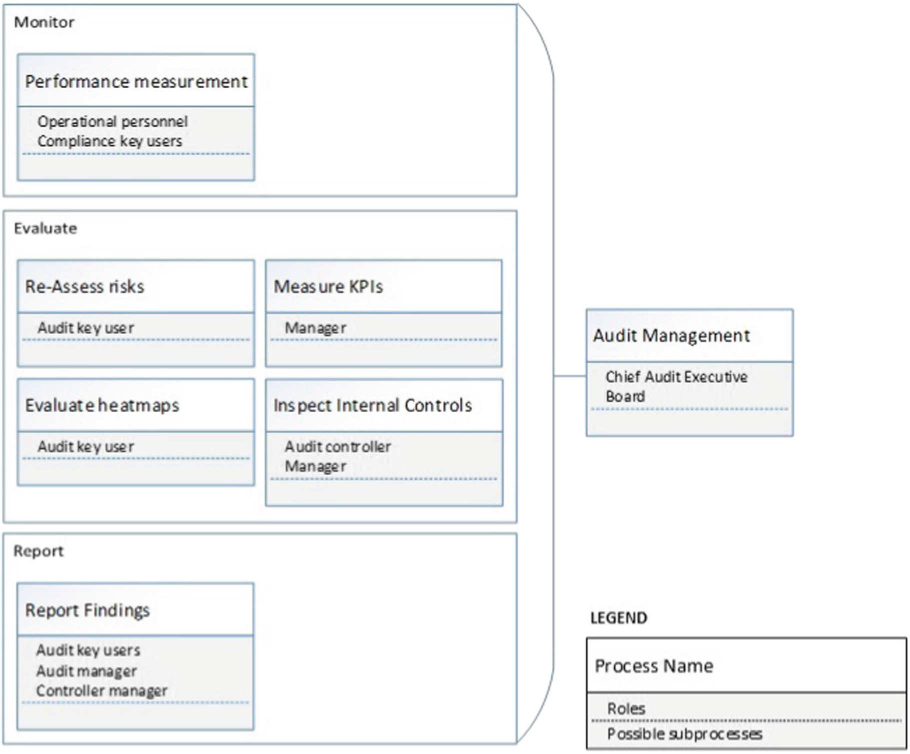

Fig. 3. Audit Management processes and roles

Audit management processes are:

- Evaluate
  - **Re-assess risks – risk assessment** – overall process of risk identification, risk analysis and risk evaluation [19].

  - **Inspect internal controls – (internal) audit** – “systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled” [20].
  - **Evaluate heatmaps** – evaluating current status of the auditable subject according to reported heatmaps [13].
  - **Measure KPI** (Key Performance Indicators) – measuring organization/IT/department performance using its agreed KPIs [13].
- Report
  - **Report compliance (-findings)** – “The governing body, management and the compliance function should ensure that **they** are effectively informed on the performance of the organization’s compliance management system and of its continuing adequacy, including all relevant non-compliances, in a timely manner” [20].
- Monitor
  - **Performance measurement** – “track and monitor strategy implementation, project completion, resource usage, process performance and service delivery, using, for example balanced scorecards that translate strategy into action to achieve goals measurable beyond conventional accounting” [7].

For each GRC main functionality, we use the following notation for presenting processes: the processes are displayed in a class diagram-like box as presented in Fig. 3, where process name is class name, proposed roles are above the line and possible sub-processes under the line in class members’ area. These processes are positioned in groups represented by rectangles with the group name in upper left corner. These groups are all connected by brace and form together the main functionality process put on the right side of the brace.

3.3 Implementation

To better visualise the IT GRC framework and help to assess companies’ maturity regarding IT GRC, a web application was developed[4]. The same components introduced in previous section are presented interactively. The main screen of the web application has the GRC-triangle in top of the screen including main functionalities and associated processes. Users can explore processes in the framework by clicking on these process flow elements. When clicking on the processes, a panel appears in the screen allowing performing a maturity assessment for each process related to the functionality. The maturity assessment of processes is performed on a scale of four items extracted from process assessment best practices: *Not achieved*, *Partially achieved*, *Largely achieved*, or *Fully achieved* [21].

3.4 Validation

We have established a 2-step validation protocol aiming at validating the completeness and soundness of our proposal. First, we compared our framework with the

[4] http://mihkel.joulukiri.ee.

ISO-specific one proposed by Mayer *et al*. [7]. In this work, Mayer *et al*. constructed an ISO-compliant IT GRC integrated model from the ISO standards related to the GRC individual domains. This framework has been chosen because its scope is equivalent to ours (i.e. IT GRC as a whole). Some more specific/focused ones, but better established, could also have been chosen in this validation step (e.g., COBIT for IT governance [2]). However, it would have only given a partial validation in terms of scope. The comparison is focused on processes, the ISO-compliant model of Mayer *et al*. being process-based. Then, as second step, the completeness and soundness of our model will be evaluated by a focus group composed of experts in the field, selected based on the systematic review results.

Comparison with the ISO-compliant IT GRC integrated model from Mayer *et al.* [7]. To compare the models, all the processes need to be processed in a comparable state. The comparison is done in a two-column table, both models being placed in columns and their functionalities/processes in rows accordingly. While detecting equivalence in the models, similar functionalities are grouped together in the same row or row-group (if several processes in one framework correspond to one in the second framework) and if no equivalence was found, an empty cell is on this row for the framework lacking the process. Details of this table can be found in a technical report [18].

In total we extracted 16 elements from Mayer *et al*. model and our model has 34 elements out of which 9 elements of Mayer *et al*. model corresponds to 14 elements in our model. 20 elements in our model have no direct correspondence in Mayer *et al*. model and 7 elements of Mayer *et al*. model have no correspondence in our model. As there are different numbers of corresponding elements in our model (14) to Mayer *et al*.'s model (9), Mayer *et al*.'s one had more compliance related elements, ours more risk management related elements. One assumption would be that the level of abstraction of the elements is not equal. In order to have them at the same abstraction level, more domain specific knowledge would be needed. Another finding is that, as Mayer *et al*.'s study based its framework on some non-IT specific reference documents (for the domains of risk management and compliance), the processes for their model are more generic and thereby have less details.

Validation with a focus group of experts. Second, we will assess the completeness and soundness of our model through a focus group. This focus group will be composed of authors of the papers selected during the literature review, as those authors were mainly in research groups dealing with the issue at hand and would be able to give the most relevant feedback. In addition to studies finally selected to be used in the review, the authors of all the relevant studies, which were excluded by some reasons, were also included to the focus group. This focus group will be asked to assess the proposed framework by going through the IT GRC framework web application (see Sect. 3.3) and complete a web form[5] associated to the framework. The feedback form consists of 4 pages split by main functionalities of the IT GRC framework (i.e. *Policy management, Issue management, Audit management and Risk management*), organized by process flows (i.e. *Direct, Monitor, Evaluate, Report*). Each process associated to

[5] http://mihkel.joulukiri.ee/evaluate/renderform.

functionality can be commented and assessed on the following scale: "definitely include", "maybe include", "maybe exclude" and "definitely exclude". This validation work is still in progress.

4 Concluding Remarks

In this paper, we described how we developed a framework for integrated IT GRC. The approach chosen was to perform first a systematic literature review of the IT GRC field. Following the systematic review protocol established, seven studies compose the results of our review. Then a proposal for the integrated IT GRC framework is made, based on a consolidation of the research results identified during the systematic review. This framework is implemented in a web application, to be used primarily as validation artefact. The proposed framework and its supporting web application are intended to assist companies to integrate their IT GRC processes. Application of the framework in real life could especially help assessing maturity of IT GRC according to the framework. Regarding future work, we first need to finish the validation work involving a focus group of experts and improve our model based on the conclusions drawn. Then, the use of our framework in an organization with a purpose of assessing IT GRC of this organisation will help us to check the adequacy and relevance of our approach.

Acknowledgments. Supported by the National Research Fund, Luxembourg, and financed by the ENTRI project (C14/IS/8329158).

References

1. Racz, N., Weippl, E., Seufert, A.: A frame of reference for research of integrated governance, risk and compliance (GRC). In: De Decker, B., Schaumüller-Bichl, I. (eds.) CMS 2010. LNCS, vol. 6109, pp. 106–117. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13241-4_11
2. ISACA: COBIT 5: A Business Framework for the Governance and Management of Enterprise IT (2012)
3. ISO/IEC 27005:2011: Information technology – security techniques – information security risk management. International Organization for Standardization, Geneva (2011)
4. ISO/IEC 38500:2015: Information technology - Governance of IT for the organization. International Organization for Standardization, Geneva (2015)
5. Racz, N.: Governance, Risk and Compliance for Information Systems: Towards an Integrated Approach. Sudwestdeutscher Verlag, Saarbrücken (2011)
6. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. School of Computer Science and Mathematics, Keele University (2007)
7. Mayer, N., Barafort, B., Picard, M., Cortina, S.: An ISO compliant and integrated model for IT GRC (Governance, Risk Management and Compliance). In: O'Connor, R., Umay Akkaya, M., Kemaneci, K., Yilmaz, M., Poth, A., Messnarz, R. (eds.) Systems, Software and Services Process Improvement. CCIS, vol. 543, pp. 87–99. Springer, Cham (2015). doi:10.1007/978-3-319-24647-5_8

8. De Smet, D., Mayer, N.: Integration of IT governance and security risk management: a systematic literature review. In: 2016 International Conference on Information Society (i-Society), pp. 143–148 (2016)
9. Racz, N., Weippl, E., Seufert, A.: Governance, risk & compliance (GRC) software - an exploratory study of software vendor and market research perspectives. In: 44th Hawaii International Conference on System Sciences, pp. 1–10 (2011)
10. Vicente, P., da Silva, M.M.: A business viewpoint for integrated IT governance, risk and compliance. In: 2011 IEEE World Congress on Services, pp. 422–428 (2011)
11. Krey, M.: Information technology governance, risk and compliance in health care - a management approach. In: 2010 Developments in E-systems Engineering, pp. 7–11 (2010)
12. Racz, N., Weippl, E., Seufert, A.: Integrating IT governance, risk, and compliance management processes. In: Proceedings of the 2011 Conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010, pp. 325–338. IOS Press, Amsterdam, The Netherlands (2011)
13. Vicente, P., Mira da Silva, M.: A conceptual model for integrated governance, risk and compliance. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 199–213. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21640-4_16
14. Puspasari, D., Hammi, M.K., Sattar, M., Nusa, R.: Designing a tool for IT governance risk compliance: a case study. In: 2011 International Conference on Advanced Computer Science and Information Systems, pp. 311–316 (2011)
15. Shahim, A., Batenburg, R., Vermunt, G.: Governance, risk and compliance: a strategic alignment perspective applied to two case studies. In: Hercheui, M.D., Whitehouse, D., McIver, W., Phahlamohlaka, J. (eds.) HCC 2012. IAICT, vol. 386, pp. 202–212. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33332-3_19
16. Rath, D.M., Sponholz, R.: IT-Compliance: Erfolgreiches Management regulatorischer Anforderungen. Erich Schmidt Verlag GmbH & Co., Berlin (2009)
17. Racz, N., Weippl, E., Seufert, A.: A process model for integrated IT governance, risk, and compliance management. In: Proceedings of the Ninth International Baltic Conference on Databases and Information Systems, DB&IS 2010, Baltic. pp. 155–170 (2010)
18. Vunk, M.: A framework for assessing organisational IT governance risk and compliance (2017). http://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=57229&year=2017
19. ISO 31000:2009: Risk management – principles and guidelines. International Organization for Standardization, Geneva (2009)
20. ISO 19600:2014: Compliance management systems — guidelines. International Organization for Standardization, Geneva (2014)
21. ISO/IEC 33020:2015: Information technology – process assessment – process measurement framework for assessment of process capability. International Organization for Standardization, Geneva (2015)

A Process Reference Model and A Process Assessment Model to Foster R&D&I Management in Organizations: MGPDI

Kival Chaves Weber[1(✉)], Cristina Filipak Machado[2,3], Renato Ferraz Machado[3], Ana Liddy Magalhães[4], Ana Marcia Debiasi Duarte[5], Maria Teresa Villalobos Aguayo[6], Cristiano Schwening[7], Rosane Melchionna[8], and José Antonio Antonioni[8]

[1] Senior Consultant on ICT, Quality and Innovation, Curitiba, PR, Brazil
kival_weber@yahoo.com.br

[2] CELEPAR – Companhia de Tecnologia Da Informação E Comunicação Do Paraná, Curitiba, PR, Brazil
cristina@pr.gov.br

[3] QUALITYFOCUS – Consultoria E Serviços Em Tecnologia Da Informação Ltda, Curitiba, PR, Brazil
cristina.machado@gmail.com, renato@qualityfocus.com.br

[4] UFMG – Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
analiddy@eee.ufmg.br

[5] UNOESC – Universidade Do Oeste de Santa Catarina, Chapecó, SC, Brazil
ana.duarte@unoesc.edu.br

[6] PUCP – Pontificia Universidad Católica Del Perú, Lima, Perú
mtvillalobosa@pucp.edu.pe

[7] ENGSOFT – Consultoria Em Melhoria de Processos Ltda, Florianópolis, SC, Brazil
cristiano@engsoft.com.br

[8] SOFTSUL – Associação Sul-Riograndense de Apoio Ao Desenvolvimento de Software, Porto Alegre, RS, Brazil
{rosane,jaa}@softsul.org.br

Abstract. A strategy to foster innovation in organizations consists of the adoption of a Research, Development and Innovation (R&D&I) management model. This paper describes the MGPDI model focused on Process Improvement & Assessment that is applicable to any organization independently of size, type and activity. This new model is based on: (i) requirements based on innovation best practices and Brazilian and Spanish Standards; (ii) the ISO/IEC 330xx family of standards for Process Assessment; (iii) lessons learned with the Brazilian model (MPS) for software process improvement. The MGPDI model has three components: a Process Reference Model (MR-MGPDI), a Process Assessment Model (MA-MGPDI), and a Business Model (MN-MGPDI). This paper also describes the validation of this model and its pilot implementation and assessment in three Brazilian companies. In addition to its relevance in Brazil, it has a high potential for replication in other countries.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 351–368, 2017.
DOI: 10.1007/978-3-319-67383-7_26

Keywords: Innovation management · MGPDI model · Process improvement & assessment · R&D&I management

1 Introduction

Increasing innovation is essential for organizations to survive and thrive. Essentially innovation can arise in two ways:

- Closed Innovation in which all R&D is done internally seeking to improve the competitiveness of the organization in its current market;
- Open Innovation in which the pursuit of knowledge (not just technology) in R&D is done both externally and internally in the organization seeking to increase their competitiveness either in the current market or in new innovative businesses. In a firm, Open Innovation also can be described as combining internal and external ideas as well as internal and external paths to market to advance the development of new products and services, as shown in Fig. 1 [1].

The Closed Innovation model is still used, but now prevails Open Innovation models such as:

- linear innovation without feedback as the Innovation Value Chain model [2];
- innovation based on local productive clusters and regional clusters as the ORIS - Open Regional Innovation System [3];
- innovation systems based on the Triple Helix [4];
- initiatives seeking to increase innovation through commitment of corporations with startups as the Corporate Venture model [5];

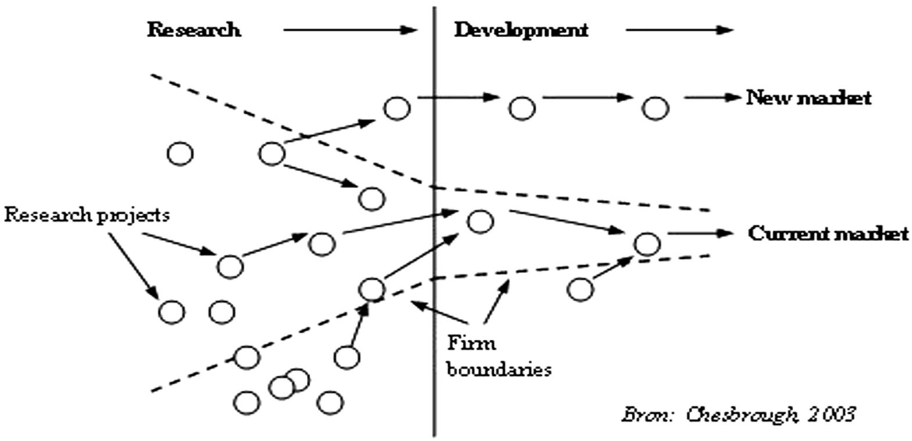

Fig. 1. Open Innovation [1]

- actions taken to foster the R&D&I Management based on Process Improvement & Assessment models as the MGPDI[1] [6, 7].

This paper describes the creation and validation of the MGPDI, including three pilot practical experiences, which is a new model aiming at Process Improvement & Assessment of the Research, Development and Innovation (R&D&I) management in organizations. Similar work exists but with a different purpose, e.g. innoSPICE™ is a standard based model for innovation, knowledge- and technology transfer. It is an evaluation procedure that can help knowledge-intense institutions generate more innovation while helping investors and research institutions optimize public funds to achieve economic added value [8].

SOFTEX – a software industry association <www.softex.br/mpsbr/> – since December 2003 has been performing successfully the MPS.BR Program aiming at creating and commercializing in Brazil and abroad a successful Software Process Improvement model named MPS (*Melhoria do Processo de Software*, in Portuguese) [9, 10].

In 2015, based on lessons learned in the MPS.BR Program and with the MPS Model for software process improvement, SOFTSUL[2] – a SOFTEX Agent – decided to transform its already existing MGPDI Methodology into the new MGPDI Model - a process model to foster the R&D&I Management in organizations. In 2015-2016 this new MGPDI Project had two goals:

- a technical goal aiming at the definition of both a PRM – Process Reference Model and a PAM – Process Assessment Model;
- a market goal aiming at performing MGPDI pilot implementation and assessment in organizations.

Thus actual research work on the MGPDI model comprehended the creation of the PRM MR-MGPDI and the PAM MA-MGPDI, which were documented in a General Guide and an Assessment Guide respectively, including the validation of the new model in three pilot implementations and assessments in Brazilian organizations. Next, Sect. 2 describes Process Improvement & Assessment and the MGPDI model, highlighting its PRM and PAM. Section 3 presents the main achieved results in pilot MGPDI implementations and assessments in three Brazilian companies. Section 4 brings our final considerations.

---

[1] MGPDI™ (*Modelo de Gestão da Pesquisa, Desenvolvimento e Inovação*, in Portuguese) is a trademark registered at INPI <http://www.inpi.gov.br/english>, owned by SOFTSUL.

[2] SOFTSUL <www.softsul.org.br> is a Brazilian private, non-profit organization created in 1994 aiming at the socio-economic development and the increase of the competitiveness of organizations ICT-intense, not only ICT companies. SOFTSUL has a Technology Development Center (CTEC) and has large experience in project coordination in the country and abroad, including Inter-American Development Bank (IDB) projects and the CONECTA 2020 project of the EU HORIZON 2020 Program.

2 Process Improvement & Assessment and the MGPDI Model

This section presents the basics of Process Improvement & Assessment based on the ISO/IEC 330xx family of standards [11], but it mainly describes the MGPDI Model highlighting its PRM and PAM, including the validation of this new process model. The section also describes succinctly two software tools to support MGPDI implementation and assessment in organizations, and the MGPDI Business Model.

2.1 Process Improvement & Assessment

ISO/IEC 33001 [12] defines:

- Process Improvement as actions taken to improve the quality of the organization's processes aligned with the business needs and needs of other concerned parties;
- Process Assessment as a disciplined evaluation of an organizational unit's processes against a process assessment model;
- Process Profile as a set of process attribute ratings for an assessed process.

Figure 2 depicts that the two-dimensional PAM – Process Assessment Model consists of a set of processes defined regarding their purpose and process outcomes (mapping a PRM – Process ReferenceModel), and a Process Measurement Framework which contains a set of process attributes related to the process quality characteristic of interest.

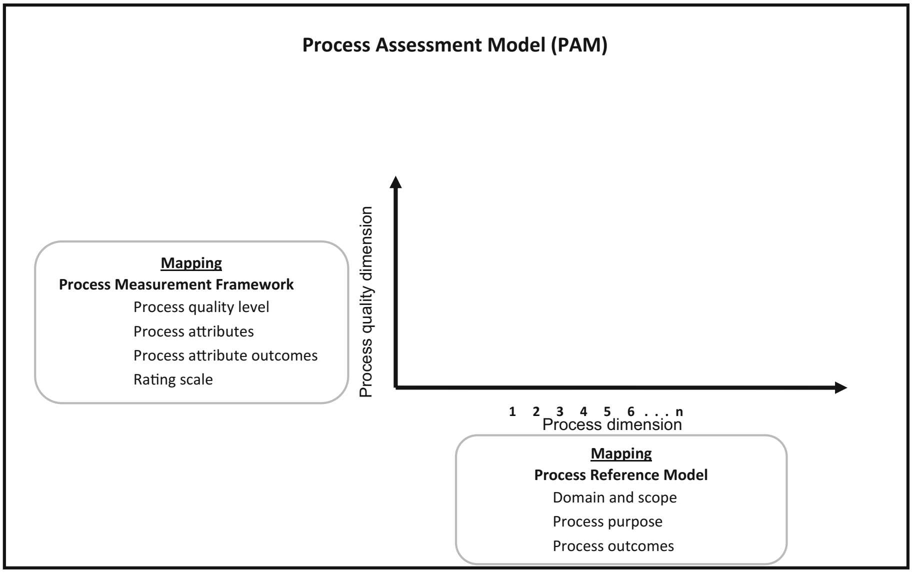

Fig. 2. Process Assessment Model relationships (ISO/IEC 33001:2015. p.12) [12]

In the process quality dimension, the process capability is determined by a set of process attribute (PA) outcomes. PAs are measurable properties of a process quality characteristic. They are accumulative and required for all processes. They may be grouped into process quality levels that may be used to characterize the process.

The assessment output includes a set of process profiles and optionally a process quality level rating for each process assessed. To maximize the repeatability, reliability, and consistency of assessments, documented evidences justifying the ratings must be recorded and retained. These evidences are in the form of assessment indicators, which typically take the form of objectively demonstrated characteristics of work products, practices and resources associated with the processes assessed. A process assessment model contains details of the assessment indicators to be used.

2.2 MGPDI™: From an Existing Methodology to a New Process Model

In 2008 SOFTSUL launched MGPDI as a methodology based on innovation best practices such as Frascati Manual – OECD [15], Oslo Manual – OECD [16], Open Innovation [1], GoInnovate [17], TRIZ – the Russian acronym for the "Theory of Inventive Problem Solving" [18], Risk Management, and Knowledge Management. The MGPDI methodology development was supported by FINEP (*Financiadora de Estudos e Projetos*, in Portuguese, or Funding Authority for Studies and Projects) <www.finep.gov.br/>.

In 2011–2012, SOFTSUL gave courses on the MGPDI Methodology in several Brazilian cities. This has been supported by CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*, in Portuguese, or National Council for Scientific and Technological Development) <www.cnpq.br/>.

In 2014 the MGPDI Methodology activities were reactivated in the framework of a Cooperation Agreement SOFTSUL-UNOCHAPECÓ <www.unochapeco.edu.br>.

In 2015, under the MGPDI Project, the existing MGPDI Methodology (*Metodologia de Gestão da Pesquisa, Desenvolvimento e Inovação*, in Portuguese) was the basis to develop the new MGPDI Model (*Modelo de Gestão da Pesquisa, Desenvolvimento e Inovação*, in Portuguese) to foster the R&D&I Management in organizations. Lessons learned from the successful Brazilian MPS Model for Software Process Improvement were very useful [9, 10].

The strategic and executive management of the MGPDI Project is based on:

- innovative ideas on the management of services [19, 20];
- the Logical Framework Approach (LFA) [21].

The MGPDI Project organizational structure comprises:

- a Project General Coordination (*CGP – Coordenação Geral do Projeto*, in Portuguese) integrated by the MGPDI stakeholders which meet half-yearly by Skype;
- a Project Performing Unit (*UEP – Unidade de Execução do Projeto*, in Portuguese) with five members, including the SOFTSUL CEO, which meet monthly by Skype;
- an Executive Project Coordination (*CEP – Coordenação Executiva do Projeto*, in Portuguese) with a senior consultant in charge;

- a Model Technical Team (*ETM – Equipe Técnica do Modelo*, in Portuguese) integrated by experts on Process Improvement & Assessment and R&D&I Management, invited by SOFTSUL, which is responsible to develop and maintain the model, and to prepare and execute people training;
- a Front Stage Collaborators network (rede CLF – Colaboradores na Linha de Frente, in Portuguese), which is composed by MGPDI implementation consultants, assessors, auditors and instructors [19, 20].

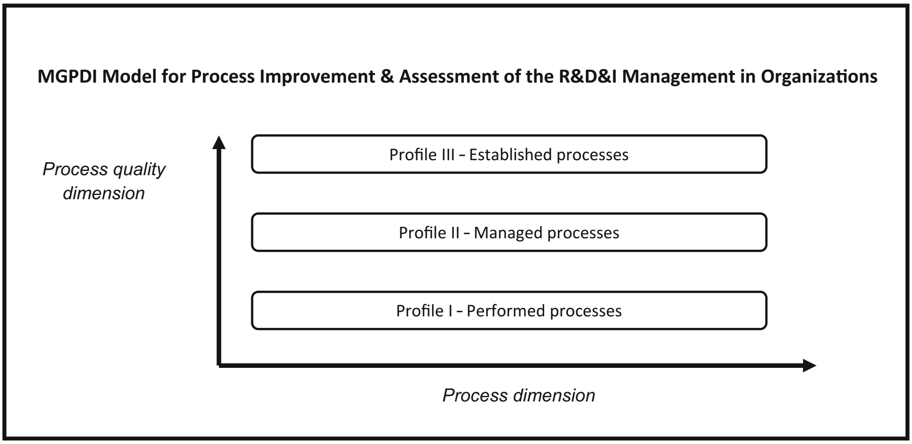

Fig. 3. MGPDI process profiles

As depicted in Fig. 3, the MGPDI Model is based on the concepts of Process Improvement & Assessment. The model has two dimensions: process dimension and process quality dimension. It comprehends three process profiles (profile I – Performed processes, profile II – Managed processes, and profile III – Established processes). This new process model was designed so that the capability in the profile I progressively provides the basis for improving the process quality level in the profiles II and III, idem from the profile II to the profile III.

In the MGPDI model the basic rule is “no one can assess the processes in the same organization where he/she had been an implementation consultant and vice versa”.

Figure 4 shows that the MGPDI Model has three components:

- the MGPDI Process Reference Model (MR-MGPDI) is a PRM based on requirements related to the innovation best practices of the existing MGPDI Methodology, and on requirements of Standards such as the Brazilian ABNT NBR 16501 [22] and the Spanish AENOR UNE 166001–166002 [23, 24] for the management of R&D&I in organizations. The PRM MR-MGPDI is described in the General Guide [25], a publicly available document at the softsul/mgpdi Website http://softsul.org.br/mgpdi/wp-content/uploads/2015/10/GUIA-GERAL-MGPDI.pdf;
- the MGPDI Process Assessment Model (MA-MGPDI) is a PAM based on the family of standards ISO/IEC 330xx [12] for process assessment. It is described in

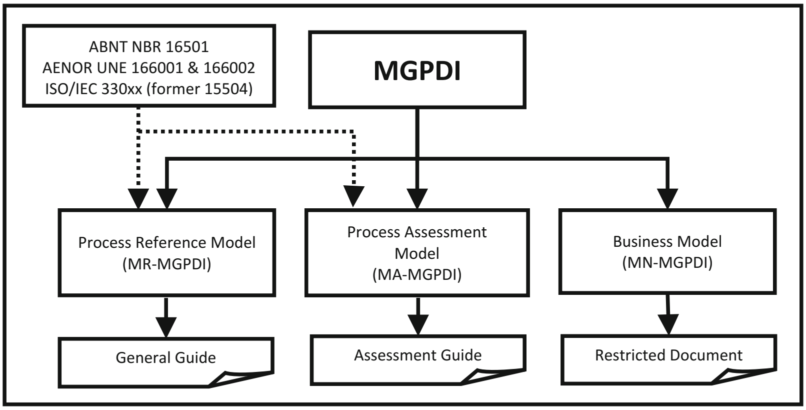

Fig. 4. MGPDI components

the MGPDI Assessment Guide [26], a publicly available document at the softsul/mgpdi Website http://softsul.org.br/mgpdi/wp-content/uploads/2015/10/GUIA-DE-AVALIAÇÃO-MGPDI.pdf;

- the MGPDI Business Model (MN-MGPDI) with the business rules to commercialize MGPDI courses/exams, implementations and assessments, which is described in a restricted document published at the softsul/mgpdi Website but only available to whom that has an username and a password to access it.

2.3 PRM – Process Reference Model (MR-MGPDI – *Modelo de Referência de Processos Do MGPDI*, in Portuguese)

The purpose of a PRM – Process Reference Model is to define a set of processes that can collectively support the primary aims of a community of interest. According to ISO/IEC 33004 [13], a PRM must contain:

- a declaration of the domain of the PRM;
- a description of the relationship between the PRM and its intended context of use;
- descriptions of the processes within the scope of the PRM;
- a description of the relationship between the processes defined within the PRM.

The domain of the MGPDI Process Reference Model (MR-MGPDI) is the management of Research, Development and Innovation (R&D&I). Its community of interest comprehends both practitioners, instructors, implementers and assessors of the MGPDI model, and students, professors and researchers interested in the theme of Process Improvement & Assessment in the Academia.

The MGPDI General Guide provides a general description of the MGPDI model and details both the Process Reference Model (MR-MGPDI), and the common definitions that are necessary to its understanding and application [25]. Table 1 shows that the MGPDI Process Reference Model (MR-MGPDI) comprehends three areas, 13 processes and their respective purposes. For each process this PRM also defines the process outcomes (not presented here due to limitation of space).

Table 1. MR-MGPDI areas, processes and purposes.

Area	**Process**	**Purpose (To establish and maintain…)**
Innovation	1.GIO – Innovation Management	…the context and the qualification of innovative ideas.
	2.GPE – Research Management	…new knowledge from research in innovation.
	3.EIN – Innovation Strategy	…a strategic innovation plan and to define a set of significant techniques and tools to support the management of innovative business.
	4.GCI – Innovation Cycle Management	…the processes related to the management of the innovation cycle.
	5.GPP – Intellectual Property Management	…activities regarding patents, transfers and records on innovation.
Management	6.GPI – Innovation Project Management	…each innovation project.
	7.GRI – Risk Management	…the uncertainties and risks that may occur during the project.
	8.GPO – Portfolio Management	…innovations and projects that are necessary, sufficient and sustainable in order to meet the strategic objectives of the organization.
Support	9.GOV – Governance	…governance initiatives creating a favorable environment for innovation in organizations.
	10.GIN – Indicator Management	…the indicators that can measure and assess innovation management in organizations.
	11.GCO – Configuration Management	…the integrity of versions of items related to the process work products.
	12.GQU – Quality Management	…a set of definitions and factors related to the quality of process work products.
	13.GMU – Change Management	…activities and responsibilities to ensure the integrity of the model allowing that suggestions for improvements and exception treatments can be implemented.

As you can see in Table 1, there is no a process neither a purpose (nor an outcome) explicitly related to Open Innovation. But there are several Open Innovation best practices related to these areas, processes, purposes and outcomes that can be used by the organizations. Some of these best practices are used as assessment indicators. For instance, those related to the Idea Generation through collaboration. A report identified 11 Open Innovation best practices based on research into how leading companies are tapping external sources of expertise. These best practices are categorized into four principle areas: strategies, roles, processes, and measurement/improvement [27].

The top three best practices highlighted by the report are:

- establishing a central and dedicated group to drive Open Innovation (75% of the surveyed best practice companies have staff members specifically dedicated to pursuing and deploying Open Innovation strategies);
- partnering broadly across a variety of external and internal organizations;
- inviting participation in Open Innovation via experiences.

Among the other best practices revealed by the report are:

- position your organization to build and manage key relationships;
- embrace broad and specific scouting for new ideas;
- use change management to drive commitment to Open Innovation.

2.4 PAM – Process Assesssment Model (MA-MGPDI – *Modelo de Avaliação de Processos Do MGPDI*, in Portuguese)

According to ISO/IEC 33001 [12], a PAM – Process Assessment Model is a model suitable for the purpose of assessing a specified process quality characteristic, based on one or more process reference model. According to ISO/IEC 33020 [14], a PAM is based in a set of assessment indicators that:

- explicitly address the purpose and process outcomes of a PRM;
- demonstrate the achievement of the process attributes within the scope of the PAM;
- demonstrate the achievement (where relevant) of the process quality levels within the scope of the PAM.

The community of interest of the PAM MA-MGPDI is the same as the PRM MR-MGPDI, but it mainly includes those interested in process assessment based on the ISO/IEC 330xx family of standards [11]. The MGPDI Assessment Guide describes this assessment process detailing its activities, tasks, tools, artifacts, assessment participants, process quality levels and rating scale [26]. Table 2 summarizes the PAM MA-MGPDI profiles, processes and process attributes. The new processes and process attributes in Profile II and II are bolded.

Remark. Based on the ISO/IEC 33020 [14], the capability of the MGPDI model could still be expanded with the addition of the following process attributes: "PA 4.1–4.2: Quantitative analysis and quantitative control of processes" creating the profile IV – Predictable process; "PA 5.1–5.2: The innovation processes are being optimized" creating the profile V – Optimizing innovation process. Although we have good knowledge of these two higher profiles from the lessons learned with the Brazilian MPS model [9, 10], we decided not include them now due to the adoption of a 'divide to conquer' strategy.

As shown in Fig. 5, the MGPDI Process Assessment Model (MA-MGPDI) also defines a MGPDI Assessment process which comprehends four steps: 1 – Enable assessment, 2 – Remote pre-assessment, 3 – Visit the organizational unit, and 4 – Conclude assessment, that make up a set of activities to be performed during the assessment in each organization as well as the key outputs that should be generated and informed. The MGPDI Assessment process begins with the notice of an assessment and

Table 2. PAM MA-MGPDI profiles, processes and process attributes

Profile	Process	Process Attributes (PA)
I - Performed processes	GIO – Innovation Management GPI – Innovation Project Management GIN – Indicator Management GOV – Governance GPE – Research Management	PA 1.1: Process is performed
II – Managed processes	GIO – Innovation Management GPI – Innovation Project Management GIN – Indicator Management GOV – Governance GPE – Research Management **GRI – Risk Management** **EIN – Innovation Strategy** **GCI – Innovation Cycle Management** **GMU – Change Management** **GCO – Configuration Management** **GQU – Quality Management** **GPO – Portfolio Management**	PA 1.1: Process is performed **PA 2.1-2.2: Process performance and work products are managed**
III – Established processes	GIO – Innovation Management GPI – Innovation Project Management GIN – Indicator Management GOV – Governance GPE – Research Management GRI – Risk Management EIN – Innovation Strategy GCI – Innovation Cycle Management GMU – Change Management GCO – Configuration Management GQU – Quality Management GPO – Portfolio Management **GPP – Intellectual Property Management**	PA 1.1: Process is performed PA 2.1-2.2: Process performance and work products are managed **PA 3.1-3.2: Process is defined and deployed**

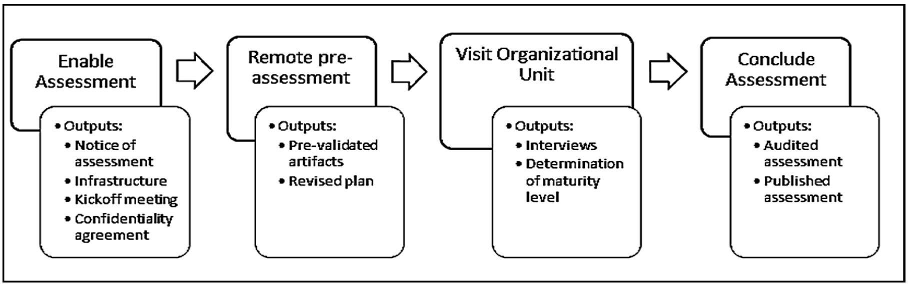

Fig. 5. Steps of the MGPDI Assessment process

ends when the organizational unit returns the feedbacks about the implementation and assessment processes, so the assessment results can be published at softsul/mgpdi Website (http://softsul.org.br/mgpdi/?page_id=56&lang=pb).

Each MGPDI assessment has a validity period of three years. During this term there are two annual follow-ups, respectively at the end of first and second years.

2.5 MGPDI Support Tools

There are two online software tools to support respectively the MGPDI implementation and assessment in organizations.

- **SGPDI Implementation Tool**
  In the MGPDI there is a software tool named SGPDI (*Sistema de Gestão da Pesquisa, Desenvolvimento e Inovação*, in Portuguese) to provide a work environment integrated with the innovation process and to support its implementation. It is available at (http://www.softsul.org.br/mgpdi_base/).
  This software includes the steps of identification, design, and validation of an innovative idea and assists in the development and implementation of innovation projects, including risk monitoring. It also provides user's authentication and secure access control. All activities implemented in organizations that use the MR-MGPDI have their data stored safely in the SGPDI system.
- **Appraisal Assistant – AA**
  The software tool 'Appraisal Assistant - AA' was configured to support the Assessment Team in MGPDI assessments, aiming at measuring the process outcomes and the process attribute outcomes in organizations. This tool has an approach based on the validation of pieces of evidences, visualization of processes by profile, and generation of assessment reports. It is available at (https://www.sqi.griffith.edu.au/AppraisalAssistant/about.html).
  Thus it is possible to make reviews based on the MGPDI model requirements and to identify weaknesses in the implemented innovation processes, providing assessment feedbacks that point to improvement opportunities in each assessed organization. The assessment feedbacks of all assessed organizations also contributes to improve the assessment process and the MGPDI Model.

2.6 Validation of the New MGPDI Model

The PRM MR-MGPDI was conceived by the ETM-MGPDI (Model Technical Team) and a draft version of the MGPDI General Guide was available in October 2015 to support the three pilot implementations of the MGPDI Profile I – Performed processes from November 2015 and June 2016.

The PAM MA-MGPDI was created by the ETM-MGPDI and a draft version of the MGPDI Assessment Guide was available in September 2016 to support the three pilot assessments of the MGPDI Profile I – Performed processes from October and December 2016.

The main conclusion of this two-years work is that the MGPDI model was validated as adequate to its purpose, which is to contribute effectively to foster the R&D&I management in organizations.

In December 2016 there were published at the softsul/mgpdi Website the first version of the MGPDI General Guide [25] and MGPDI Assessment Guide [26].

Some lessons learned were raised by the team of MGPDI implementation consultants and assessors:

- some improvements should be made in the MGPDI General Guide to clarify some process outcomes and overlapping;
- the SGPDI Implementation tool needs to be interface improved to increase the usability;
- a map between the Process & Process Attribute outcomes and the SGPDI Implementation tool can be done to facilitate the implementation and assessment;
- training in the use of the AA tool should be carried out and several descriptive fields need to be standardized to generate standards reports;
- the remote pre-assessment has promoted interaction between the assessors and the organizational unit team. This mechanism proved to be efficient and economical without lose the quality of the assessment.

2.7 Business Model (MN-MGPDI – *Modelo de Negócio Do MGPDI*, in Portuguese)

The Business Model MN-MGPDI is the component of the MGPDI model related to its trade practices and culture. It only concerns to the purpose of commercialization (*Go2Market*) of the MGPDI model, including marketing, value proposition, target customers, business process, suggested prices (reference values), offerings, strategies, infrastructure, organizational structures, operational processes and policies.

Business rules are described in the MGPDI Business Model (MN-MGPDI) both to support the commercialization of MGPDI implementations/assessments/annual follow-ups, and the offering of MGPDI courses/exams.

The MN-MGPDII comprises:

- a cooperated Business Model (MNC-MGPDI) suitable for groups of small and medium enterprises (SMEs) that want to share costs and part of the efforts;
- a specific Business Model (MNE-MGPDI) suitable for private and public organizations that prefer exclusive attendance.

3 Implementing and Assessing the MGPDI Model in Organizations

This section reports three pilot practical experiences of MGPDI implementations and assessments in organizations in the Southern Region of Brazil.

3.1 Companies that Adopted the MGPDI Model in 2015-2016 (Pilot A – MGPDI Profile I)

In 2015–2016 the new MGPDI model was created and it was documented in beta versions of the MGPDI General Guide and the MGPDI Assessment Guide. In this period, under SOFTSUL coordination, the first people were trained to act as implementation consultants and assessors in organizations that want to adopt the MGPDI model. In 2016, a beta version of the course C3 – MGPDI Assessment was created to begin the training of new assessors. The implementation of the MGPDI model, profile I, in these three companies has taken from November 2015 and June 2016.

Table 3 summarizes some demographic characteristics of the organizations that adopted the MGPDI Model in the three MGPDI implementation and pilot projects.

Table 3. Organizations that adopted the MGPDI Model in 2015-2016 (Pilot A – MGPDI Profile I)

Company	**A**	**B**	**C**
Location	Porto Alegre-RS, Brazil	Porto Alegre-RS, Brazil	Chapecó-SC, Brazil
Industry	Software	ICT	Software
Public or private	Private	Private	Private
Small, medium or large enterprises *	Medium	Small	Small
Approx. company annual revenue * (USD 1 = R $ 3)	Greater than USD 1,2 million and less than or equal USD 100 million	Greater than USD 120 thousand and less or equal to USD 1,2 million	Greater than USD 120 thousand and less or equal to USD 1,2 million
Approx. number of staff	38	31	23
Product portfolio	**DRS-Audience** - Tool for recording audio, video and text of court hearings **DRS-Plenary** - Plenary session recording tool, audio and video distribution with shorthand management	**Self-Service Terminals** - Product with modern design that adapts to various types of environments **Thin Client** - Compact CPUs with integrated network processing **Mini PC** - Solution for applications that need a	**Slim ERP** - Solution with a focus on solving all problems in a single system **Middle ERP** – Solution that works by creating control parameters for: taxes, processes and use **High End ERP** – Full ERP based on simplicity and customer

(*continued*)

Table 3. (*continued*)

Company	A	B	C
	DRS-Inquiry - Tool for audiovisual recording of testimony and expertise in police investigations	better balance between CPU performance and multimedia **POS Fusion Touch** - Screen with Touch SAW technology that provides usability in harsh environments (kitchens, dusty environments, etc.) where other Touch technology is inefficient **Digital Signage (DS)** - Products for total visual communication with the public of commercial companies	focus with features beyond what a system presents

* *Remark*. BNDES criteria (http://www.bndes.gov.br/wps/portal/site/home/financiamento/guia/quem-pode-ser-cliente).

Table 4 summarizes the results of the process assessments in these organizational units. All the organizations achieved the Profile I – Performed processes.

Table 4. Assessment Results of the Organizational Units

Process	**Process Purpose (rating)**			**Process Attribute AP 1.1 (rating)**			**Final Result**		
	Company			**Company**			**Company**		
	A	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
GIO – Innovation Management	F	F	F	F	F	F	SATISFIED Profile I	SATISFIED Profile I	SATISFIED Profile I
GPI – Innovation Project Management	F	F	F	F	F	F			
GIN – Indicator Management	F	F	F	F	F	F			
GOV – Governance	F	F	F	F	F	F			
GPE – Research Management	F	F	F	F	F	F			

Remark. Rating used: F: Fully Achieved, L: Largely Achieved, P: Partially Achieved, N: Not Achieved

Table 5 highlights the outcomes of two indicators related to the GIN – Indicator Management process assessed in the three companies. These innovation indicators did not exist in these companies before the adoption of the MGPDI model.

Table 5. MGPDI Assessed Indicators

Assessed indicator	**Company A**	**Company B**	**Company C**
Number of new Ideas	–	19	60
Index of Ideas that become projects	14%	31,5%	51,6%
Index of Innovation Projects completed in time	80%	–	–

3.1.1 Company A

In this company the MGPDI assessment comprised a remote pre-assessment on Nov 22, 2016, and a visit to the Organizational Unit on Dec 9, 2016. Company A is a well-structured company in the area of Quality, with certifications such as ISO 9000, CMMI and MPS.BR. Thus, the profile I of the MGPDI model was incorporated into the existing Quality Management system, which deals with Process Improvement & Assessment and already defined innovation processes, named differently and with other approaches.

The company IT Director was the assessment sponsor and he sent the following testimony as a feedback: "Our company always valued and guaranteed the quality of its products and services based on the main Market Certifications. Thus, starting from this premise, the MGPDI model emerged both to continue the improvement of our processes and now to assess and certify our technological differential, which are the R&D and Innovation projects. We consider this new model to be very complete and productive, from implementation to assessment, and now we are motivated and focused on its adoption aiming at continuous improvement, which will bring relevant results to our company."

3.1.2 Company B

In this company the MGPDI assessment comprised a remote pre-assessment on Aug 5, 2016, and a visit to the Organizational Unit on Nov 9, 2016. In Company B. two employees were selected to be the Innovation Leaders, under the supervision of the Quality Director. The fact of having two Innovation Leaders favored the Governance process (GOV) and brought a particular aspect about the distribution of roles and responsibilities to the implementation and assessment.

The company CEO was the assessment sponsor and he sent the following testimony as a feedback: "Our organization was born from the development of a bank check printer. At that time this was an innovation. Since then, throughout its 26 years, the company has been evolving by the development of new products and it has in its DNA a very strong bias for innovation. However, there was a lack of a methodology or model that could organize the innovation management and control all stages of the innovation processes, from the idea to its implementation and assessment, including measurements through indicators. For a long time, we have looked for management models that could meet our need. Knowing the MGPDI, I realized that this model had everything we were needed. The best was to realize that, even in the course of its implementation and assessment in our company, this model not only answered us positively but it exceeded all our expectations."

3.1.3 Company C

In this company the MGPDI assessment comprised a remote pre-assessment on Oct 10, 2016 and a visit to the Organizational Unit on Oct 31, 2016 - so this was the first organization to achieve the MGPDI "certification". Company C had already dealt with innovation processes, but these were not organized and were not managed, and had not been able to measure the innovation improvements.

A company Director was the assessment sponsor and she sent the following testimony as a feedback: "For 17 years, since the founding of our company, one of our values has been innovation. However the challenge of promoting innovation within companies is to create a culture and practice it with employees. So we saw in the MGPDI model a way of organizing innovation management. We adopted this new model and we already perceived great differences because we started to measure employees' activity in a more efficient way, bringing new ideas that were latent within the company. Also, we saw the possibility of interacting with the market, involving customers so that they live with us this new way of managing companies. The adoption of the MGPDI model (profile I) is being very effective, but we know that there is a road to be covered from the profile I to higher ones."

4 Final Considerations

This paper described the MGPDI Model that was created in Brazil in 2015-2016 to conduct Process Improvement & Assessment aiming to foster the management of the R&D&I in organizations, independently of their size, type and activity – not only ICT companies, highlighting:

- its PRM – Process Reference Model (MR-MGPDI) and PAM – Process Assessment Model (MA-MGPDI), including the validation of the MGPDI Model;
- three pilot implementations and assessments in Brazilian companies of the software & ICT industries.

Thus this research work (creation and validation of the MGPDI model aiming at fostering the R&D&I management in organizations), including its pilot practical experience (implementations and assessments of the new model in three Brazilian organizations), has contributed to the body of knowledge in Process Improvement & Assessment.

In addition to its relevance to Brazilian organizations, this new model has a high potential for replication in other countries - firstly in Portuguese and Spanish speaking countries, and later in English-speaking countries as the model is re-written in English.

The new MGPDI model was conceived and developed as a whole but a limitation is that only was detailed the Profile I – Performed processes. The Profile II – Managed processes and the Profile III – Established processes will be detailed in 2017 based on lessons learned until now.

As next steps, the 2017 MGPDI Annual Plan foresees:

- a complete revision of the MGPDI General Guide to detail both Profile II and Profile III;

- the consequent fixes in the MGPDI Assessment Guide;
- the development of a beta version of the MGPDI Implementation Guide aiming at providing non-prescriptive guidelines for the Implementation Consultants;
- the training of 12 (twelve) people - with good experience in Process Improvement & Assessment both on the Brazilian MPS model and the CMMI - aiming to qualify them as new MGPDI Instructors and Implementation Consultants.

But the main challenge in 2017 is to begin an initial offer (*Go2Market*) of:

- the MGPDI model (Profile I – Performed processes) in the marketplace, both in Portuguese in the five Brazilian regions and in Spanish in Latin America and the Caribbean (LAC) countries, seeking totalize over 12 (twelve) MGPDI implementations and assessments in organizations;
- course and exam C1/P1 – MGPDI Introduction both face-to-face and online distance learning.

Last but not least we hope that this paper can contribute to a better understanding of R&D&I management in organizations, either by practitioners, instructors, implementers and assessors of this new process model, or at the Academia by students, professors and researchers interested in Process Improvement & Assessment, and also to foster the diffusion of the SPICE-based process assessments.

References

1. Chesbrough, H.: Open Innovation: The New Imperative for Creating and Profiting from Technology. HBS Press, Harvard (2003)
2. Hansen, M.T., Birkinshaw, J.: The Innovation Value Chain. Harvard Business Review – HBR, June issue (2007)
3. Belussi, F., Sammarra, A., Sedita, S.R.: Learning at the boundaries in an ‘Open Regional Innovation System – ORIS’: A focus on firms’ innovation strategies in the Emilia Romagna life science industry. Elsevier B.V. (2010). doi:10.1016/j.respol2010.01.014. www.elsevier.com/locatel/respol
4. Ranga, M., Etzkowitz, H.: Triple Helix Systems: An Analytical Framework for Innovation Policy and Practice in the Knowledge Society. Ind. High. Educ. **27**(3), 237–262 (2013). doi:10.5367/ihe.2013.0165
5. Weinblein, T., Chesbrough, H.: Engaging with startups to enhance corporate innovation. Calif. Manag. Rev. **57**(2), 66–90 (2015). http://www.jstor.org/stable/10.1525/cmr.2015.57.2.66
6. Weber, K.C., Antonioni, J.A., Melchionna, R., Pereira, R., Toniazzo, J.C., Schwening, C., Machado, C.F., Keglevich, P., Herbert, J., Villalobos-Aguayo, M.T.: MGPDI: Modelo de gestión de la I+D+i en las organizaciones. In: I Congreso Internacional de Gestión de la Innovación, 28 septiembre–02 octubre, PUCP, Lima, Perú (2015)
7. Weber, K.C., Melchionna, R., Machado, C.F., Machado, R.F.: Implementações e Avaliações Piloto Usando o Modelo MGPDI para Gestão da Pesquisa, Desenvolvimento e Inovação nas Organizações. In: International Symposium on Project Management, Innovation and Sustainability – V SINGEP, 20–22 Nov, UNINOVE, São Paulo-SP, Brasil (2016)
8. innoSPICE. Innovation, iso15504, SPICE. Innovation Capability Center of Bremen University, Germany. http://innospice.ning.com/

9. Montoni, M., Rocha, A.R., Weber, K.C.: MPS.BR: a successful program for software process improvement and practice. Softw. Process Improve. Pract. **14**, 289–300 (2009). Wiley InterScience. www.interscience.wiley.com. doi:10.1002/spip.428
10. Kalinowski, M., Weber, K.C., Santos, G., Franco, N., Duarte, V., Travassos, G.H.: Software Process Improvement Results in Brazil Based on the MPS-SW Model. In: SQP ASQ – American Society for Quality, vol. 17(4) (2015). www.asq.org
11. ISO/IEC 330xx family of standards. Information technology: Process assessment. New family ISO/IEC 330xx updates ISO/IEC 15504 series of standards (2015)
12. ISO/IEC 33001. Information technology – Process assessment – Concepts and terminology (2015)
13. ISO/IEC 33004. Information technology – Process assessment – Requirements for process reference, process assessment and maturity models (2015)
14. ISO/IEC 33020. Information technology – Process assessment – Process measurement framework for assessment of process capability (2015)
15. OECD. Frascati Manual: Proposed Standard Practice for Surveys on Research and Experimental Development, 6th edn. (2002)
16. OECD. Oslo Manual: Guidelines for Collecting and Interpreting Innovation Data, 3rd edn. (2005)
17. Papageorge, A.: GoInnovate!: A Pratical Guide to Swift, Continual and Effective Innovation. GoInnovate! Publishing, San Diego (2004)
18. Altshuller, G.: Innovation Algorithm. Worcester: Technical Innovation Center, 1st Russian edn. 1969 (1999)
19. Teboul, J.: Service is Front Stage: Positioning Services for Value Advantage. Palgrave McMillan, New York (2006)
20. Teboul, J.: Serviços em Cena: O Diferencial que Agrega Valor ao seu Negócio. INSEAD, CNI/IEL (2008)
21. NORAD: The Logical Framework Approach: Handbook for objectives-oriented planning, 4th edn. ISBN 82-7548-160-0 (1999)
22. ABNT NBR 16501: Diretrizes para sistemas de gestão da pesquisa, do desenvolvimento e da inovação (PDI). Norma Brasileira (2011)
23. AENOR UNE 166001. Gestión de la I+D+i: Requisitos de un proyeto de I+D+i. Norma Española (2006)
24. AENOR UNE 166002. Gestión de la I+D+i: Requisitos del sistema de gestión de la I+D+i. Norma Española (2014)
25. Guia Geral do MGPDI, December 2016. http://softsul.org.br/mgpdi/wp-content/uploads/2015/10/GUIA-GERAL-MGPDI.pdf
26. Guia de Avaliação do MGPDI, December 2016. http://softsul.org.br/mgpdi/wp-content/uploads/2015/10/GUIA-DE-AVALIAÇÃO-MGPDI.pdf
27. APQC: Open Innovation: Enhancing Idea Generation Through Collaboration. In: ideaCONNECTION report (2013). https://www.ideaconnection.com/blog/2013/08/open-innovation-best-practices-a-new-report/

SPI and Gamification

Gamification for Improving IT Service Incident Management

Elena Orta(✉), Mercedes Ruiz, Alejandro Calderón, and Nuria Hurtado

Department of Computer Science and Engineering, University of Cadiz, Avda. de la Universidad de Cádiz, 11519 Puerto Real (Cádiz), Spain
{elena.orta,mercedes.ruiz,alejandro.calderon, nuria.hurtado}@uca.es

Abstract. Support groups' agents are a very important element of IT service incident management process, and to increase their motivation and commitment, to improve their skills and to modify their behaviors is fundamental for improving the process and meeting the business objectives. Gamification, the application of game elements in non-game contexts to modify and influence the behavior of the people, is helpful in this field. This paper introduces a method for gamifying the incident management process that is based on the gamification method proposed by Werbach and Hunter and ITIL incident management process. To illustrate the usefulness of the method proposed an example of use is also presented.

Keywords: IT service incident management · Gamification · ITIL · Process improvement

1 Introduction

IT Service Management (ITSM) is focused on the implementation and management of quality IT services through an appropriate blend of people, processes and implementation technology [1]. Since people are one of the main elements of ITSM, change initiatives require important effort to modify the people's behavior. Attitude towards change, to increase the motivation and commitment, and to acquire and practice new behaviors and skills are crucial for the success of the process improvement initiatives [2].

The incident management process is a necessary and very important process of IT support organizations. Its main aim is to manage and restore the normal service interruption as rapidly as possible, and with the minimal impact on the business [3]. Normally these organizations are structured in several levels of support groups whose agents perform the necessary tasks to resolve the incidents. Much of the working time of the agents is spent answering redundant questions and conducting monotonous and repetitive tasks. Introducing game elements into the incident management process, it is possible to add an element of excitement to their tasks and a motivation layer, and the process becomes rewarding for the agents [4]. Besides, gamification gives agents something that their job is inherently lacking: a sense of achievement. The agents' sense of achievement boots because gamification makes their improvements measurable,

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 371–383, 2017.
DOI: 10.1007/978-3-319-67383-7_27

it enables them to obtain a reputation for the quality of their work, and it gives them rewards for improvement.

The gamification of the incident management process not only make the workday more fun and interesting, it also helps agents adapt their behaviors for improving the process by being clear about the goals to achieve and how they will be rewarded in return [5, 6]. Besides, gamification can increase the motivation and commitment of the agents, encourage their participation, and improve their productivity, skills, performance and engagement in the process [5, 7–9].

Despite of the benefits of applying gamification for the incident management process improvement, there is currently very little research in this field. The review of published research articles conducted in this paper shows that actual works present initial proposals as a step toward gamification, or examples of incident management process tool gamified. However, none of them propose a concrete gamification method that helps organizations define the problems to solve, identify the agents behaviors to promote and determine the game elements to include in a gamification project.

The main contributions of this paper are the following: (a) a method to gamify the incident management process that is based on both the gamification method proposed by Werbach and Hunter [5], and ITIL incident management process [3]; and (b) an example of use of the method proposed.

The rest of the paper is structured as follows. Next section summarizes the ITIL incident management process and introduces a study of published research works in the field of incident management gamification. Section 3 describes the method proposed to gamify the incident management process. Section 4 introduces an example of use of the method. Finally, Sect. 5 contains our conclusions and further works.

2 Incident Management Process and Gamification

2.1 ITIL Incident Management Process

Information Technology Infrastructure Library (ITIL) is globally accepted as de facto standard for ITSM [10, 11]. ITIL incident management process is focused on "*to restore normal service operation as quickly as possible and minimize the adverse impact on business operations, thus ensuring that agreed levels of service quality are maintained*" [3]. The main objectives of the process are to ensure that standardized methods and procedures are used for an efficient incident resolution, to increase visibility and communication of incidents to business and IT support staff, to align incident management activities with business objectives, and to maintain user satisfaction with the quality of IT services [3].

The main activities of the ITIL incident management process are the following [3]:

1. *Incident logging*: opening a new incident and record the information needed (service desk's agents).
2. *Incident categorization*: determining and recording an adequate incident category (service desk's agents).

3. *Incident prioritization*: determining and recording a suitable incident priority normally taking into account both the urgency and the business impact of the incident (service desk's agents).
4. *Initial diagnosis*: performing an initial incident diagnosis to try to discover all the symptoms of the incident and to determine what has gone wrong and how to correct it (service desk's agents).
5. *Incident escalation*: escalating the incident for further support group if the agent assigned is unable to resolve the incident or the target times for resolution have been exceeded (support groups' agents).
6. *Investigation and diagnosis*: investigating and diagnosing what has gone wrong (support groups' agents).
7. *Resolution and discovery*: applying and testing the incident resolution identified, and updating the incident record with actions taken (support groups' agents).
8. *Incident closure*: checking that the incident is fully resolved and the users are satisfied (service desk's agents).

The incident management process has objectives that are specific to the organization and are specified in the Service Level Agreements (SLAs) that the organization agrees with its customers. Many factors influence the process outcomes and the SLA fulfilment, such as the efficiency of the support groups, the time required to perform the process activities, and the incident management procedures and methods adopted in the organization.

2.2 Incident Management Process Gamification: Related Works

Gamification is the application of game elements in non-game contexts to modify and influence the behavior of the people [5] and solve business problems [12]. Given the high influence of the support groups' agent behavior on the incident management process performance, change initiatives need that agents modify their behaviors. Gamification is helpful in this field because it promotes in agents the behavior necessary for meeting the established process performance targets. Besides, agents engage and become focused in situations with defined goals, a measurable sense of progress leading to the goals, a notion of status as a result of achieving the goals, and meaning rewards for reaching the goals [6].

This section introduces a review of published research papers that present works focused on the incident management process gamification. The search string (gamification OR gamify OR gamifying OR funware) AND ("Service Desk" OR "Help Desk" OR "IT incident" OR "IT incidence" OR "service incident" OR "service incidence") used in databases SCOPUS, IEEE Xplorer, Web of Science and Springer Link recovered very few papers that have been studied to identify the main issues addressed. The main purpose of the works found are described below.

On the one hand, [13] introduces a conceptual framework for implementing service supports which integrates ITIL best practices and a gamification model to improve the motivation of service supports' employees. The conceptual frameworks activities are the following: (a) *Feasibility study*; (b) *Analysis* (study the current process state, the people involved in the process, and the desired targets); (c) *Design* (identify and prepare services, identify points, prepare game elements, and prepare flow chart system); (d) *Testing* (validate that the design is accepted); and (e) *Implementation* (build the service system, combine game elements, services and points, test the gamified-service, install IT infrastructure, create user guide, and workshop); and (f) *Monitoring and evaluation* (ask users' opinion, review feedback and gap analysis). In [14], the authors of [13] introduce an application case of the conceptual framework to develop a gamified service support prototype. The prototype uses points as core elements, and difference two types of points: redeemable points (RP) and experience points (XP). A service desk analyst gets points (RP and XP) when he closes an incident within the established resolution time. In contrary, he gets XP points because he gains experience with every action conducted, but he loses RP points. The prototype also includes leaderboards to show service desk analysts with highest XP and RP, and create a competition environment.

The authors of [15] also present an initial gamification model to improve the service desk performance. The game elements considered in this gamification model are similar to those considered in the previous work [14]: rewards and rankings. When an incident is solved, the service desk operators that worked on the incident receive rewards depending on (i) the difficult and relevance of the incident and (ii) an unprecedented problem is solved. Rewards can assume different aspects: (a) internally to the gamification system (points, levels and badges), and (b) externally to it (social recognition, reputation, and job and salary promotion). The gamification model also includes a ranking that shows the points, levels, progress, badges and accomplishments achieved for each service desk operator. [16] introduces an user-support worker's activity model that enhances the interactivity of the employees of cyber-infrastructures with the incidents. As a step toward gamification, the model provides a point count system to arouse the interest of the employees and to stimulate them to solve the incidents within the established times. Moreover, employees get points if the incident solution is appreciated by users, and obtain additional recognition (promotions, extra remuneration, free hours or holidays, etc.) if they put additional effort to solve the incidents.

On the other hand, the authors of [17] describe the integration of gamification and persuasion mechanisms as incentives into the current HP Service Manager tool. Several game elements (points, badges, prizes, rewards, leaderboards, progress bars, rankings, achievements and reinforcement) have been incorporated in the HP system to meet the following targets: (1) to improve the quantity and quality of the knowledge exchange in the groups; (2) to increase the case record quality (cited count and helpfulness), and (3) to enhance the customer satisfaction.

Finally, in [18] a conceptual framework for improving ITSM processes which integrate simulation modeling techniques and gamification is presented. It is focuses on (i) building a process simulation model, and (ii) gamifying the simulation model experimentation to engage IT managers and drive their behavior through model simulations.

The results of the literature review conducted notice that actual research in the context of incident management gamification is still in an incipient state. Some of the works studied introduce initial gamification models or conceptual frameworks which propose to integrate some game elements (mainly points, rewards and leaderboards) into the process. These works propose that agents receive rewards when they meet the following objectives: (1) closing an incident within the established times [14–16];

(2) solving an unprecedented problem [15]; and (3) the incident solution is well valued by the users [16]. The issues addressed in [17] and [18] are different with respect the previous works. On the one hand, [17] describes how the HP Service Manager tool has been gamified. On the other hand, [18] proposes a conceptual framework to gamify the experimentation of ITSM process simulation models.

The referenced works introduce initial proposals of incident management process gamification. They indicate what game elements can be included and what agents' behavior can be promoted in an incident management process gamification project. However, none of them propose a concrete method to design a suitable gamification strategy aligned with the business objectives defined in the organization.

3 A Method for Gamifying the Incident Management Process

Although the technology is a very important aspect in a gamification project, many projects fail because an adequate gamification method that helps the organization (i) define the problem to solve, (ii) identify the behaviors to promote, and (iii) determine the game mechanics most suitable to engage and motivate the people involved in the system [19] is not applied.

This section introduces a novel method for gamifying the incident management process that has been developed by the authors of this paper. Figure 1 shows the activities of the method which is based on both the gamification method proposed by Werbach and Hunter [5] and ITIL incident management process [3].

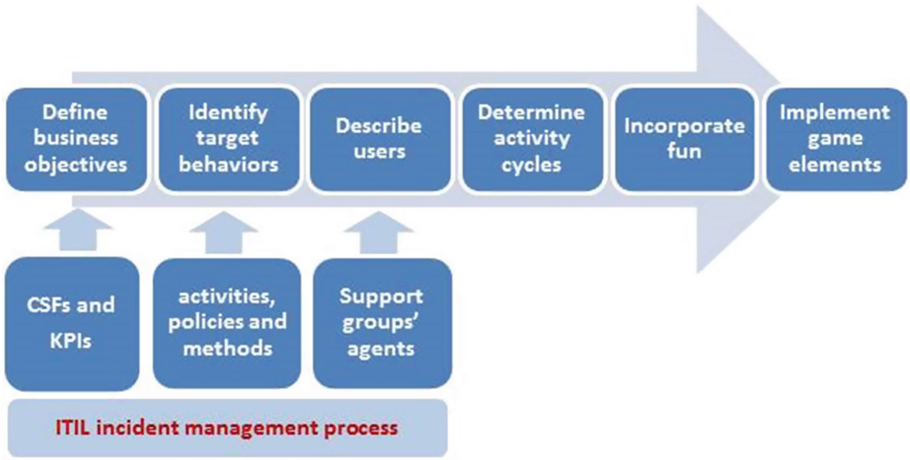

Fig. 1. Method for gamifying the incident management process

In the following paragraphs the activities of the gamification method proposed are described.

(1) *Define business objectives*

A project to gamify the incident management process has to begin identifying the problems to address, looking at how gamification can help solve them, and determining goals that could be used to measure the gamification project success. Defining concrete objectives will help organizations map them to the agents' behavior they need encourage.

ITIL proposes that organizations identify appropriate critical success factors (CSFs) for the incident management process that are based on their objectives. Besides, ITIL proposes that organizations define key performance indicators (KPIs) to support their CSFs. The compliance of the KPIs target values should be monitored and used to identify the incident management problems to address and to measure the gamification project success.

Table 1 shows several examples of CSFs and KPIs proposed by ITIL for the incident management process. The process KPIs can also be categorized by different criteria such as category, impact or urgency [3]. Based on ITIL recommendations, each organization will define its concrete KPIs according to its level of maturity, its CSFs and its particular circumstances [3].

Table 1. Critical success factors and key performance indicators for incident management

Critical success factor	Key performance indicator
CSF1. Resolve incidents as quickly as possible	KPI1. Average incident resolution time KPI2. Number and percentage of incidents resolved remotely KPI3. Number and percentage of incidents resolved without impact to the business
CSF2. Maintain quality of IT services	KPI4. Total number of incidents KPI5. Size of current incident backlog for each IT service KPI6. Number of entries in the system KPI7. Number and percentage of major incidents for each IT service
CSF3. Maintain user satisfaction with IT services	KPI8. Average user survey score (total and by question category) KPI9. Percentage of satisfaction surveys answered
CSF4. Align incident management activities with business objectives	KPI10. Number and percentage of incidents resolved within agreed response time (SLA) KPI11. Cost per incident
CSF5. Use adequate procedures for an efficient incident management	KPI12. Number and percentage of incidents incorrectly assigned KPI13. Number and percentage of incidents incorrectly categorized KPI14. Number and percentage of incidents incorrectly processed per agents

(2) *Identify target behaviors*

The purpose of this activity is to determine what agents' behaviors will help organization meet the established objectives. Besides, metrics to measure such behaviors will be defined. A systematic examination of the metrics values will allow organizations to determine the behaviors that it is necessary to improve.

In a gamification project of the incident management process, the organization will define target behaviors aligned with its objectives and problems taking into account the activities, methods and procedures of the process recommended by ITIL.

Examples of target behaviors that could be promoted for an efficient incident management are the following: (a) record a new incident, (b) determine and record an adequate incident category and priority, (c) perform an initial incident diagnosis and solve the incident (service desk's agents), (d) resolve the incident with a known error solution, (e) resolve the incident within the agreed response times (SLA), (f) escalate the incident to the next support level if the agent is unable to resolve the incident or the target resolution times are exceeded, (g) record the actions taken to solve the incident, (h) check whether the incident is fully resolved, and (i) carry out an user satisfaction survey, among others.

(3) *Describe users*

IT support organizations usually consist of a network of support groups structured in several levels (typically 3 to 5). The agents of the support groups at lower levels conduct generic tasks, while the agents of higher levels groups perform more technical and specific tasks. In our case, the support groups' agents are the users of the gamified system.

This activity focuses on understand the support groups' agents, and define and characterize target groups which include the identification of agents' needs, motivations and behavior in the process [20]. To collect and analyze information about the agents, several methods can be used, such as interviews [21], observations [22] or measurements of actual agents' behavior [5].

(4) *Determine activity cycles*

An adequate way to model the action in a gamified system is through engagement loops: agents actions result from motivation and in turn produce feedback in the form of responses from the system (points, badges, levels, etc.) [5].

In a gamification project of the incident management process, the engagement loops will be defined to drive the agents' actions through the process to ensure that they use suitable methods and procedures for an efficient incident management, and perform in an adequate manner the activities of the process adopted in the organization to meet the agreed SLAs. Besides, progression stars will be defined to indicate the agents' progress through the process.

(5) *Incorporate fun*

The aim of this activity is to make agents' work more entertainment. The concrete characteristics of the game to include will depend on diverse aspects such as agents' personality, the objectives to meet and the particularities of the incident management process adopted in the organization, among others.

(6) *Implement game elements*

This activity consists on mapping the desired agents' behavior to specific game elements that state what agents will do, and why and how they will be rewarded. Since the challenges of the agents of each organization are unique, defining appropriate game elements that will work for the organization is fundamental for the gamification project success.

According to their abstraction level, the game elements can be categorized as dynamics, mechanics and components (see Table 2) [5]. Using the adequate mechanics and components, it is possible to create an experience that drives the agent's behavior by satisfying the game dynamics.

Finally, the game mechanics and components selected will be implemented in the incident management support tool installed in the organization.

Table 2. Game elements

Game elements	Definition	Examples
Game dynamics	The most abstract game elements which are related to the agent needs and aspects that motivate them intrinsically	Enhance, emotions, sense of progression, relationships, etc.
Game mechanics	Basic actions that motivate and engage agents and drive their behavior	Challenges, competition, chance, feedback, rewards, win states, etc.
Game components	Concrete instantiation of game dynamics and mechanics	Achievements, points, badges, leaderboards, levels, etc.

4 Example of Use

This section presents an example of use of the method proposed to gamify the incident management process in a hypothetical organization. Given that the gamification strategy has not been implemented, we focus on describing the process problems to address, the business objectives to meet, the target behaviors to promote, and the activity cycles and the game elements considered in the gamification project.

(1) *Process problems and business objective*

The problems of the incident management process to address and the business objectives to meet considered in the example of use are shown in Table 3.

(2) *Target behaviors and metrics*

The target behaviors to promote in the agents to help meet the business objectives shown in Table 3, and the metrics defined to measure such behaviors are indicated in Table 4.

Table 3. Problems and business objectives

Process problem	Business objectives
P1. A small proportion of agents are fully trained and committed	O1. Improve the percentage of agents that complete the learning activities
P2. 20% of incidents are not recorded correctly	O2. Increase the percentage of incidents correctly recorded
P3. Only 1/3 of incidents are resolved at first support level	O3. Increase the percentage of incidents resolved at first support level
P4. 25% of incidents are solved without meeting the agreed times (SLA)	O4. Reduce the average incident resolution
P5. There are few entries into the known error database	O5. Increase the percentage of incident solutions recorded
P6. Customers are not satisfied with IT services	O6. Improve customer satisfaction with IT services

Table 4. Target behaviors and metrics

Target behavior	Metric
O1. Improve the percentage of agents that complete the learning activities	
• Perform the process learning activities	• Percentage of agents that complete the learning activities
• Conduct a process quiz	• Percentage of agents whose quiz scores are greater than or equal to a target value
O2. Increase the percentage of incidents correctly recorded	
• Record the necessary information of the incident	• Percentage of incidents correctly recorded
O3. Increase the percentage of incidents resolved at first support level	
• Perform an initial incident diagnoses and solve the incident	• Percentage of incidents solved by service desk's agents within the agreed times
O4. Reduce the average incident resolution time	
• Resolve the incident within the agreed times (SLA)	• Percentage of incidents solved within the agreed times
• Escalate the incident if the agent is unable to resolve the incident or the agreed times are exceeded	• Percentage of incidents escalated (for each support level)
O5. Increase the percentage of incident solutions recorded	
• Record the actions taken to resolve the incident • Re-use known incident solutions	• Percentage of incident solutions recorded into the known error database • Percentage of incidents solved with a known solution
O6. Improve customer satisfaction with IT services	
• Check the incident is fully solved • Close the incident • Carry out an user satisfaction survey	• Percentage of incidents fully solved • Percentage of incidents closed • Average user survey score

(3) *Activity cycles*

The actions of the support groups' agents that will produce system feedback have been grouped in different evolution levels which are aligned with the incident lifecycle (see Table 5). In the first level, the agent creates his system account, performs the learning activities and conducts a process quiz. The agent will pass to the second level when his quiz score is greater than or equal to the established target value. In the second level, the agent records the necessary information to treat the incident. In the third level, the agent performs the actions to solve the incident. Finally, in the last level the agent closes the incident and performs an user satisfaction survey.

Table 5. Agents' actions and evolution levels

Level	Action
1. Learn the process	A1. Create an account in the system A2. Perform a process learning activity A3. Conduct a process quiz
2. Record incidents (service desk's agents)	A4. Record a new incident A5. Determine and record the incident category A6. Determine and record the incident priority
3. Resolve incidents	A7. Perform an initial incident diagnosis and solve the incident within the agreed times (service desk's agent) A8. Resolve the incident within the agreed times with a known incident solution A9. Resolve the incident within the agreed times without a known incident solution A10. Escalate the incident to the next support level A11. Record the actions taken to solve the incident
4. Close incidents (service desk's agents)	A12. Check the incident is fully resolved A13. Close the incident record A14. Conduct a user satisfaction survey

(4) *Game elements*

The game dynamics considered in the example of use are the following:

- *Emotions:* engagement, motivation, competitiveness, feeling of progression and desire of status of the agents.
- *Progress:* improve agent skills and performance.

The game mechanics included to motivate and engage the agents, and to drive their behavior through the process are the following:

- *Challenges:* agents will learn how to treat the incidents and will improve their performance conducting the actions indicated in Table 5.
- *Chance:* agents will receive unexpected rewards depending on the improvement of their performance.
- *Feedback:* agents will receive feedback and rewards for performing the actions shown in Table 5 and meeting the agent objectives indicated in Table 6.

Table 6. Agent objectives

Business objective	Agent objective
O1. Improve the percentage of agents that complete the learning activities	AO1. Complete all the learning activities AO2. Process quiz score >= target value
O2. Increase the percentage of incidents correctly recorded	AO3. The percentage of incidents correctly recorded improves by a target value
O3. Increase the percentage of incidents resolved at first support level	AO4. The percentage of incidents resolved by the service desk's agent improves by a target value
O4. Reduce the average incident resolution time	AO5. The percentage of incidents solved by the agent within the agreed times improves by a target value AO6. The percentage of incidents solved by the agent within the agreed times is greater than or equal to a target value
O5. Increase the percentage of incident solutions recorded	AO7. The percentage of actions taken to resolve the incidents recorded improves by a target value AO8. The percentage of incidents solved using a known solution improves by a target value AO9. The percentage of incidents solved using a known solution is greater than or equal to a target value
O6. Improve customer satisfaction with IT services	AO10. The percentage of incidents closed by the agent improves by a target value AO11. The percentage of incidents closed by the agent is greater than or equal to a target value AO12. The percentage of customers satisfied with the incident solutions improves by a target value AO13. The percentage of customers satisfied with the incident solutions is greater than or equal to a target value

- *Win states:* agents will achieve win states when they greatly improve their performance or stand out from the other agents.

The game mechanics above indicated have been instantiated through the following game components:

- *Points:* agents will receive points when they conduct the actions included in Table 5.
- *Badges*: agents will receive badges when they meets the agent objectives indicated in Table 6 which are aligned with the business objectives shown in Table 3.
- *Trophies:* agents will receive trophies weekly when they obtain the highest value of the metrics defined to measure the target behaviors (see Table 4).
- *Progress bar*: it shows the agents' progress with respect the compliance of the established objectives.
- *Leaderboard*: it allows to compare the progress and performance of the agents.

The objectives compliance will be measured on intervals of on a week duration so that agents feel they have an opportunity to do better.

5 Conclusions

The work presented in this paper shows the usefulness of applying game elements in the context of the incident management process to increase the motivation and commitment of the support groups' agents, and to drive their behaviors through the process. Likewise it highlights the importance of disposing of a gamification method that helps organizations determine the most suitable game elements to include for improving the skills and performance of the agents. The example of use of the method presented in this work shows how to identify the following: (a) the process problems to address, (b) the objectives to meet, (c) the agents target behaviors to promote, and (d) the game elements that state what agents will do, and why and how they will be rewarded.

Finally, our further work will be focused on the definition and implementation of strategies to gamify the incident management process in real organizations applying the method proposed in this paper. The incident management process KPIs and the target behavior metrics defined in the organization will be monitored to evaluate the gamification project success. Besides, we will work on the definition of method for gamifying other ITSM processes and their applications in real cases.

Acknowledgments. This research paper has been funded by the Spanish State Research Agency and ERDF funds (grants TIN2013-46928-C3-2-R and TIN2016-76956-C3-3-R), and the Andalusian Plan for Research, Development and Innovation (grant TIC-195).

References

1. Office of Government Commerce (OGC), ITIL® Continual Service Improvement (2011)
2. Korsaa, M., Johansen, J., Schweigert, T., Vohwinkel, D., Messnarz, R., Nevalainen, R., Biro, M.: The people aspects in modern process improvement management approaches. J. Softw.: Evol. Process **25**(4), 381–391 (2012). doi:10.1002/smr.570
3. Office of Government Commerce (OGC), ITIL® Service Operation (2011)
4. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining gamification. In: Proceeding of the 15th International Academic MindTreck Conference: Envisioning Future Media Environments, pp. 9–15 (2011)
5. Werbach, K., Hunter, D.: For the win: how game thinking can revolutionize your business. Wharton Digital Press, Philadelphia (2012)
6. Bunchball: Using gamification to engage employees (Technical report) (2014)
7. Gartner: Gamification 2020: What Is the Future of Gamification? 5 November 2012
8. Cognizant Reports: Gamifying business to drive employee engagement and performance (2013)
9. Robson, K., Plangger, K., Kietzmann, J.H., McCarthy, I., Pitt, L.: Game on: engaging customers and employees through gamification. J. Bus. Horiz. **59**(1), 29–36 (2016)
10. Mesquida, A.L., Mas, A., Amengual, E., Calvo-Manzano, J.A.: IT service management process improvement based on ISO/IEC 15504: a systematic review. Inf. Softw. Technol. **54**, 239–247 (2012). doi:10.1016/j.infsof.2011.11.002

11. Iden, J., Eikebrokk, T.R.: Implementing IT service management: a systematic literature review. Int. J. Inf. Manage. **33**, 512–523 (2013). doi:10.1016/j.ijinfomgt.2013.01.004
12. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th International Academic MindTreck Conference: Envisioning Future Media Environments, pp. 9–15 (2011). doi:10.1145/2181037.2181040
13. Raflesia, S.P., Surendo, K.: A conceptual framework for implementing gamified-service to improve user engagement by using ITIL. In: Proceedings of the 9th International Conference on Telecommunication Systems Services and Applications (2015). doi:10.1109/TSSA.2015.7440440
14. Raflesia, S.P., Surendo, K.: Designing gamified-service towards engagement and service quality improvement. In: Proceedings of the 9th International Conference on Telecommunication Systems Services and Applications (2015). doi:10.1109/TSSA.2015.7440439
15. Conceicao, F., Silva, A., Filho, A., Cabral, R.: Toward a gamification model to improve IT service management quality on Service Desk. In: Proceedings of the 9th International Conference on the Quality of Information and Communications Technology (QUATIC) (2014). doi:10.1109/QUATIC.2014.41
16. Chunpir, H.I.: Prioritizing tasks using user-support-worker's activity model (USWAM). In: Yamamoto, S. (ed.) HIMI 2016. LNCS, vol. 9735, pp. 379–390. Springer, Cham (2016). doi:10.1007/978-3-319-40397-7_36
17. Yuan, Y., Qi, K.K., Marcus, A.: Gamification and persuasion of HP IT service management to improve performance and engagement. In: Nah, F.F.-H., Tan, C.-H. (eds.) HCIB 2015. LNCS, vol. 9191, pp. 550–562. Springer, Cham (2015). doi:10.1007/978-3-319-20895-4_51
18. Orta, E., Ruiz, M.: A simulation and gamification approach for IT service management improvement. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 84–97. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_7
19. Gartner (2012). http://www.gartner.com/newsroom/id/2251015
20. Morschheuser, B., Werder, K., Hamari, J., Abe, J.: How to gamify? a method for designing gamification. In: Proceedings of the 50th Annual Hawai International Conference on System Sciences (2017)
21. Deterding, S.: The lens of intrinsic skills atoms: a method for gameful design. Hum.-Comput. Interact. **30**(3–4), 294–335 (2015)
22. Herzig, P., Ameling, M., Wolf, B., Schill, A.: Implementing gamification: requirements and gamification platforms. In: Reiners, T., Wood, L.C. (eds.) Gamification in Education and Business, pp. 431–450. Springer, Cham (2015). doi:10.1007/978-3-319-10208-5_22

A Systematic Investigation into the Use of Game Elements in the Context of Software Business Landscapes: A Systematic Literature Review

Serhan Olgun[1,2], Murat Yilmaz[1,2(✉)], Paul M. Clarke[3,4], and Rory V. O'Connor[3,4]

[1] Department of Computer Engineering, Çankaya University, Ankara, Turkey
c1271202@student.cankaya.edu.tr, myilmaz@cankaya.edu.tr

[2] Virtual Reality Research Laboratory, Çankaya University, Ankara, Turkey

[3] School of Computing, Dublin City University, Dublin, Ireland
{paul.m.clarke, rory.oconnor}@dcu.ie

[4] Lero – The Irish Software Research Centre, Dublin, Ireland

Abstract. The software development process is a set of socio-technical activities to produce software artifacts in which humans play a crucial role. Since it is a people centric activity, factors such as user motivation, engagement, communication and collaboration might constrain these activities. Therefore, software business organizations stand to benefit from adopting different tools and methods in order to overcome these obstacles and to improve their software business processes. Research has been made to increase software quality and enhance the software development process. Alongside these studies, innovative techniques and concepts are beneficial. As a solution, the notion of gamification (i.e. employing game elements in non-gaming contexts) has been introduced to enhance the software development process and overcome the challenges mostly related to human factors. However, the applicability of game elements in the context of software business landscapes is still a controversial issue and not totally proven as of yet. Numerous studies have been conducted to examine the benefits of gamification and how game elements affect the software development process. Thus, in this paper, a systematic literature review was conducted in order to investigate the application of game elements both in research and industrial levels of software development and as well as in software business landscapes.

Keywords: Gamification · Software development · Game elements · Systematic review

1 Introduction

Software development organizations are professional business firms, which are founded so as to develop high quality software products that satisfy the customer expectations and their business objectives. In these organizations, valuable robust and reliable software products are produced as a result of the development teams following

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 384–398, 2017.
DOI: 10.1007/978-3-319-67383-7_28

an appropriate development process. Software product development may require an extensive development process, which includes analysis, development, testing and maintenance steps. Yilmaz states that [1, p. 1] "*A software development process is considered as the coordination of structural social activities (e.g. management, production and maintenance) coupled and constrained with a set of individuals' (i.e. participants who perform the activities) roles and skills for producing software artifacts in a predefined productivity level*". Theoretically, software development process is an organized structure that has consecutive steps to produce software applications. This process is performed by small, medium or large-scaled experienced software development teams by communicating with their customers in software business companies. In other words, software development is a teamwork that requires quite effective people involvement, engagement, collaboration and motivation to accomplish the process steps, so it concerns both software developers and customers.

Since it is a human centric activity, in the course of proceedings some obstacles may occur to achieve the goals. Human factors (e.g. user motivation, engagement, communication and collaboration) are major parameters for the success of the development process. This can be investigated by understanding the human role in the software development. Individuals can affect the complexity process by being a stakeholder (e.g. customer, developer or manager, etc.) and ultimately the quality of a software product. The following are just some of *the common reasons* for why these problems occurred during software production: (i) lack of communication among team members that causes collaboration problems, (ii) misunderstood of business requirements and objectives from customer, (iii) late performance evaluation, (iv) lack of reward systems by the management and (v) software practitioners with insufficient technical experiences [19].

With respect to the state of the art as described the idea is to use game elements to engage, motivate, train and monitor all the employees [20] to make them passionate to involve them in the whole development process in the context of software business landscapes. The gamification broadens a new horizon almost all areas in the non-game contexts. Its description has been made with the following statement: "*gamification is the use of game design elements in non-game contexts*." [4, p. 2]. In gamified contexts, a rewarding mechanism frequently exists to encourage people and excite people's attention to increase the engagement. If gamification applied to the software development it might bring several advantages. From employee perspective, because of its rewarding mechanism to increase engagement, awareness, motivation and collaboration issues. From managers' perspective, it may also have advantages for the performance management to monitor the development team members.

The application of game elements in the context of software business landscapes is still not an obvious issue and needs research and experiments on it. Therefore, this study investigates the applicability of the use of game elements in the context of software business landscapes and how these game elements affects the quality of software development in software business landscapes in order to overcome the challenges in the software industry.

The purpose of this research is to investigate the use of game elements in the context of software business landscapes. To summarize, this paper aims to address the following research questions: **(i)** What is currently known about people's motivation, engagement and performance issues of software development process? **(ii)** Which game elements and gamification approaches can be applied into software business landscapes to increase the quality and performance of software development? **(iii)** What are the examples of these approaches for the software industry? **(iv)** How these approaches affect the quality of software development in software business context?

The paper is structured as follows: Sect. 2 presents background information about gamification and software development process. Section 3 presents the research methodology about how the systematic literature review was planned. Section 4 presents the analysis and results that are obtained from the study. Section 5 discusses the future work and concludes the paper.

2 Background

In this section background information about the related topics will be presented. In first, we provide brief information about gamification.

2.1 Gamification

The term gamification is defined in [4, p. 2] as "*using game design elements in non-game contexts to motivate and increase user activity and retention*". As it is clearly stated in its definition gamification uses game design elements to gamify environments and ultimately aims to change people's behavior in positive manner and keep engaged and motivated them in particular tasks in non-gaming environments such as workplaces, schools or in software development organizations. By the increase in the application of real life examples of gamification in different domains, the popularity and usage of it has been growth in the last years. One of the domains where gamification is popular is education and mostly for training purposes [7]. The goal is to increase the motivation, engagement and productivity of students. Gamification has also been used to maximize the user engagement and keep user motivated by entrepreneurs, customer oriented web site owners [8]. StackOverflow [9] is a good an example where the game elements are intensely used. This web site is a knowledge exchange system for developers where users take badges or performance rates according to their activities in particular actions. Because of its effectiveness and efficiency, game design elements have also been applied in business environments to improve employee's performance while they accomplish their tasks and works [10].

2.2 Gamification and Software Development Process

This study focuses on the applicability of the gamification in software business landscapes and how these game elements affect the quality of software development process in software business landscapes in order to overcome the challenges in the software industry. Thus, this makes gamification a promising field to overcome the challenges related to human factors such as people involvement, engagement, collaboration and motivation throughout software development process. Due to the above stated reasons, some software development process tools have started to integrate to game elements to benefit from gamification principles. Visual Studio Achievements [11], JIRA Hero [12], PropsToYou [13], ScrumKnowsy [14], MasterBranch [15], RedCritter [16] are examples of commercial tools that are offering gamification in software development landscapes. Therefore, researchers and practitioners recognized that the game elements could be applied to the software business landscapes. However, the applicability of this issue is not obvious so it needs some research on it. Thus, a lot of proposals and academic researches about the topic have been published. In this paper, a research approach is proposed for understanding all following effects to improve the quality of a software development process by conducting a systematic literature review.

3 Research Methodology

We conduct a Systematic Literature Review (SLR) to identify a group of papers that discuss the application of game elements in software development landscapes. A systematic literature review is a research methodology to find out what we know and what we do not know based on the focused research questions. "*As a research area matures there is often a sharp increase in the number of reports and results made available, and it becomes important to summarize and provide overview*." [2, p. 1] Therefore, systematic literature reviews should certainly contain the question that it tries to answer and should report fully on the methods that have been utilized. According to Kitchenham, "*Systematic Literature Review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.*" [3, p. 1] As described, the word systematic stands for planned, methodical acting and review stands for critical appraisal of works.

$$\begin{array}{lcl} \text{Systematic} & \rightarrow & \text{Planned, methodical acting.} \\ \text{Review} & \rightarrow & \text{Critical Appraisal of Works.} \\ \text{Synthesis} & \rightarrow & \text{Get together findings.} \end{array}$$

This planned and methodical literature review is conducted by carrying out these steps. The Fig. 1 shows the Systematic Literature Review process and steps. These steps are briefly explained below:

- **Planning:** It is the first step of the review where the need for review needs to be identified, research questions are identified, and a review protocol is developed and evaluated.
- **Conducting:** It is the second step of the review which includes the following sub steps: Primary study selection, the data extraction, study quality assessment, and the data synthesis where the obtained data are synthesized.
- **Documenting:** It is the last step of the review to report the dissemination of information by drawing a conclusion and considering threats.

Hence, the systematic literature review is used to review the primary studies to find out the answers about the research questions and observe the results. As Petersen [2] states SLR uses existing studies related to research topic describes the context and summarize the results. Kitchenham [6, p. 3] explains the reasons why systematic literature review is conducted:

- "To summarize the existing evidence concerning a treatment or technology e.g. to summarize the empirical evidence of the benefits and limitations of a specific agile method.
- To identify any gaps in current research in order to suggest areas for further investigation.
- To provide a framework/background in order to appropriately position new research activities."

By this systematic literature review the existing proposals and research works for applying game elements into software development process in software business landscapes are determined, analyzed and classified to attempt to answer the questions and report them clearly for future research. While conducting this review, recommendations in [3, 17] are followed to make the review better and decide the best solution for the investigation. This section continues with describing search strategy and definition of research questions, identification of inclusion and exclusion criteria, searching relevant studies, extracting data and, at the end, synthesis of the study.

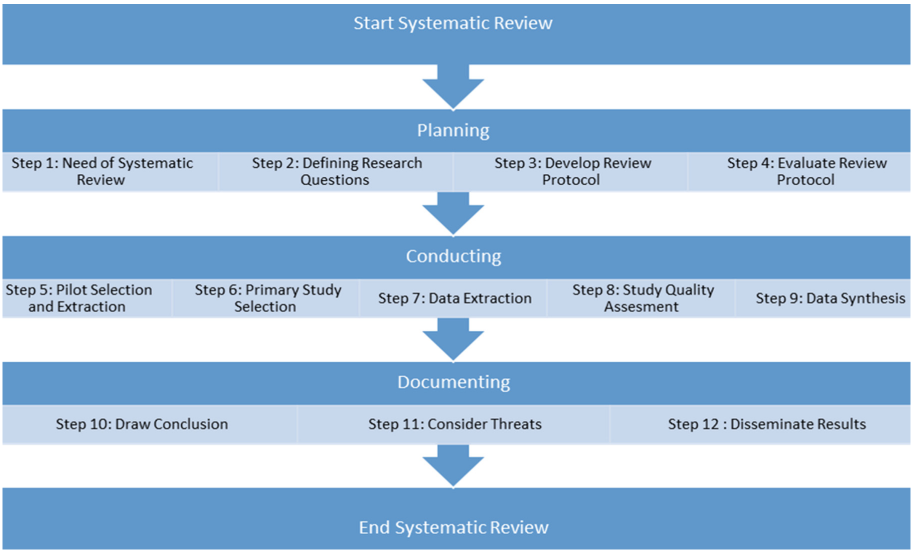

Fig. 1. Systematic literature review steps (Adapted from [6])

3.1 Research Questions, Data Sources and Search Strategy

Depending on the systematic literature review principles [3], in this step, some research questions are established for this study to find answers from existing research work and proposals related to application of game elements to software development process in the context of software business landscapes. Research questions are related to

Table 1. Aim of the research questions for the systematic review

ID	Question	Aim
RQ1	What is currently known about people's motivation, engagement and performance issues of software development process?	To identify which factors affects the people's motivation, engagement and performance during the software development process
RQ2	Which game elements and gamification approaches can be applied into software business landscapes to increase the quality and performance of software development?	To determine which game elements and gamification approaches can be applied to software business landscapes to enhance the success stories by increasing the quality and performance of software development
RQ3	What are the examples of these approaches for the software industry?	To show the applied approaches in the software industry
RQ4	How these approaches affect the quality of software development in software business context?	To determine which approaches remedy the software development process quality

applicability of the use of game elements in the context of software business landscapes and how these game elements affect the quality of software development in software business landscapes in order to overcome the challenges in the software industry. Different perspectives for applying game elements into software business landscapes are considered and associated with each research question separately. The research questions and their aims for this systematic review are described in Table 1.

These research questions are going to be guide for the data extraction phase and it helps to avoid reading full text of current research works related to the review. By the help of the research question that are stated above we focused on how game elements and mechanics remedy to increase the quality of software development process. During this review we reviewed the current research works and inspected how they focus on the gamification and software development process. Moreover, we observed other papers' proposals and approaches to find the efficient and effective solutions. For example, Research Question 1 (RQ1) focuses on investigation of identification for which factors affects the people's motivation and performance during the software development while Research Question 2 (RQ2) focuses on which game elements and gamification approaches or game elements such as badges, points award mechanisms can be applied to software business landscapes to enhance the success stories and increase the motivation of the people of the software development process. In Research Question 3 (RQ3) we identified the real life examples that adopts the gamification technique to their software development process. And finally, in Research Question 4 (RQ4) we observed how these approaches affect the quality of software development process in the industry. To get a clear picture of current research and to find the answers to research questions, popular and reliable data sources are used to carry out this review. The search strategy included the academic electronic databases and Google

Search is used for including extra publications and contributions for this study. The following academic electronic databases are used in this review:

- Çankaya University Electronic Database
- IEEE Explore
- Science Direct
- Access Engineering
- ACM Digital Library
- SpringerLink Journals

To find the answers to conduct this review using the electronic data sources some special combinations, keywords and search strings are designed. To design the search strings, the following keywords are used as major terms: Software Development Process and Game Elements. This search string is designed according to steps which are stated in [18] and all these terms related to this review topic is combined using Boolean AND or OR operator to retrieve the articles that are only related to this topic. The search string that is used in this review study is indicated in Table 2.

Table 2. Search strings for this systematic literature review

Major terms	Alternative terms
Software development process	((software development) OR (software engineering) OR (software process) OR (software development phases) OR (software development lifecycle) OR (software planning) OR (software testing) OR (software analysis) OR (software maintenance) OR (software design) OR (software quality) OR (software configuration management) OR (software validation) OR (software verification)) AND
Game Elements	((gamification) OR (gamifying) OR (gamify) OR (gamification mechanisms) OR (game mechanics))

These keywords and search strings are built up based on the research questions to get as many papers possible. Also, the alternative keywords and terms are added to major terms (shown in Table 2. Search strings for this Systematic Literature Review) to retrieve and cover more papers from the stated electronic databases. These search strings are applied to title, abstract and keywords to get the results. The Table 3 is presented to summarize the search strategy which is described above.

Table 3. Search strategy for this study.

Data source	Academic search	• Çankaya University Electronic Database • IEEE Explore • Science Direct • Access engineering • ACM Digital Library • SpringerLink Journals
	Non – academic search	• Standard google search engine
Items		• Academic papers • Academic publications • Journals, Book sections • Conference papers • Online academic publications
Search Applied to		• Title, Abstract, Keywords
Language		• Only written in English
Publication between		• 2010–2017

3.2 Study Selection Criteria

Study selection criteria describe the inclusion and exclusion standards whether to include or not the existing research works and proposals depending on the research questions, search string and strategy when conducting a systematic literature review. Therefore, a search result must meet the constraints that are defined in inclusion and exclusion standards. Thus, the inclusion standards include the search result whereas the exclusion standards eliminate the search result in the review.

The inclusion and exclusion criterion need to be defined to increase to more reliable results and decrease the risk to stray away from the topic. In other words, study selection criteria are designed and used for processioning for the review. Thus, this enables other researchers to rework by carrying out review using the same standards. When defining the study selection standards some criterion should be followed. The inclusion criteria should not be too general or too strict. When it is too general, the poor quality studies may be included and it affects the final results. On the other hand, when it is too strict, the entire related studies might not be presented in the final result. For this study, the inclusion standards are defined to get clear and reliable results. Studies are eligible for inclusion in the review if they presented empirical data on software development that are used game elements in their business context. In addition, the studies which are related to human factors (e.g. user motivation, engagement and collaboration, etc.) in software development are included to study selection criteria. Therefore, only academic papers and publications, professional forums and contributions such as conferences, online publications and book sections which related to this review domain are included to inclusion standards. Moreover, publications between 2010 and 2017 and only written in English is considered as the part of inclusion standards. On the other hand, according to search results from search strings, the search results including irrelevant data are excluded for this systematic review. Also, non–academic results, people opinions, personal blogs and personal web pages from the search results are excluded for this study. The review also excluded the academic results that meet the following criterion: duplicate papers that are retrieved from different databases, the papers available with only its abstracts, papers related to gamification of other domains (education, health, etc.).

The Table 4 presents the study selection criteria that are used in the review.

Table 4. Study selection criteria

Inclusion criteria	• Results including the relevant data • Academic papers • Academic publications • Journals • Book sections • Conference papers • Online academic publications • Papers written in English • Publications between 2010 and 2017
Exclusion criteria	• Main focus is not related gamification in software development process and human factors in software development process • Papers available only with its abstracts • Duplicate papers retrieved from different databases • Non – academic publications • Magazines, Personal web pages and blogs, Personal opinions
Exclusion criteria for full text	• Papers related to gamification of other domains (education, health, etc.) • Papers included the summary of conference notes

4 Results

We made our research using systematic literature review methodology by following the recommendations that are explained in [3, 17]. In first we developed search strings, identified data sources and decided study selection criteria, which are described in detail in Sect. 3. At the end, by following this search strategies we accessed and examined a lot of research work extracted the irrelevant data related to topic and collect the results to address the research questions. Table 5 shows the search results of each steps of the review process. As a result, number of 12 studies were obtained and these primary studies are listed in Appendix A section.

Table 5. The results obtained

Step	Process	Number of papers
Step 1	Search, obtain data	1093
Step 2	Data extraction, reading keywords, title or abstract parts	64
Step 3	Reading full text paper	33

4.1 Analysis Results of Research Questions

In this section we made an analysis to answer the research questions using the data that we obtained while conducting this review (Table 6).

Table 6. Analysis results

Research questions	Result
Research Question 1 (RQ1): *What is currently known about people's motivation, engagement and performance issues of software development process?*	User involvement plays a crucial role in software development process Therefore; the lack of user involvement affects the software development process in negative manner. Efficient and effective user involvement and user engagement requires high motivational aspects to increase the performance issues related to human factors. [A3] and [A12] propose that using game elements and mechanics increases the user involvement by increasing the user motivation, engagement and collaboration issues. As performance issues considered in software development, an explicit analysis is a key factor. Therefore, in their proposals in [A12] collaborative environment with team members and also stakeholders are important for better understanding to produce a well-understood software product. To build a collaborative environment this study proposes to use game elements such as ranking systems and badges. In [A3] researchers show the reward mechanism in software development to increase performance
Research Question 2 (RQ2): *Which game elements and gamification approaches can be applied into software business landscapes to increase the quality and performance of software development?*	We use this research question to determine which game elements and gamification approaches can be applied into the software development process in software business landscapes to enhance the success stories by increasing the quality and performance of software development. We classify and examine the primary studies that are about using game elements and mechanics. High-level motivation of team members causes to increase the quality of software development. In [A7], using game elements (rankings, badges, etc.) strengthens motivation of people; thus, it fosters to increase the quality and performance of software development. Human related issues affect the

(*continued*)

Table 6. (*continued*)

Research questions	Result
	quality and performance of development process. In [A10], researchers propose that using game elements as a solution to fix the human related obstacles maximizes the quality of development process. Moreover, according to the analysis that we made, we identify the following game elements and mechanics [A1], [A2] [A4], [A8], [A11] that may be applied into the software business landscapes to increase the quality and performance of software development process • Rankings • Badges • Levels • Quests • Awards, Point Based Reward Systems
Research Question 3 (RQ3): *What are the examples of these approaches for the software industry?*	We choose this research question to show and exemplify the applied approaches in the software industry. In this research question, we focused on the real-life examples in which gamification elements are used. In [A3] they present an online gamified tool which mainly uses reward mechanism as a game element to conduct the requirement elicitation process. Results show that it has positive impacts on people such as increasing motivation, engagement and collaboration. In [A4], the usage of gamification elements increases motivation and performance of the team members in Visual Studio platform. Also in [A4], they show the positive impacts of gamification on software development process by presenting real-life gamified applications which are *iThink (*for requirement), *CodeHunt* and *Visual Studio Achievements* (for software development) and *Microsoft Language Quality Game (*for testing). All these approaches have positive impacts on development process. Therefore, we have observed that some glorious version tools and source controls as JIRA [11], RedCitter [12], Visual Studio Achievements [16] in the industry. These examples use some basic game elements such as badges, levels and award mechanisms so as to monitor and

(*continued*)

Table 6. (*continued*)

Research questions	Result
	evaluate the performance of each team member. Hereby the most productive team member could be determined and may also rewarded
Research Question 4 (RQ4): *How these approaches affect the quality of software development in software business context?*	We choose this research question to determine which approaches remedy the software development process quality. Software development process requires people involvement, user engagement, collaboration and motivation. In [A3] and [A12] it is stated that gamification in software development overcome the user involvement problem. In [A12], gamification provides a collaborative environment by voting and ranking mechanisms. In [A9], they present a case study in which gamification is applied. From the results, they claim that applying game elements increases the performance of software development actors so it enriches the quality of development process. In [A1], [A4] and [A11] using game elements such as rewarding mechanism increases the performance and quality of the development process. Dorling and McCaffery [A6], argue that the adoption of gamification in software development causes the process quality improvement. In their example [A5] using gamification dynamics in a bug tracking system overcomes the human factors and increases quality of software development process

5 Conclusion

In this paper, we conducted a systematic literature review to classify and characterize the state of the art as regarding applying game elements into software development process. Firstly, we made a deep literature research by using the search strings that we developed at the beginning of the literature review to address each research questions to reach the goal. After we accessed data sources to show the applicability of gamification into software development process we performed a data inclusion and exclusion process to reach the goal. After this process, we classified primary studies according to their properties, which were their process areas, kind of game elements and mechanics that are applied in their studies and further discussed their effects. Moreover, we analyzed some empirical evidences (i.e. real life examples) to see the impacts of game elements on user motivation, engagement and collaboration to encourage the business organization to replace the current development process with the proposed methodology.

The results that we obtained from our preliminary investigation on applying game elements into software business landscapes provides a basic guidance for the software development organizations which aims to benefit from gamification. At the beginning of this literature review, we envisioned that using gamification in software development process should reduce the problems related to human factors (e.g. user motivation, engagement and collaboration, etc.). Thus, the results that we obtained from our research confirmed that using game element in this area supports our expectations. In addition, our results suggest that using gamification in software development increases the user motivation, engagement and collaboration. It also proves that it improves the software development process in terms of quality and performance, which creates a potential to resolve some obstacles related to human factors [5, 21, 22]. However, most of the studies that we analyzed show that there are some gaps in the areas such as project management and team configuration. In studies based on real life examples [23–25], the adoption of gamification approach into software development process provides some concrete but preliminary evidence. Therefore, it shows that it can be partially adopted into software development from a software management perspective.

Moreover, from the analyzed results, we have observed that applying game elements into other business domains such as marketing, education is going faster than software development. However, our results suggest that if the number of proposals, empirical studies and evidence increases for the adoption of gamification into software development to make the whole process better, it may complement the existing ecosystem of software development process.

Appendix A

[A1] Yilmaz, M., Yilmaz, M., O'Connor, R., Clarke, P. A gamification approach to improve the software development process by exploring the personality of software practitioners. In: Clarke, Paul and O'Connor, Rory and Rout, Terry and Dorling, Alec, (eds.) Software Process Improvement and Capability Determination. Communications in Computer and Information Science, 609. Springer, pp. 71–83. ISBN 978-3-319-38980-6

[A2] Kosa, M., Yilmaz, M. Designing Games for Improving the Software Development Process. Systems, Software and Services Process Improvement, 22nd European Conference, EuroSPI 2015, vol. 543, pp. 303–310, 2010.

[A3] Fernandes J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J. Madeiras, and Mira da Silva, M. "iThink: A Game-Based Approach Toward Improving Collaboration and Participation in Requirement Elicitation," Procedia Computer Science, vol. 15, pp. 66–77, 2012.

[A4] Lombriser, Philipp and Valk, Ronald van der. Improving the Quality of the Software Development Lifecycle with Gamification. Department of Information and Computing Sciences, Utrecht University.

[A5] S. Tomasso Dal, M. Andrea, L. Michele, M. Ebrisa. How to Gamify Software Engineering. 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 261–271.

[A6] Dorling, A., McCaffery, F., "The gamification of SPICE," Communications in Computer and Information Science, vol. 290, pp. 295–301, 2012.

[A7] Herranz, E., Palacios, R. C., de Amescua Seco, A., & Yilmaz, M. (2014). Gamification as a Disruptive Factor in Software Process Improvement Initiatives. J. UCS, 20(6), 885–906.

[A8] Yilmaz, M., O'Connor, R., Collins, J. Improving software development process through economic mechanism design. In: 17th European Software Process Improvement Conference, 1–3 Sept 2010, Grenoble, France. ISBN 978-3-642-15666-3

[A9] Passos, Erick B, Danilo B Medeiros, Pedro AS Neto and Esteban WG Clua. Turning Real-World Software Development into a Game. Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on, IEEE. 2011.

[A10] Yilmaz, M and O'Connor, R. (2010) Maximizing the value of the software development process by game theoretic analysis. In: 11th International Conference on Product Focused Software, 21–23 Jun 2010, Limerick, Ireland. ISBN 978-1-4503-0281-4

[A11] A. A. de Melo, M. Hinz, G. Scheibel, C. D. M. Berkenbrock, I. Gasparini, and F. Baldo, "Version Control System Gamification: A Proposal to Encourage the Engagement of Developers to Collaborate in Software Projects," presented at the Proceedings of the 6th Int. Conf. on Social Computing and Social Media (SCSM'2014).

[A12] D. Duarte, C. Farinha, M. M. da Silva, and A. R. da Silva, "Collaborative Requirements Elicitation with Visualization Techniques," presented at the Proceedings of the IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'12), 2012.

References

1. Yilmaz, M.: A Software Process Engineering Approach to Understanding Software Productivity and Team Personality Characteristics: An Empirical Investigation, Ph.D. thesis, Dublin City University, January 2013
2. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE 2012), pp. 68–77 (2008)
3. Kitchenham, B.: Procedures for Performing Systematic Reviews. Joint Technical report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia (2004)
4. Deterding, S., Khaled, R.: Gamification: toward a definition. In: CHI 2011 Gamification Workshop. ACM (2011)

5. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th International Academic MindTrek Conference, Envisioning Future Media Environments (MindTrek 2011), pp. 9–15 (2011)
6. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Software Engineering Group, Keele University and Department of Computer Science, University of Durham, United Kingdom, Technical report EBSE-2007-01 (2007)
7. Kapp, K.M.: The Gamification of Learning and Instruction, Pfeiffer (2012)
8. Currier, J.: Gamification: Game Mechanics is the New Marketing. OogaLabs Blog (2008)
9. StackOverflow. http://stackoverflow.com
10. Hugos, M.: Enterprise Games: Using Game Mechanics to Build a Better Business. O'Reilly, Sebastopol (2012)
11. Visual Studio Achievements. https://channel9.msdn.com/achievements/visualstudio. Retrieved 19 July 2017
12. Atlassian, JIRA Hero. https://marketplace.atlassian.com/plugins/com.madgnome.jira.plugins.jirachievements. Retrieved 19 July 2017
13. PropsToYou. https://www.propstoyou.com.au/. Retrieved 19 July 2017
14. ScrumKnowsy. http://www.scrumknowsy.com/. Retrieved 19 July 2017
15. Masterbranch. http://masterbranch.com/. Retrieved 19 July 2017
16. RedCritter. http://www.redcritter.com/. Retrieved 19 July 2017
17. Kitchenham, B.A., Budgen, D., Pearl Brereton, O.: Using mapping studies as the basis for further research – a participant–observer case study. Inf. Softw. Technol. **53**, 638–651 (2011)
18. Brereton, P., Kitchenham, B., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. **80**, 571–583 (2007)
19. Tsumaki, T., Tamai, T.: Framework for matching requirements elicitation techniques to project characteristics. Softw. Process Improv. Pract. **11**(5), 505–519 (2006)
20. Herger, M.: Enterprise Gamification: Engaging People by Letting Them Have Fun, vol. 1, 1st edn. CreateSpace Independent Publishing Platform (2014)
21. Paharia, R.: Loyalty 3.0: How to Revolutionize Customer and Employee Engagement with Big Data and Gamification. McGraw-Hill, New York (2013)
22. Werbach, K., Hunter, D.: For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press, Philadelphia (2012)
23. Aydan, U., Yilmaz, M., Clarke, P., O'Connor, R.V.: Teaching ISO/IEC 12207 software lifecycle processes: a serious game approach. Comput. Stand. Interfaces J. **54**(3), 129–138 (2017)
24. Kosa, M., Yilmaz, M., O'Connor, R.V., Clarke, P.: Software engineering education and games: a systematic literature review. J. Univ. Comput. Sci. **22**(11), 1558–1574 (2016). (available online)
25. Yilmaz, M., Yilmaz, M., O'Connor, R.V., Clarke, P.: A gamification approach to improve the software development process by exploring the personality of software practitioners. In: Clarke, P.M., O'Connor, R.V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 71–83. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_6

Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game

Alejandro Calderón[1(✉)], Mercedes Ruiz[1], and Rory V. O'Connor[2]

[1] University of Cádiz, Cádiz, Spain
{alejandro.calderon,mercedes.ruiz}@uca.es
[2] Dublin City University, Dublin, Ireland
rory.oconnor@dcu.ie

Abstract. Bringing professional practice into the learning/teaching process is an especially difficult task in the scope of software project management and can turn into a challenge in the context of software process standards education. The ISO 21500 standard is an international reference standard that provides generic guidance and good practices in project management. In this paper, we perform a literature review in order to analyze the current studies related to the use of serious games for understanding, teaching and supporting the education of the ISO 21500 standard. Moreover, we propose ProDec, a serious game for software project management training, and provide a mapping between the different stages of the game lifecycle and the ISO 21500 standard applying its management processes in the context of software projects. As a result, we observe that in this context, ProDec is able to cover seven of the ten subject groups and almost 75% of the project management processes of the ISO 21500 standard.

Keywords: ISO 21500 · Software project management · Serious games · Teaching standards · Simulation · Education

1 Introduction

Software engineering is a complex activity that requires a good integration of theoretical and practical information in order to create quality software [1]. To facilitate this activity, the software industry defines standards that provide guidance and processes with the goal to structure the activities and tasks for supporting the development of software. Within this discipline, software project management consists of applying knowledge, skills, tools, and techniques to software project activities in order to meet the project requirements [2].

Software project management is an important field for succeeding in the development of quality software [3]. Although there are many international standards that provide best practices, guidance and support software project management such as the ISO/IEC 12207 [4] or the ISO/IEC 29110 [5] and project management in general such

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 399–412, 2017.
DOI: 10.1007/978-3-319-67383-7_29

as the ISO 21500 [6], we can observe a crucial need for better understanding and training in them [7]. Thus, practitioners need to be involved in a practical and realistic learning/teaching process that allows them acquiring more practical experience in software process to be enough experts to produce more quality software [8].

This necessity moves trainers towards the development and use of methods and techniques to teach in a highly practical way, promote active learning and increase the motivation and the engagement of learners in software project management [9]. In this context, games can be considered as a learning resource to train novice software practitioners and to allow them to learn from their own mistakes and acquire experience in a free-risk environment [10].

Regarding the ISO 21500 standard as a basic guidance for project management that can be applied in the context of software project management, we can observe that there is a lack of works for understanding, teaching or supporting the education of the project management processes of the ISO 21500 standard. For that reason the main contributions of this paper are: (i) providing a complete view of the current studies related to the use of serious games for understanding, teaching or supporting the education of the ISO 21500 standard, (ii) analyzing the coverage of the project management processes of the ISO 21500 standard by the gameplay's lifecycle process of a proposed serious game in the context of software projects and (iii) evaluating the idea of using the proposed serious game as a learning resource for supporting the education of the software project management processes.

The remainder of this article is structured as follows: Sect. 2 shows the related works of this study. Section 3 describes a simulation-based serious game and evaluates the coverage of the project management processes of the ISO 21500 in the context of software projects. Finally, Sect. 4 summarizes the paper and presents our conclusions and future work.

2 Background

Several organizations have published best practices and standards to provide guidance and describe processes for supporting project management. For instance, the PRINCE2 method defined by AXELOS [11], the PMBOK guide proposed by the Project Management Institute [2] or the ISO 21500 standard [6]. In this work, we have decided to take the guidance and processes provide by ISO 21500 as a reference model because comparing with the other guides, it provides more general and basic guidance for project management. In addition, their processes do not need to be applied uniformly on all projects, the standard can be complied be adopting other recognized project management methods and it is well accepted on the international level by the industry and the scientific population.

2.1 ISO 21500

ISO 21500 international standard [6] aims to provide guidance for project management and a high-level description of concepts and processes that are considered to form good practice in project management. The standard is intended to be used by any type of organization as: (a) a reference in an audit; (b) a link between different project management and business processes; (c) a checklist to prove the knowledge and skills of project managers and project workers in executing projects; (d) a common reference between different methods, practices and models; (e) and a common language in project management.

The ISO 21500 standard is structured in four clauses that define the scope, the terms and definitions and the project management concepts and processes, and an informative annex. The fourth clause of the standard identifies the recommended project management processes in a generic way, with the goal they can be used by any project in any organization or entity. In this section, the ISO 21500 identifies five process groups regarding the management perspective of a project: Initiating, Planning, Implementing, Controlling and Closing On the other hand, regarding the project management practices, the ISO 21500 identifies 39 processes divided into ten project management topics named subject groups. Table 1 shows these processes related to the process and subject groups where they take place.

2.2 ISO 21500 and Serious Games

As a generic standard that recommends best practices in project management, ISO 21500 should be considered in any project management curricula in order to provide a best education, more attached to the practice and more realistic, where learners can put into practice their knowledge acquisition within real-life scenarios [3]. However, regarding the software engineering scope, teaching international standards can turn into a challenge for both industry and university trainers [7].

In this context, the use of games and simulation-based experiences helps trainers to achieve these goals by supporting the knowledge acquisition within a risk-free environment. These games, are called serious games, are designed with a different purpose than entertainment and allow participants to experiment, learn from their own mistakes and acquire experience [12].

With the goal of providing a complete view of the current studies related to the use of serious games for understanding, teaching or supporting the education of the standard ISO 21500, we have performed a systematic review of the literature for identifying the relevant related studies. For that, taking into account Kitchenham and Charters guidelines [13], we have performed the following steps:

(a) **Search strategy**

First, we identified a set of key terms. Taking into account these key terms and their synonyms, we performed some initial pilot searches to test and tune the search string. Table 2 shows the different search strings defined.

Using these search strings, we performed the search in the following digital databases: Wiley Online Library, ACM Digital Library, IEEE Xplore, ISI Web of Science, SCOPUS and SpringerLink. We adapted the different search strings to each digital database and restricted the search to title, abstract and keywords.

Figure 1 shows the results of applying the different search strings to each database. As we can observe, there are only reflected the data to four of the six selected databases. The reason is that SpringerLink and IEEE Xplore databases did not report any

Table 1. ISO 21500 processes.

Subject groups	Process groups				
	Initiating	Planning	Implementing	Controlling	Closing
Integration	P1. Develop project charter	P2. Develop project plans	P3. Direct project work	P4. Control project work P5. Control changes	P6. Close project phase or project P7. Collect lessons learned
Stakeholder	P8. Identify stakeholders		P9. Manage stakeholders		
Scope		P10. Define scope P11. Create work breakdown structure P12. Define activities		P13. Control scope	
Resource	P14. Establish project team	P15. Estimate resources P16. Define project organization	P17. Develop project team	P18. Control resources P19. Manage project team	
Time		P20. Sequence activities P21. Estimate activities durations P22. Develop schedule		P23. Control schedule	
Cost		P24. Estimate costs P25. Develop budget		P26. Control costs	
Risk		P27. Identify risks P28. Assess risks	P29. Treat risks	P30. Control risks	
Quality		P31. Plan quality	P32. Perform quality assurance	P33. Perform quality control	
Procurement		P34. Plan procurements	P35. Select suppliers	P36. Administer procurements	
Communication		P37. Plan communications	P38. Distribute information	P39. Manage communications	

Table 2. Search strings.

Search strings (SS)
SS1. "ISO 21500" AND "serious game"
SS2. "ISO 21500" AND game
SS3. "ISO 21500" AND (teach OR train OR educate)
SS4. "ISO 21500"

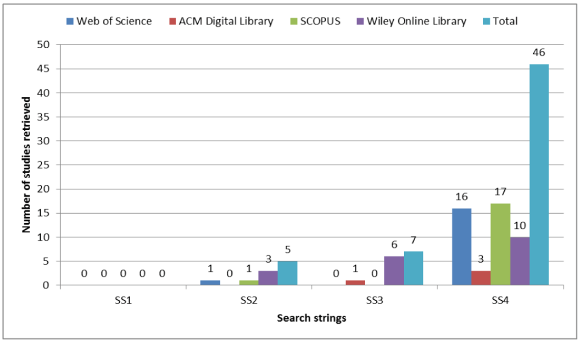

Fig. 1. Number of studies retrieved.

work with any of our search strings. Moreover, we can observe how the search string SS1 did not allow finding any study and the rest of the search strings allowed finding few studies. The search string that allowed finding more related studies was SS4 (46 studies), for that reason we decided to use it as the main search string of our review.

(b) **Study selection**

After we retrieved the studies from the selected search string and deleted the duplicated studies, we performed a selection process based on two phases, through a test-retest approach.

In Phase 1, studies found during the search process were evaluated for their suitability based on the analysis of their title and abstract. In this phase, studies that were clearly irrelevant were excluded. Studies related to the use of games for understanding, teaching or supporting the education of the standard ISO 21500 were classified as possible select (PS) and the rest as non-selected studies (NS).

In Phase 2, studies identified as possible select during Phase 1 were exposed to a more thorough analysis that included reading the full text. This phase was done to ensure that the study in question definitely contained information that is relevant to the study.

Figure 2 represents firstly, the papers that were retrieved from each consulted database; secondly, the number of different studies that were collected from each database after removing duplicates; thirdly, the number of studies that were collected from each database that passed the first review of the selection process. Finally, it shows the number of papers that were included as primary studies in our review.

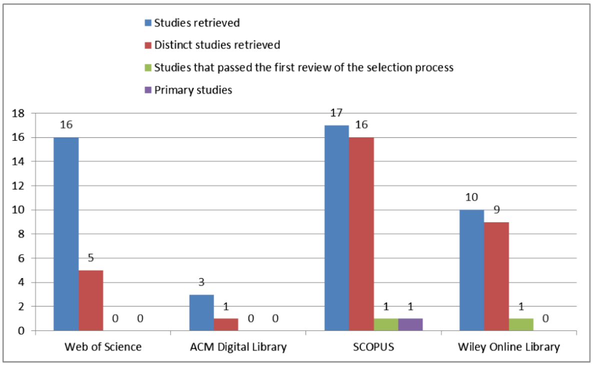

Fig. 2. Evolution of studies retrieved in each digital database.

(c) **Results**

The selection process began with 46 studies and ended with only the work of Mesquida et al. [14] as suitable for our review. In this work, authors proposed the use of games as a technique to facilitate the implementation of the project management processes proposed by the ISO 21500 standard. Concretely, authors presented a specific game to facilitate the implementation of two project management processes of the ISO 21500 international standard [14]. However, they do not introduce a serious game that allows teaching in the whole processes of the standard. Then, we can observe a lack of serious games as learning/teaching resources for understanding, training and supporting the education of the project management processes of the ISO 21500 standard.

For that reason, in this work, we take advantage of the features of a simulation-based serious game, called ProDec, to assess how it can be used for covering the project management processes of the ISO 21500 and supporting learners and practitioners in learning, understanding and practicing the project management processes of ISO 21500 standard.

3 Coverage of ISO/IEC 21500

In this work, we apply the guidelines of the ISO 21500 in the context of software project management with the goal to observe how ProDec can support the understanding and teaching of project management processes in software education. For that, in this section, we describe the main functionalities of ProDec related to the process and subject groups of the ISO 21500 and discuss how ProDec covers the different processes identified by the ISO 21500.

3.1 Game Description

ProDec [15] is a simulation-based serious game to train and motivate in learning, understanding and practicing the principles of software project management. As a learning resource, its main goal is that players put into practice their knowledge related to the concepts and practices of software project management in a risk-free virtual environment where they take the role of a project manager.

Regarding its main functionalities, we can highlight that ProDec: (a) provides a training environment that allows learners to take contact with all the software lifecycle stages of a software project from its conception to its closure; (b) provides trainers with an environment for supporting players assessment through the gameplays of the game; (c) provides both, learners and trainers, with an environment for game scenarios designing that allows them to create every project plan scenario they can think of; and (d) provides a learning/teaching environment that promotes learning by doing, active and social learning.

On the other hand, as a game, its objective is that players be able to manage a software project in a successfully way, this involves to complete the project within the time and cost limits. In the contrary, the game is over when the project significantly overruns either the approved budget or the allocated time.

For that, as Fig. 3 shows, starting from a statement of activity that defines the scope of the project scenario, players need to immerse in the gameplay's lifecycle process in order to win the game and get a final assessment report that allows them to learn from their own performance with ProDec. The gameplay's lifecycle process is composed of three main stages: Onset, Execution and End stage.

In the Onset stage, taking into account the statement of activity, players follow a process that guides them to create from scratch the game scenario that involves defining the project plan for the gameplay (see Fig. 3). This process is made of five sequential sub-stages which are the following:

- *Project Information (PI).* In this sub-stage, players provide the general information of the project about its scope and features, such as the salary of the workers, the length of the project, the number of use cases, etc., that are necessary to begin the size estimation stage.
- *Size Estimation (SE).* In this sub-stage, players make the size estimation of the project using Albrecht's method [16] of function points-base estimation.

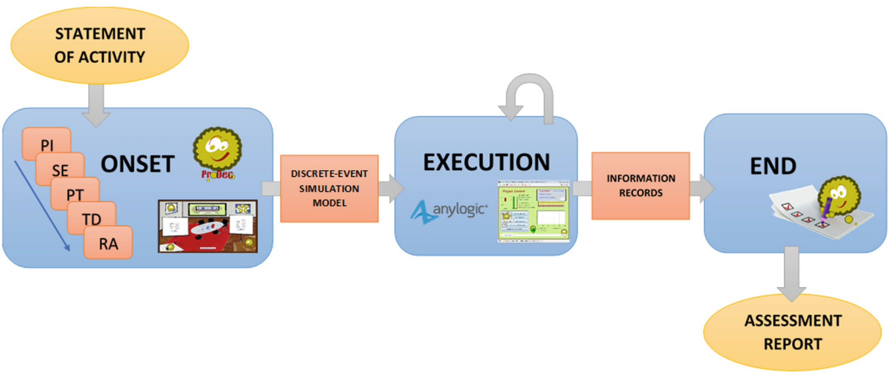

Fig. 3. ProDec gameplay's lifecycle process.

- *Project Team Definition (PT).* In this sub-stage, players define their project team. For this, they have to select their past work experience and some features for their personality based on the sixteen personality factors described by Cattell [17].
- *Tasks Definition (TD).* In this sub-stage, players define the tasks of the project based on PERT diagram [18], and enter, for each of them, the time data, the budget allocated, and its predecessor tasks. Moreover, players have to allocate the personnel for each task.
- *Risk Analysis (RA).* In the last stage of the process, players make a quantitative risk analysis.

Once the project plan of the game scenario is defined, the game automatically generates a source code file with the equations of a discrete-event simulation model that simulates the execution of a project plan and allows players to start the Execution stage of the gameplay's lifecycle process. During the simulation of the project plan execution, the game allows players to practice their decision-making skills by controlling and monitoring the progress of the project execution in order to correct the potential deviations of the progress of the project.

Finally, when the simulation of the project plan ends, players immerse in the End stage of the gameplay's lifecycle process. In this stage, they perform the closure of the project and get an assessment report related to their performance during the gameplay.

We can observe how the gameplay's lifecycle process of ProDec can be easily mapped to the process groups defined by the ISO 21500. In the following subsections, we describe the different activities associated with each stage of the gameplay's lifecycle and discuss their mapping with the project management processes of the ISO 21500 standard from a software point of view.

3.2 Onset Stage

The process of a ProDec's gameplay begins with a Statement of Activity that establishes the objectives of the gameplay and provides players with the needed information about the scope and requirements of the project scenario involved in the gameplay. Table 3 shows the activities players need to perform for starting a gameplay and the project management processes of the ISO 21500 that the game is able to cover with these activities. Before starting a gameplay, players have to review the Statement of Activity (*A1*) and analyze all the information provided in order to develop the project charter and the project plans (*P1* and *P2*). Once players know the features of the project scenario, they can start the gameplay (*A2*).

During the Onset stage of the gameplay's lifecycle process, players are involved in a process that guides them to create from scratch the project scenario. This process is composed of six main activities that allow players defining the general information of the project (*A3*), performing the size estimation of the project (*A4*), establishing the composition of the project team (*A5*), creating the schedule of the project tasks (*A6*), performing the risks analysis of the project (*A7*) and creating the project plan of the game scenario (*A8*).

In Table 4, we can observe these activities with all the sub-activities that players need to perform in order to create the project plan and define all the information related

Table 3. Coverage of ISO 21500 processes by the activities of the ProDec's Onset stage (I).

Activities of the onset stage		ISO 21500 processes
A1. Review the statement of activity	A1.1. Identify the objectives of the gameplay A1.2. Identify the scope and requirements of the project scenario	P1. Develop project charter P2. Develop project plans
A2. Start the gameplay	A2.1. Select the type of gameplay A2.2. Identify the players involve in the gameplay	

to the size, the project team, the tasks and the risks of the project scenario. Moreover, Table 4 maps the different activities of the Onset stage with the processes of the ISO 21500 standard that the game is able to cover through these activities. As we can observe the activity "*Define the Project Information (A3)"* allows covering the *"Develop project charter (P1)", "Develop project plans (P2)", "Identify stakeholders (P8)"* and *"Define scope (P10)"* processes of the ISO 21500 that belong to the Initiating and Planning process groups of the standard.

The activity "*Estimate the Size of the Project (A4)*" in which players have to perform the size estimation of the project, allows covering the "*Estimate resources (P15)"* and *"Estimate costs (P24)*" processes of the Planning process group of the ISO 21500 standard.

The process "*Establish project team (P14)"* of the Initiating process group and the process "*Define project organization (P16)"* of the Planning process group of the ISO 21500 standard are covered by the activity "*Define the Project Team (A5)"* in which players have to select and define the human resources for composing the work team of the project.

Through the activity "*Define the Project Tasks (A6)",* players define the schedule of the project tasks regarding the estimated start and completion dates, the assigned human resources and the dependency of the tasks. Therefore, through this activity players take contact with the following processes of the ISO 21500 Planning process group: *"Create work breakdown structure (P11)", "Define activities (P12)", "Sequence activities (P20)", "Estimate activities durations (P21)", "Develop schedule (P22)"* and *"Develop budget (P25)"*. Moreover, this activity also allows covering the "*Develop project team (P17)"* process of the ISO 21500 Implementing process group.

The activity "*Analyze the Project Risks (A7)"* allows covering the *"Identify risks (P27)* and *"Assess risks (P28)*" of the ISO 21500 Planning process group. Finally, the activity "*Create the Project Plan (A8)*" allows players to accept all the defined information and establish the project plan to be used in the gameplay scenario. Then, this activity supports the "*Develop project plans (P2)"* process of the ISO 21500 Planning process group.

Therefore, the activities of the ProDec's Onset stage allow players to take contact with the Initiating, Planning and Implementing process groups of the ISO 21500 standard.

Table 4. Coverage of ISO 21500 processes by the activities of the ProDec's Onset stage (II).

Activities of the onset stage		ISO 21500 processes
A3. Define the project information	A3.1. Define the context of the project A3.2. Identify the companies that are involved in the project A3.3. Establish the requirements of the project A3.4. Set the features of the development company related to the salary, workday and effort values A3.5. Establish the initial duration of the project	P1. Develop project charter P2. Develop project plans P8. Identify stakeholders P10. Define scope
A4. Estimate the size of the project	A4.1. Calculate the function points of each use cases A4.2. Calculate the total function points of the project A4.3. Estimate the workforce needed to perform the project A4.4. Estimate the initial budget	P15. Estimate resources P24. Estimate costs
A5. Define the project team	A5.1. Select the human resources of the project A5.2. Define the personality traits of each human resource that is involved in the project A5.3. Define the experience of each human resource that is involved in the project A5.4. Establish the composition of the work team	P14. Establish project team P16. Define project organization
A6. Define the project tasks	A6.1. Define the project tasks A6.2. Estimate the duration of each project task A6.3. Estimate the cost of each project task A6.4. Allocate the human resources for each task A6.5. Define the dependencies of the tasks A6.6. Create the schedule of the project tasks	P11. Create work breakdown structure P12. Define activities. P17. Develop project team P20. Sequence activities P21. Estimate activities durations P22. Develop schedule P25. Develop budget
A7. Analyze the project risks	A7.1. Identify the project risks A7.2. Make a quantitative analysis of the project risks	P27. Identify risks P28. Assess risks
A8. Create the project plan		P2. Develop project plans

3.3 Execution Stage

The second stage of ProDec gameplay's lifecycle process is the Execution stage. In this stage of the gameplay, players should perform three main activities with the goal to simulate the execution of the project scenario and perform the control and monitoring of the project (see Table 5).

Table 5. Coverage of ISO 21500 processes by the activities of the ProDec's execution stage.

<table>
<tr><th colspan="2">Activities of the execution stage</th><th>ISO 21500 processes</th></tr>
<tr><td>A9. Execute the simulation of the project plan</td><td>A9.1. Generate the simulation model of the project plan
A9.2. Start the simulation of the project plan</td><td></td></tr>
<tr><td>A10. Monitoring the execution of the project plan</td><td>A10.1. Review the Earned Value Analysis indicators
A10.2. Review the progress of the project in terms of tasks completion, duration and budget
A10.3. Review the motivation of the work team
A10.4. Review risks status</td><td rowspan="2">P3. Direct project work
P4. Control project work
P5. Control changes
P13. Control scope
P17. Develop project team
P18. Control resources
P19. Manage project team
P23. Control schedule
P26. Control costs
P29. Treat risks
P30. Control risks</td></tr>
<tr><td>A11. Control the execution of the project plan</td><td>A11.1. Evaluate project progress with respect to the project plan
A11.2. Identify the problems, deviations or risks that could be affecting the adequate progression of the project plan
A11.3. Make decisions (according to the game set of actions) to correct the potential deviations against the project plan with the goal of ending the project within the time, cost and quality established</td></tr>
</table>

Once the players have ended the Onset stage, ProDec generates the simulation model that simulates the execution of the project plan and players can start its execution (*A9*). During the Execution stage, players have to direct, monitor and control the simulation of the execution of the project plan in order to success the project scenario (*A10* and *A11*). Therefore, they have: (a) to review and evaluate the progress of the project simulation against the project plan; (b) to identify the problems, deviations or risks that could be affecting the adequate progression of the project plan; and (c) to make decisions for correcting the potential deviations.

As we can observe in Table 5, through the activities of this stage, ProDec is able to cover the "*Direct project work (P3)*", "*Develop project team (P17)*" and "*Treat risks (P29)*" processes of the ISO 21500, which are related to the Implementing process group of the standard. In addition, ProDec allows players to take contact with the following processes of the ISO 21500 Control process group: *"Control project work (P4)", "Control changes (P3)", "Control scope (P13)", "Control resources (P18)", "Manage project team (P19)", "Control schedule (P23)", "Control costs (P26)"* and *"Control risks (P30)"*. Therefore, the Execution stage of the gameplay's lifecycle process of ProDec allows covering processes belong to the Implementing and Controlling process groups of the ISO 21500 standard.

3.4 End Stage

The last stage of ProDec gameplay's lifecycle process is the End stage. In this stage of the gameplay, players should perform three main activities with the goal to end the game (see Table 6).

Table 6. Coverage of ISO 21500 processes by the activities of the ProDec's End stage.

Activities of the end stage		ISO 21500 processes
A12. Close the simulation of the project	A12.1. Accept the completion of the project plan	P6. Close project phase or project
A13. Collect the lessons learned	A13.1. Get the assessment report of the gameplay A13.2. Analyze the assessment report to get the lessons learned	P7. Collect lessons learned
A14. End the gameplay		P6. Close project phase or project

First, players have to accept the completion of the project plan in order to end the simulation of the project (*A12*). Once they accept to close the simulation of the project, ProDec generates an assessment report that allows players to get the lessons learned from their performance during the game (*A13*). Finally, after getting the assessment report, players can close the project scenario, ending, at the same time, the gameplay (*A14*).

Through the activities of this stage, as Table 6 shows, ProDec is able to cover the *"Close project phase or project (P6)"* and *"Collect lessons learned (P7)"* processes of the ISO 21500, which are related to the Closing process group of the standard. Thus, the End stage of the gameplay's lifecycle process of ProDec allows covering the Closing process group of the ISO 21500 standard.

4 Conclusions and Further Works

Bringing professional practice into the learning/teaching process is an especially difficult task in the context of software project management and can turn into a challenge in relation to software process standards education. According to many authors, the use of serious games, simulations and gamification strategies can help to overcome the difficulties and enable learners to acquire practical experience with real-life scenarios during their learning/teaching process.

The ISO 21500 standard provides generic guidance for project management and can be used by any type of organization, including public, private or community organizations, and for any type of project, irrespective of complexity, size or duration. In this paper, we have performed a systematic literature review to analyze the current works related to the use of games for understanding, teaching and supporting the education of the project management processes of the ISO 21500 standard. The results of our review allow us giving evidence about the lack of serious games for supporting the education of the ISO 21500 standard and considering this topic as a research opportunity.

For that reason, we have analyzed how ProDec, a simulation-based serious game, covers the project management processes of the ISO 21500 standard in order to assess its suitability to be integrated as a learning resource for understanding, teaching and supporting the education of the project management processes of the standard in the context of software projects.

The game, through the activities that are involved during the gameplay's lifecycle, is able to provide coverage of the five process groups of the ISO 21500 standard. Moreover, ProDec covers the 100% of the processes involved in the Integration, Scope, Resource, Time, Cost and Risk subject groups and the 50% of the processes involved in the Stakeholder subject group. Therefore, ProDec allows taking contact with seven of the ten subject groups of the ISO 21500 standard and covering almost 75% of the project management processes of the ISO 21500 standard. On the other hand, the current version of ProDec is not able to cover the processes involved in the Quality, Communication and Procurement subject groups and the process *Manage Stakeholders (P9)* involved in the Stakeholder subject group.

We believe that the use of this kind of learning resources is beneficial for learners and helps them to consolidate their knowledge. The coverage of the standard allows us to consider that ProDec can be a helpful learning resource to be used within the learning/teaching process of the ISO 21500 standard in the context of software projects. Nevertheless, more research and evaluations in this scope are needed to consider ProDec as a potential tool for software project management process education. For that reason, we are currently working to perform evaluations of the educational effectiveness of ProDec with both, university students and industrial practitioners, with the goal to get the necessary feedback to improve ProDec and integrate it as a learning resource to support project management processes education in the context of software projects.

Acknowledgements. This work was funded by the Spanish National Research Agency (AEI) with ERDF funds under projects amuSE (TIN2013-46928-C3-2-R), BadgePeople (TIN2016-76956-C3-3-R) and the Andalusian Plan for Research, Development and Innovation (grant TIC-195).

References

1. Clarke, P., O'Connor, R.V., Leavy, B.: A complexity theory viewpoint on the software development process and situational context. In: International Workshop on Software and Systems Process, pp. 86–90 (2016)

2. Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 5th ed., Project Management Institute (2013)
3. ACM/IEEE-CS Joint Task Force on Computing Curricula. Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press (2013)
4. ISO/IEC, ISO/IEC 12207:2008 - Systems and software engineering — Software life cycle processes (2008)
5. ISO/IEC, ISO/IEC TR 29110-1:2016 - Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 1: Overview (2016)
6. ISO, ISO 21500:2012 - Guidance on project management (2012)
7. Aydan, U., Yilmaz, M., Clarke, P.M., O'Connor, R.V.: Teaching ISO/IEC 12207 software lifecycle processes: a serious game approach. Comput. Stand. Interfaces **54**, 129–138 (2017)
8. Dorling, A., McCaffery, F.: The gamification of SPICE. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 295–301. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_35
9. Calderón, A., Ruiz, M.: Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game. In: Proceedings of SPICE, pp. 59–70, Dublin, Ireland (2016)
10. Kosa, M., Yilmaz, M., O'Connor, R., Clarke, P.: Software engineering education and games: a systematic literature review. J. Univ. Comput. Sci. **22**(12), 1558–1574 (2016)
11. AXELOS, Managing Successful Projects with PRINCE2®, The Stationery Office (2017)
12. Abt, C.: Serious Games. University Press of America, Lanhan (2002)
13. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Keele University and Durham University Joint Report (2007)
14. Mesquida, A.-L., Jovanovic, M., Mas, A.: Process improving by playing: implementing best practices through business games. In: European Conference on Software Process Improvement, pp. 225–233 (2016)
15. Calderón, A., Ruiz, M.: ProDec: a serious game for software project management training. In: Proceedings of the 8th ICSEA, pp. 565–570, Venice, Italy (2013)
16. Albrecht, A.: Measuring application development productivity. In: Proceedings of the joint SHARE, GUIDE and IBM Application Development Symposium, pp. 83–92, Monterey, California (1979)
17. Cattell, R., Eber, H., Tatsuoka, M.: Handbook for the sixteen personality factor questionnaire (16PF). Insitute for Personality and Ability Testing (1988)
18. Moder, J.J.: Project Management with CPM, PERT, and Precedence Diagramming, 3rd edn. Van Nostrand Reinhold, New York (1983)

SPI Case Studies

Exploration of a Practical Approach for Assessing the Measurement Capability of Software Organizations

Murat Salmanoğlu[1(✉)], Onur Demirörs[2,3], Ahmet Coşkunçay[4], and Ali Yıldız[4]

[1] METU, Ankara, Turkey
musalman@metu.edu.tr
[2] IYTE, Izmir, Turkey
demirorso@gmail.com
[3] University of New South Wales, Sydney, Australia
[4] Bilgi Grubu, Ankara, Turkey
{ahmet.coskuncay,ali.yildiz}@bg.com.tr

Abstract. Measurement is the foundation for successful software management. However, it is not easy for software organizations to evaluate their measurement practices and to determine what they should do to improve them. There are models to evaluate capability and maturity of measurement processes. However, they frequently focus on the measurement process in relation with a well-defined capability model like CMMI or SPICE. Organizations following recent agile methodologies do not desire to apply these holistic models. We have developed a model to assess measurement capability of software organizations by inspecting individual measures, independent from software development approach and process architecture. The model includes sample of core measures for aspects and defines generic practices for three capability levels. Organizations can use the model to determine and improve their measurement capability. In the paper, an exploratory case study conducted in a large telecommunication company is discussed and the results are evaluated.

Keywords: Software measurement · Measurement capability · Capability assessment · Software measurement improvement

1 Introduction

Software organizations can use measurement as a key for project success and an important enabler to identify improvement opportunities towards more productive processes. As improvement requires careful analysis of the current performance, continuous measurement of the processes, projects, and products is critical. Keeping measurement practices and activities up-to-date, and making sure that the measures are used correctly is a demanding task. Frequently, software organizations cannot cope with this demand due to the inherent difficulties of the products produced and the processes utilized [1]. Measurement assessment models give the organizations the ability to understand their current measurement capability, and provide a guideline on how to improve.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 415–429, 2017.
DOI: 10.1007/978-3-319-67383-7_30

In the software engineering discipline most products are one of a kind. Usually processes are not standardized and practices change with the rapid pace of technology. These properties force software organizations to set their own set of rules on what to measure, how to measure, and how to analyze the results. However, measurement domain knowledge is usually not part of an organizational know how. To ease this challenge several methodologies are developed to assess and guide improvement of organizational measurement practices [2–7]. Nevertheless, these methodologies are not widely utilized in practice. The problem is partially related with the inherent assumptions of the methods. The methods assume or require organizations to embrace the certain process improvement frameworks such as CMMI or SPICE. Although their benefits are widely accepted, these frameworks are criticized by organizations adopting agile methodologies as being heavyweight.

In this paper, we explore a new model that focus directly on individual measures, providing a flexible approach. Dealing directly with the measures eliminates the obligation to handle all organizational measures within the scope of measurement process. As a result, according to their needs organizations may focus on different measures in different areas with different levels of capability. Specifically, organizations utilizing flexible or agile development approaches can use this method without having to follow a holistic, process centric assessment framework.

The model also guides organizations to improve their measures to achieve higher capability levels. To be able to guide them, model groups the measures in several aspects. Each aspect is related with one area of software development life-cycle and includes a set of core measures. When an organization require to assess measures of a specific area, it should look at the core measures in the related aspect. In addition to these core measures, organizations can add their own measures according to their needs. Aspects are also grouped according to their effects on development. There are three groups: operational aspects, technical aspects, and strategic aspects.

The motivation to conduct this research started when a large scale telecom company requested to improve the measurement practices for its procurement process. The motivation and the initial works conducted are explained in [8]. Their measurement practices are analyzed to structure an improvement plan. As a generalization of this analysis and its findings, the model is structured. The details are given in exploratory case study section with the results of the application.

In the next section, a brief summary of literature about measurement maturity and capability is given. In the third section, the measurement capability model is explained and in the fourth section application of the model in an organization is described. Lastly conclusion section includes the results of the research, validity threads and future work.

2 Literature Review

Frequently utilized process improvement frameworks (ISO/IEC 15504 [2], CMMI [3]) include key process areas for measurement and define related practices. These frameworks are supported by well-established methods for assessment such as SCAMPI [9] for CMMI. In addition to these models and supporting assessment methods that covers the whole software development processes, there are studies that focus on assessing maturity of software measurement practices in organizations. These methods can be broadly classified in two main categories. The studies from the first category analyze utilization of the measures or supporting components of measurement process. The studies from the second category analyze the maturity of the measurement processes. First six models in this review belong to the first category, and the two coming after them belong to the second category.

First one is Daskalantonakis's [10] method for assessing software measurement technology. It is a maturity model developed specifically for measurement processes. It uses SEI's CMM as its foundation and defines five levels similar to CMM. These levels in ascending order are Initial, Repeatable, Defined, Managed, and Optimized. It has ten themes and each theme has definitions for maturity levels on a scale of one to five levels. The measurement technology maturity level is determined by assessing the conformance to the themes. Yes-no questions are provided to be used in assessment for each maturity level. A guide for evaluation for this maturity model is described by Budlong & Peterson [11]. The model takes CMM [12] as a basis, yet its main focus is the technology used in software measurement rather than the practices for measurement.

The approach introduced by Tarhan and Demirörs [13] provides an assessment process, model and tool [14] for evaluating metric's usability for statistical analysis in software organizations. The approach includes a standard set of usability attributes and metrics are evaluated with respect to the ratings given to these attributes in four scales. It is a significant study that focuses on evaluating metrics rather than assessing the capability of measurement practices.

Mendonca [5] proposed an approach to improve existing measurement frameworks. It uses data mining methods to find whether the metrics currently used include meaningful data that the users are not aware of. It compares existing measures with the organizational goals and try to determine whether the measures are necessary and whether there are enough metrics. They conclude that this methodology aims to understand the metrics and their capability to fulfil the needs. This methodology does not define a capability model for the measurements.

MeSRAM is a method aiming to assess robustness of measurement programs in organizations [7]. It includes a robustness model and an assessment method. The model defines a robust measurement program as: being able to incorporate a broad set of measures, having a support organization, and having a solid infrastructure. They pay special attention to the type of entities measured and proper definition of the measures. As MeSRAM measures the robustness or continuity of a measurement program it cannot be defined as a measurement capability method.

Practical Software and Systems Measurement (PSM) [15] defines a set of principles, best practices and techniques that takes part in tailoring, applying, implementing and evaluating activities of a project measurement process. Besides describing these activities of the measurement process, PSM provides case studies, measurement tables and indicator examples that supplement the process.

ISO/IEC 15939 [16] describes activities and tasks of a systems and software measurement process. A measurement information model and criteria for selecting measures also accompanies the process model definition.

The models in the first category do not aim to determine the maturity or capability of measurement process. Their focus are to determine maturity of measurement technology, usability of metrics for statistical analysis, capability of metrics to satisfy organizational goals, robustness of the measurement program, and defining a measurement process.

First method of the second category is Measurement-CMM [4], which is a model aiming to assess the maturity of measurement programs in an organization. It has a similar motivation to our research. They ask the questions: "How to introduce measurement in a software organization? What are the necessary steps to set up a measurement program and in which order should they be performed? How can existing measurement programs be enhanced?" To answer these questions, model suggests a 5 level maturity scale, in parallel with Software CMM [12], and gives measurement processes related with each of the maturity levels. They suggest that organizations should adopt Measurement-CMM together with software CMM. Although the model defines the levels and processes, the methods of assessment are not provided. The model also suggests new processes that are not defined in software CMM.

MIS-PyME [6, 17] is a model focusing on small-medium enterprises. It is formed with two main parts: measurement program definition methodology and measurement capability maturity model. It aims to align with 15504 [2] and provides SMEs a set of generic goals and indicators, a reference measurement process definition, a maturity assessment process definition and questionnaire, and a tool for supporting maturity assessments [18]. The work products provided by measurement program definition methodology are mainly: measurement goals table (i.e. process improvement goals required to implement improvement activities), indicator templates (i.e. guide user defining indicators for each goal), indicator database (i.e. successfully implemented indicator database). MIS-PyME measurement capability maturity model (MCMM) includes three main components: maturity levels and attributes that need to be fulfilled by measurement processes, assessment process aiming to determine the capability, and an interface with MIS-PyME methodology to define measurement programs. MCMM assumes higher maturity require three main conditions: a better established and performed measurement process, more ambitious goals to be measured, and establishing better support tools, procedures, and resources. It utilizes an ISO/IEC 15504 [2] based assessment methodology to determine the maturity.

Measurement-CMM and MIS-PyME belong to the second category, aiming to understand the organizational maturity level of measurement processes. However, there are some significant gaps in these models that we aim to fil with our proposed model.

Measurement-CMM doesn't explicitly provide the methods for assessment and the practices that belong to the key processes. It suggests to adopt software CMM together with Measurement-CMM. However, traditional process based improvement models alienate organizations that prefer agile approaches. These organizations demand adaptive models to improve specific points in their lifecycles according to their needs, instead of holistic approaches.

Similar to Measurement-CMM, MIS-PyME based its capability model on a process improvement model, ISO 15504 [2], and provide a reference measurement process definition. Aiming small and medium scale enterprises, it expects the organizations to implement defined attributes in their measurement processes. MIS-PyME also fails to satisfy the demands of organizations requiring an adaptive approach.

Considering the available literature, current methods mainly focus on improving measurement processes of the organizations by using common process improvement models as guidelines. It is hard for the organizations that doesn't want to use process oriented approaches to receive the intended benefit from these models. Lack of an up-to-date measurement maturity model answering the needs and demands of current development lifecycles creates a gap in the literature. In the next section proposed Measurement Capability Method, which aims to fill this gap, is explained.

3 Measurement Capability Method

Main purpose of the Measurement Capability Method is to present a practical approach to organizations to evaluate their measures. The method aims to fill the gap in the literature by providing a flexible approach focusing on measures instead of processes. With this method, organizations utilizing agile lifecycles can focus on specific measures according to their needs.

The method aims to guide the users by grouping the measures according to the phases of the development lifecycle. The building-block of the method is called as aspects, which include related core measures. Organizations are expected to work on all measures under an aspect to reach a certain capability. The term aspect is used by Ozcan-Top and Demirörs as "sets of interrelated and interacting activities" to represent traditional processes in an agile approach [19]. There are three aspect groups: operational, technical and strategic. The purpose of this general grouping is to provide a basic guidance to the organizations while deciding their aspects.

Organizations should evaluate their measures by using general practices defined in the model. A Measurement Capability Level is determined for each aspect according to the satisfaction level of the general practices. There are three capability levels: measured, institutionalized, and improved. These levels aim to help organizations to understand current capability of their measures, by knowing the current capability they can draw a path to improve their measures.

An organization can focus on a specific aspect by working on its defined core measures. Different core measures defined in different aspects may have commonalities especially for the derived measures.

Two main components of the model are explained in detail in the remaining part of this chapter. The first component being aspects, and second being capability levels and related practices. Organizations first need to determine which aspects they aim to assess and then proceed by checking the practices for each aspect. To be on a capability level, a measure need to satisfy all practices of the levels below and at least one practice of this level. Proving that its capability easily satisfies practices in previous level and capable of some practices in this level.

In addition to the two main components of the model, there are two additional concepts related with the method that are explained in following parts: measurement comparability and measurement capability assessment and improvement.

3.1 Aspects and Core Measures

There are three groups of aspects in the methodology:

Operational; includes aspects related with the operational workings of projects.
Technical; includes technical aspects adding value to the resulting products.
Strategic; includes aspects effecting strategic workings of organizations.

Aspects include different core measures that provide guidance to organizations to determine what they need to measure for selected aspect. All aspects and core measures are given in Table 1. Aspects that will be used to assess organizations measurement capability can be selected according to the organization's domains, needs, and goals. Total number and type of measures used in an organization may be quite large, in the model only a set of core measures are suggested. Additional measures can be defined according to the organizations needs and requirements.

Measures are also divided into two: direct measures and derived measures. Direct measures don't require another measure to be calculated, whereas derived measures are calculated by using other measures according to the formulas given in Table 1. Derived measures also help the organizations to identify the relations between different aspects by using the measures included in the calculation formula. Aspects can be chosen from all of the three groups or only from one, according to the needs of the organization.

3.2 Capability Levels

Measurement Capability Level 1 - Measured: In this level organizations conduct measurements but not necessarily uniquely in all units. Measurements are usually conducted as ad-hoc analysis, and the results might not be comparable among units and projects. Rules and applications usually stays in the specific project or team boundaries.

Measurement Capability Level 2 - Institutionalized: In this level organizations collect the same measures as the first level, yet, which measures to collect and how to collect them are defined in organizational level. The definitions may or may not be written. In addition, the results are analyzed systematically and performance comparison and benchmarking is possible. Rules and applications are usually same among projects and teams in the organization and measurement results are used as feedback in the project or team level. Organization follow defined measurement and estimation methods that are adjusted for the organizational needs.

Measurement Capability Level 3 - Improved: In this level, collected measures are evaluated and used to conduct statistical analysis. The results of the analyses are used to increase organizational efficiency and to modify available measures or to add new measures. Measurement results are used as feedback in organizational level. Organizations also need to be sure that the measures are stable and satisfy necessary attributes for statistical control [20].

For each measurement capability level their generic practices are defined in Table 2. Model doesn't provide any specific practices unlike models based on CMMI [3] and 15504 [2] because the model expects the organization increase the list of measures according to their needs, goals, and practices if necessary. Defining specific

Table 1. Aspects and Core measures

Group	Aspects	Core measures
Operational	Project	- Effort Direct Measures: Planned effort: Use an estimation method to plan required effort for project tasks before starting project Actual effort: Collect effort spent for tasks in individual project Derived measures: % of effort estimation efficiency: Total actual effort/total planned effort % of actual effort (at a specific time): Actual effort/Total planned effort
		- Duration Direct measure: Planned duration: Use an estimation method to plan required duration for project tasks before starting project. Actual duration: Record duration of actual project tasks Derived measure: % of duration estimation efficiency: Total actual duration/Total planned duration % of actual duration (at a specific time): Actual duration/Total planned effort
		- Cost Direct measure: Planned cost: Use an estimation method to plan required cost of project tasks before starting project. Actual cost: Record occurred cost of actual project tasks Derived measure: % of cost estimation efficiency: Total actual cost/Total planned cost % of actual cost (at a specific time): Actual cost/Total planned cost
	Risk	Direct measure: Anticipated risks: Identify possible risks before starting project. Occurred risks: Record occurred risks Unidentified risks: Risks that occurred but weren't identified at the start of the project Derived measure: Risk identification efficiency: number of anticipated and occurred risk/number of occurred risk
	Quality	Direct measure: Number of non-conformance: Record quality audits findings showing a non-conformance Derived measure: Costs of corrective actions: Cost of tasks to correct nonconformance Cost of preventive actions: Cost to perform quality audits

(*continued*)

Table 1. (*continued*)

Group	Aspects	Core measures
	Configuration	Direct measure: Configuration changes: Changes on the configuration elements
	Change	Direct measure: Proposed changes: Record proposed change per element Accepted changes: Record accepted proposed change per element Derived measure: Cost of change: Cost to implement changes
	Procurement	Direct measure: Procurement contract changes: Record changes in the procurement contract Quality of supplied work product: Asses quality of supplied product
Technical	Requirement	Direct measure: - Number of requirement change: Record changes in requirements Derived measure: - Cost of requirement change: cost to implement the requirement change
	Solution	Direct measure: - Number of design change: Record changes in design Derived measure: - Cost of design change: cost to implement the design change
	Test	Direct measure: - Number of defects: Record defect found during tests Derived measure: - Internal failure cost: cost of fixing defects discovered before the software delivered - External failure cost: cost of fixing defects discovered after the software delivered
	Integration	Direct measure: - Integration errors: Record number of integration errors
Strategic	Process improvement	Direct measure: - Process improvement proposals: Record process improvement proposals Derived measure: - Cost of quality: Costs of corrective actions + Costs of preventive actions + Internal and external failure costs
	Size	Direct measure: Size: Total size of the software Derived measure: - Normalized measures, rates: Project measure to be compared/project size

Table 2. Capability levels and generic practices

Measurement capability level	Generic practices
Level 1: MCL1: Measured	MCL1.GP1 Identify measures MCL1.GP2 Collect and store measures MCL1.GP3 Analyze measures MCL1.GP4 Communicate measurement to relevant stakeholders.
Level 2: MCL2: Institutionalized	MCL2.GP1 Plan and perform measurements according to a policy MCL2.GP2 Use measurement and estimation methods suitable for organizational needs MCL2.GP4 Define required sources and make them available to perform measurements MCL2.GP5 Assign responsibility to perform the measurements MCL2.GP6 Control products of the measurements MCL2.GP7 Identify the relevant stakeholders of the measurements MCL2.GP8 Monitor and control the measurements against the plan for performing measurements and take appropriate action MCL2.GP9 Evaluate adherence of the measurements against defined measurement descriptions. MCL2.GP10 Collect and store measurement related experience to support the future use
Level 3: MCL3: Improved	MCL3.GP1 Determine factors effecting measurement MCL3.GP2 Use organizational tailored estimation models MCL3.GP3 Use control charts to evaluate measurement activities MCL3.GP4 Use statistical evaluation to improve measures

practices for each measure will lessen the effectiveness and agility of the model by preventing organizations to add new measures. Generic practices are applicable to all measures.

3.3 Measurement Comparability

To be able to receive the most benefit from the measures, organizations should have the ability to compare their measures between projects or with different organizations. However; most measures cannot be compared directly as they depend to the project that they are collected from. Organizations need to use a metric that reflect the size of the project objectively and use that as a denominator for the other measures, to make them comparable. Although there are various alternatives that enable comparisons such as the functional size, lines of code, number of requirements, story points and total effort, most of them are not objective, repeatable measures. An objective representation of the software should be used as a size measure, suggested measure is functional size of the software [16]. This measure is also required to be defined, collected and analyzed as it belongs to an aspect.

3.4 Measurement Capability Assessment and Improvement

A gap analysis should be conducted in the organization to assess the capability level of the measures. A sample assessment process may include: an awareness seminar, selection of aspects to be evaluated, conducting evaluation survey, validation of findings, and reporting findings. After the assessment an improvement process may include an improvement plan, improvement implementation, pilot application, dissemination, and independent audits. During this research both assessment and improvement processes are tested with an exploratory case study, yet, the scope of this research paper does not include implementation process. In the exploratory case study chapter, assessment process applied in a real organization is given in detail.

During assessment, selected aspects should be evaluated by conducting interviews with practitioners of the processes and examining supporting proof. If a core measure is not used at all or does not satisfy all practices of capability level 1, then it will be graded with NS meaning not satisfied. It is named as NS instead of level 0 as it is not possible to assign a level if a measure simply doesn't exist. After all measures are assessed, capability levels of the aspects will be determined by using the levels of related measures. Aspects include measures that are related with each other and usually with similar capability levels, therefor an aspect's capability level will be closest level to the average of its measures. Findings will be communicated with the organization and opinions on the findings will be requested from the stakeholders for evaluation. After the validation, final report will be prepared with the capability levels of the aspects and then will be shared with the organization.

In the next section, the application of the method with the data from the sector leader GSM provider is given with the obtained results.

4 Exploratory Case Study

Creation of the proposed model started when the technology department of the largest GSM provider of Turkey wanted to find a comparable objective measure to be used in its procurement activities. The organization uses vendors for most of their software development. They make contract with vendors for a definite time without specifying both the scope and amount of projects for this time. During the contract period, their in-house analyst teams prepare the analysis document for the required projects and transfer them to vendors. Vendors implement the software, conduct tests and deliver the product.

To be able to analyze their processes and come up ideas to improve their measures in the procurement activities, initial idea of structuring a measurement capability model came to life. The exploratory case study includes the steps conducted in the organization and the findings of the final model by applying it retrospectively using data collected from the organization during the two-year duration of the research project. Although during the project all principles of the model applied in the organization, final model could be only applied after the project completion. Findings of the project is compared with the results of the model to verify them.

4.1 Assessment

The project started with an analysis of current measures and measurement artifacts of the organization. In this section project steps are explained by using references from the final model. Measurement capability level of the organization is given retrospectively by using organizational data collected during the gap analysis phase of the project.

Main purpose for the organization is to improve measurement activities of procurement processes. In addition to the procurement activities, their project management and development activities are also evaluated to find out the measures they are collecting and to analyze the quality of their artifacts used in data collection. As they didn't have any measure related with the size of the software, it is decided to investigate the possibility of implementing a size measure. The quality of the artifacts is important as they will be used to conduct measurements that will be determined in the improvement phase. Measurement capability model also includes size as an important aspect.

During the assessment, analysis documents from 453 projects from 8 different departments in the organization are evaluated by experienced measurers to understand their compatibility with size measurement. From the 453 documents analyzed, nearly half of them weren't fit for size measurement. Mainly they did not include necessary details of the requirements for a proper size measurement.

In parallel with the analysis of the documents, procurement activities of the organization were audited to understand their measurement capabilities. For their procurement process, they have key performance indicators (KPI) connected with their service level agreements (SLA). However, comparison between productivities of different vendors was not possible, as they couldn't objectively know the size of the products sent to the vendors.

When the current model applied to the organizational data from the start of the project, results are obtained as in Table 3. The output revealed that the organization is in capability level 2 for most of its aspects with some in level 1. Being in capability level 1 for most of technical aspects is not surprising as they are outsourcing most of their development needs. Their main focus was procurement process, being in level 2 in this aspect show that they are using related core measures in organization. However, they don't have the capability to compare these measures, as reflected in the size aspect.

The findings of the model are parallel with the results of the conducted gap analysis, i.e. if the organization implement a size measure that can be used as a comparability measure, it can use it to measure productivity for both project and procurement aspects. With the help of the comparison capability, they can improve general measurement capability and conduct statistical analyses for all of their measures.

4.2 Improvement

Although the phases related with the improvement activities are not directly related with the focus of this research, they are briefly explained to demonstrate the results of analyzing the measurement capability and determining improvement opportunities in the organization.

After the gap analysis, it is concluded that the needs of the organization can be satisfied by using the size of the software as a base measure to derive normalized comparability measures from the size of the software. This derived measures will mainly be the productivity of the in-house development teams, previous costs of vendors per unit size of the software, unit in-house cost of producing software, maximum unit cost of software that the organization should pay to the vendors, and defect rate per produced unit of software.

The method to be used for the size measure decided to be COSMIC Functional Size Measurement method. COSMIC uses functional processes to measure the size of the software. To able to measure the size of the software, analysis documents need to include enough data to understand functional process detail. One of the main output of the improvement step was to include use cases in the analysis document template.

Second improvement was to include high level representation of the software in the analysis document to understand which services are affected from the software. The third improvement is inclusion of sequence diagram into the analysis document. Sequence diagram is a useful visual representation for COSMIC measurement as it shows the data movements between different modules of the software.

These three improvement ideas for the documents are thought to be helpful for size measurement and in addition they should decrease misunderstandings between analysts and developers, resulting effort savings. The improvement ideas are reported to the organization and the project management processes are updated accordingly.

4.3 Results

After the implementation is completed, organization started to actively use COSMIC functional size in its projects to determine in-house productivity, vendor productivity, procurement cost comparisons, effects of requirement changes in the projects. They defined and conduct required steps related with the general practices for measurement capability level 1, 2, and 3. Their capability levels at the end of the project is given in Table 3. Currently they improved their capability level for project, procurement, and size aspects to level 3, Improved. They also improved nearly all of their aspects that were in level 1 into level 2.

Table 3. Measurement capability levels of aspects before and at the end of the project

Aspects	Core measures	Before the project		At the end of the project	
		Level	MCL	Level	MCL
Project	- Planned effort	2	2	3	3
	- Actual effort	3		3	
	- % of effort estimation efficiency	2		3	
	- % of actual effort (at a specific time)	2		3	
	- Planned duration	2		3	
	- Actual duration	3		3	
	- % of duration estimation efficiency	2		3	

(*continued*)

Table 3. (*continued*)

Aspects	Core measures	Before the project		At the end of the project	
		Level	MCL	Level	MCL
	- % of actual duration (at a specific time)	2		3	
	- Planned cost	2		3	
	- Actual Cost	3		3	
	- % of cost estimation efficiency	2		3	
	- % of actual cost (at a specific time)	2		3	
Risk	- Anticipated risks	2	2	2	2
	- Occurred risks	3		3	
	- Unidentified risks	2		2	
	- % of risk identification efficiency	2		2	
Quality	- Number of non-conformance	2	2	2	2
	- Costs of corrective actions	2		2	
	- Cost of preventive actions	2		2	
Configuration	- Configuration changes	2	2	2	2
Change	- Proposed changes	2	2	2	2
	- Accepted changes	2		2	
	- Cost of change	1		1	
Procurement	- Procurement contract changes	2	2	3	3
	- Quality of supplied work product	2		2	
Requirement	- Number of requirement change	2	2	2	2
	- Cost of Requirement change	1		1	
Solution	- Number of design change	1	1	2	2
	- Cost of design change	1		1	
Test	- Number of defects	2	1	2	2
	- Internal failure cost	1		2	
	- External failure cost	1		2	
Integration	- Integration errors	1	1	1	1
Process improvement	- Process improvement proposals	2	2	2	2
	- Cost of quality	1		2	
Size	- Size	NS	NS	3	3
	- Normalized measures, rates	NS		3	

5 Conclusion

This paper proposes a measurement capability model and gives results of an exploratory case study conducted with the leading GSM provider of Turkey to analyze and improve their measurement capability.

The organization is still using the measurements and policies defined at the end of this research and continue to improve their measurement practices and statistical estimation models as defined.

Main contribution of this research is a proposition of a measurement capability model with its assessment method. Organizations can use this model to determine the capability of their measures, to identify improvement opportunities and to improve their measurement policies.

Although the exploratory case study shows promising success, main threat to validity is having only one retrospective case study to test the model. To resolve this threat, the authors are currently conducting a second case study and planning two additional case studies with industry partners.

References

1. Tarhan, A., Demirors, O.: Apply quantitative management now. IEEE Softw. **29**(3), 77–85 (2012)
2. ISO/IEC, ISO/IEC 15504-5:2012 - Information technology – Process assessment – Part 5: An exemplar software life cycle process assessment model (2006)
3. CMMI Product Team, CMMI® for Development, Version 1.3 CMMI-DEV, V1.3, November 2010
4. Niessink, F., Van Vliet, H.: Towards mature measurement programs. In: Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering, pp. 82–88 (1998)
5. Mendonca, M.G., Basili, V.R., Bhandari, I.S., Dawson, J.: An approach to improving existing measurement frameworks. IBM Syst. J. **37**(4), 484–501 (1998)
6. Díaz-Ley, M., García, F., Piattini, M.: MIS-PyME software measurement capability maturity model - Supporting the definition of software measurement programs and capability determination. Adv. Eng. Softw. **41**, 1223–1237 (2010)
7. Staron, M., Meding, W.: MeSRAM – A method for assessing robustness of measurement programs in large software development organizations and its industrial evaluation. J. Syst. Softw. **113**, 76–100 (2015)
8. Salmanoglu, M., Öztürk, K., Bağrıyanık, S., Ungan, E., Demirörs, O.: Benefits and challenges of measuring software size: early results in a large organization. In: 25th International Workshop on Software Measurement and 10th International Conference on Software Process and Product Measurement, IWSM-Mensura 2015 (2015)
9. SCAMPI Upgrade Team, Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document. Software Engineering Institute, Carnegie Mellon University (2011)
10. Daskalantonakis, M.K., Yacobellis, R.H.: A method for assessing software measurement technology. Qual. Eng. **3**(1), 27–40 (1990)
11. Budlong, F., Peterson, J.: Software Metrics Capability Evaluation Guide (1995)
12. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model SM for Software, Version 1.1, Pittsburgh (1993)
13. Tarhan, A., Demirors, O.: Assessment of software process and metrics to support quantitative understanding. In: Cuadrado-Gallego, Juan J., Braungarten, R., Dumke, Reiner R., Abran, A. (eds.) IWSM/Mensura -2007. LNCS, vol. 4895, pp. 102–113. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85553-8_9

14. Kırbaş, S.: An Assessment and Analysis Tool for Statistical Process Control of Software Processes. Middle East Technical University (2007)
15. DoD, & US Army. PSM: Practical software and systems measurement – a foundation for objective project management version 4.0c. Department of Defense and US Army (2000)
16. ISO/IEC, ISO/IEC 15939:2007 - Systems and software engineering – Measurement process (2007)
17. Díaz-Ley, M., García, F., Piattini, M.: Implementing a software measurement program in small and medium enterprises: a suitable framework (2008)
18. Díaz-Ley, M.: Measurement framework for the definition of software measurement programs in SME's: MIS-PyME, Universidad de Castilla-La Mancha (2009)
19. Ozcan-Top, O., Demirörs, O.: A reference model for software agility assessment: AgilityMod. In: Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2015. CCIS, vol. 526, pp. 145–158. Springer, Cham (2015). doi:10.1007/978-3-319-19860-6_12
20. Tarhan, A., Demirörs, O.: Investigating suitability of software process and metrics for statistical process control. In: Richardson, I., Runeson, P., Messnarz, R. (eds.) EuroSPI 2006. LNCS, vol. 4257, pp. 88–99. Springer, Heidelberg (2006). doi:10.1007/11908562_9

SPICE in the Real World: Success for Large Infrastructural Projects with ISO/IEC 15504 Part 6

Dirk Pfauder[1], Tomas Schweigert[2(✉)], and Paul Hendriks[3]

[1] Pfauder Management & Consult BV, Utrecht, The Netherlands
dpfauder@xs4all.nl
[2] SQS AG, Stollwerck Str. 11, 51149 Cologne, Germany
Tomas.Schweigert@sqs.com
[3] SQS Nederland, Orteliuslaan 889, 3528 BE Utrecht, The Netherlands
Paul.Hendriks@sqs.com

Abstract. Since 2005, the Dutch Infrastructure Authority Rijkswaterstaat (RWS) is changing the way of tendering huge infrastructure contracts away from traditional projects to PPP (Public Private Partnership) projects. In such PPP Projects a company/consortium has to deliver the design, build, finance and maintenance of large infrastructural works. These projects are challenging and demand a mature management system. To check the maturity of these management systems, RWS asks for frequent external independent assessments using ISO/IEC 15504 Part 6, Systems Engineering and defines tough roadmaps for process improvement, which require process capabilities at the start of the project that cannot be testified due to the structure of the capability model. This paper explains the problem and potential solutions of this issue.

Keywords: DBFM · PPP · SPICE · ISO/IEC 15504 · ISO/IEC 15288 · Assessment · Improvement · Rijkswaterstaat · Infrastructure · Systems engineering

1 The Dutch Approach for PPP Projects

In contrast to public tenders, where the contracting authority establishes the implementation detailed in a program of requirements, RWS is no longer interfering in PPP projects with the content but manages the project completely on the desired end goal (the 'output'). In this way, market parties have more freedom in realizing the project ('input'), and, besides, RWS makes use of the innovative strength of the market. Furthermore, PPP projects can often be executed cheaper and faster than standard contracts, in particular because RWS saves on overhead in the design and planning process and risks have to be managed by the party who can do it best.

The most common forms of PPP projects in the Netherlands in road infrastructure (and other programs of RWS, like their 'Lock program') are so-called DBFM-contracts (Design, Build, Finance and Maintain) in which the market party is responsible for the process from design phase until and including maintenance.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 430–437, 2017.
DOI: 10.1007/978-3-319-67383-7_31

One of the first examples of a DBFM project is the ‘Tweede Coentunnel’ in 2005. Since then, many infrastructural projects (and ongoing tenders like ‘Afsluitdijk’) in the Netherlands are or will be subject to the PPP approach with a DBFM contract (see [1]): e.g. highways ‘Schiphol-Amsterdam-Almere (SAAone)’, ‘A9 rearrangement Badhoevedorp’, ‘A12 Lunetten Veenendaal’,‘A15 Maasvlakte-Vaanplein’, traffic intersection ‘A27/A1 Utrecht-knooppunt Eemnes-Amersfoort’, ‘Sealock IJmuiden (OpenIJ)’, and various river locks.

Typically, PPP projects are contracted with an SPC (Special Purpose Company) and executed by an EPC (Engineering Project Company) and MTC (Maintenance Company). One or more construction companies found the SPC. In addition, one or more banks are involved to provide the financial capacity necessary to deliver the project and to maintain the infrastructure asset. During the transition phase, the EPC transfers its responsibility to the MTC.

An important principle of a DBFM-contract is that it is based on availability of the infrastructure. When a project is realized ahead of schedule, the contractor obtains a bonus. When a project delivers beyond the final deadline, the contractor will receive a fine. The entire process of design, construction, financing and maintenance for 15 to 30 years is completely outsourced to a single party, which realizes synergy benefits. This allows for an acceleration in the design, implementation and maintenance phase at lower costs.

The contract duration is generally between 15 and 30 years, and often between 20 and 30 years. The lifetime of the realized project is usually longer than 10 years. This means that during the contract term maintenance is executed two or three times, so that the infrastructure can be transferred in perfect condition to the client (RWS) at end of contract.

Combining private and public qualities also means that risks (and opportunities) are divided and/or shared, otherwise than in a classic public approach. The more RWS addresses risks themselves, there will be less appeal to the qualities of the private parties, even if they (the contractor) are able to manage those risks best. By sharing risks and assigning them to parties who can manage them best, lower life-cycle costs and better value for money will be achieved.

Thinking about risks delivers almost automatically a movement towards a life-cycle approach. This is another key feature of PPP: the long-term relationship between public and private whereby the management and maintenance phase of a project increase significantly in importance.

Different tasks and responsibilities that are traditionally in the public sector, in this way transfer to a private party who receives a periodic fee for a long contract period. As such, the client (RWS) receives a service rather than a product. Nevertheless, RWS uses a traditional tender procedure to decide on contract award. In the decision making process the ‘Beste PKV’ criteria (Beste Prijs-Kwaliteitverhouding; Best Price-Quality Ratio) are used to rank offerings. This makes it most likely that the best and not the cheapest solution is chosen.

1.1 Management 'on Distance'

How does RWS manage the project 'on distance'?

- *System-based Contract Management* - RWS monitors the quality of the product to be delivered using a quality management system. The contractor must demonstrate that he meets the requirements of RWS. RWS then applies System-based Contract Management (SCM) to audit the management system and its outcome.
- *Apply Standard ISO 15288 and 15504* - The standard chosen by RWS is ISO 15288 and ISO 15504 to meet the required level of capability regarding various processes. Therefore, the life cycle approach of the project is covered by this standard [2, 3].
- *Common Infrastructure to Share Information* - RWS and the supplier use a common infrastructure, normally a Relatics® Database or similar, to share information. This allows for several checks regarding progress and quality.

1.2 PPP, DBFM and ISO 15504 in Practice

Obviously, both parties (RWS as the client and the contractors) had to learn to deal with this type of contract. There were successful projects and less successful ones.

One of the biggest reasons for a DBFM project not to be successful, could be the urgent need of the contractor for a project and, at the same time (or therefore?), not estimating risks correctly. For example, unfamiliarity with the means of the contract and underestimating or uncertainty of risks, together with a not very cooperative attitude of both parties (client and contractor did not really trust each other, and were looking constantly to proof that they were right, instead of building a project together) can and will cause serious troubles. 'A15 Maasvlakte-Vaanplein' is a good example for this. Contractors lost serious amounts of money, which almost caused bankruptcy. Even though the management system was based on ISO 15288 and 15504. In contrary to other projects the standard was used literarily in this A15 project, as a result of which the employees did not recognize their own processes anymore.

Another example (almost at the same time) was the reconstruction of the A12 Lunetten (near Utrecht) to Veenendaal. Both parties realized that delivering a successful project could only be done, if they cooperate in all openness and trust. Approaches were discussed together, uncertainties were shared and mistakes were admitted and reported by the contractor even if the client did not (yet) notice it. Transparency and mutual trust were the key success factors for this successful DBFM contract, which is also internally by RWS seen as a good example of how it should be done. The contractor used their own management system and adjusted it to 15288/15504, but their own system was leading. A cross reference list created the link between the management system and the standard 15288/15504.

2 Challenges for Process Improvement Roadmaps and Capability Assessments

In an assessment, the assessor or the assessment team evaluates the process capability using the selected process assessment model and collects evidences for process performance and process capability. Looking at the capability model of SPICE a level one process is precondition for a level 2 process, which in turn is precondition for level 3 and so on. So far so good in theory. Because of this approach, a process that is not (yet) executed can never reach level 3 or 4. This does not cause a problem for the assessment team, as they have to deliver a true report. However, it causes contractual problems if RWS demands processes to be assessed at a certain level that have not (yet) been executed.

Huge infrastructure projects with up to one billion Euro budget need a very mature management system in order to deliver successfully in time, in budget and of the desired quality. Therefor RWS decided to demand from the contractor a process improvement plan as precondition for the so-called 'Aanvangscertificaat' (the Certificate to Start). The roadmap of RWS for the improvement plan asks for capability targets like level 3 for engineering processes or level 4 for the measurement process. To illustrate the point clearly, the idea behind the roadmap is valid as poor management systems are an obstacle for successful delivery and even endanger the maintenance phase, which runs between 20 and 30 years.

The idea is clear: The supplier should constantly improve the management system during the execution of the project. However, the execution of a Disposal Process is extremely rare at the beginning of a project, or even before really starting to build.

In several projects, the supplier had discussions with RWS regarding requested process capabilities. As an example, the roadmap of one project requested level 2 for the transition process at the point of the Aanvangscertificaat. The problem: In this phase of a project, no transition takes place and no evidence for process performance can be found. Therefore, the capability is level 0 and a contract issue is born. The same issue challenges nearly every engineering and management process of ISO/IEC 15504 Part 6 at this project stage.

3 Options to Deal with This Challenge

Several options exist to deal with this challenge, but all of these options have their own pitfalls.

- **Option one:** *Reduce the Ambition of Process Capability* - From the assessor perspective, this seems to be a valid option, as the assessment team only needs to rate levels, which are backed by clear evidences. The team can deliver a valid report showing the accurate process capability from the SPICE perspective. For the supplier, success is achieved as the project company sticks to the given roadmap. For RWS this approach does not work. In a construction project, you have to perform management tasks a long time before construction tasks are executed.

From RWS perspective, the project is not transparent in a critical phase of the project. Thus, this solution is not acceptable.

- **Option two:** *Accept Level Zero Ratings* - From the assessor perspective, this seems also to be a valid option, as the assessment team only needs to rate levels that are backed by clear evidences. The team can deliver a valid report showing the accurate process capability level in this case: zero.
  However, this approach does also not work. The result causes an 'Afwijking' (deviation) record and a follow-up contractual issue from which neither RWS nor the project company have any benefit.
- **Option three:** *Deliver Process Profile Only* - From the assessor perspective, this also seems to be a valid option. The assessment team does not translate the process profile into levels and by delivering the process profile only, it shows that management Process Attributes like 3.1 are in place.
  The approach works a little for the supplier as the partners of the consortium – often called mother companies – run their own process or work package management system, which allows creating a showcase in providing a fast transfer. For RWS this approach is challenging. At the one hand, process descriptions often are available on the intranet. But, on the other hand, these are not (yet) used by the process performers. In addition, the construction of roadmaps becomes painful as for each milestone of the improvement roadmap the demanded capability has to be expressed as a process profile. This option may work, but is probably not first choice.
- **Option four:** *Adapt the Assessment Method to Project Situation* - Even if at the start of a project evidence for construction work is hard to find, the need for a check of the management capability can be solved by slightly changing the assessment method. In this approach the assessments starts with a dry run that checks the tailored process and project management methodology against the requirements of ISO/IEC 15504 Part 6. The certificate of this dry run proofs that the management system of the project company is capable to support the achievement of level 2, 3 or 4 as soon as the process is to be executed. The dry run contains: A check if the process description covers all aspects of the process in the PAM; A check if necessary tools and templates are available (support for 2.2.1 and 2.2.2); A check if necessary plans are available (e.g. Schedule, Resource plan) (Support for PA 2.1); and a check if mechanisms for process deployment are in place. As a result, the potential capability of the management system is evaluated and a valid report is delivered to the project company and RWS.
  This approach has the advantage that the project company becomes aware of weaknesses that might endanger the project success and is able to take immediate action in an early phase of the project. RWS has transparency over the true capability and can request effective actions from the supplier. As soon as the process is executed and evidences for process performance are available, it is also possible to check performance by rating the level by checking whether the potential capability as evaluated in the dry run is justified by actual capability. In case of deviations, an 'Afwijking' is recorded. A root cause analysis can be performed and immediate action has to be taken to solve the contractual issue requested by RWS.

Both RWS as well as contractors are learning from each project and learn to understand the standard ISO 15504 better and better. Some processes of the ISO 15288 are not relevant for this type of DBFM contracts. Sometimes the required level cannot be achieved. Unfortunately, the thought “the higher the level; the better the performance of the process” is sometimes still in use. In addition, the contractors have a better understanding of the standard and see that it will help to improve their performance. Management systems will make sure that the requirements of the standard are covered. You will see the learning effect; management systems are improved and better prepared for a good start of a project even already in the tender phase. A good example of this is the project ‘Sealock IJmuiden’. By using the experiences of previous projects, this project was able to make a quick start and was able to fulfil the requirements of the capability levels before ‘Aanvangsdatum’ within very short notice, and using a controlled approach.

4 Challenging Processes in a PPP Environment

As prescribed, the contractor of a PPP project is a project company founded by several construction and financial companies in order to execute the project. Part of the underlying contracts is the principle, that the project company uses resources provided by the mother companies. Sometimes the project company does a ramp up and then transfers operation e.g. of ICT infrastructure to the mother company. This situation is also a challenge for roadmap definition and assessment scoping. The problem addresses mostly to the following processes.

- **Project Portfolio Process** - As the SPC is a legal entity with the only mission to deliver the PPP contract, the assessment might lead to the following results: level 0 because the Project Portfolio Process is not executed by the SPC or level 4 as there is a decision to perform just one project. The best option would be to exclude this process from the scope of PPP projects.
- **Acquisition Process** - Rating the Acquisition Process at level 2 requires a fully implemented PA 1.1. This might work when one mother company founds the SPC and this company upfront purchases all goods and services needed to execute the project. However, even in that case it is not likely that there will be evidence for the following base practices: AGR.1.BP.6: Assess supplier performance; AGR.1.BP.7: Confirm product or service compliance; and AGR.1.BP.8: Closure of agreement. As a result, the process has to be rated largely at maximum for PA 1.1 meaning that level 2 will not be reached and a contractual issue is created. The situation is even more complicated, if the SPC is allowed to use third party services. The best way to deal with this issue might be to check the availability of the process during a dry run and assess the process when first contracts are completed. As the supplier will provide a project plan during the tender phase, it is also clear at which project milestone the acquisition process is up and running.
- **Supply Process** - Analyzing the Supply Process, it is obvious that an SPC is not capable to show this process to be at level 2 until the project work is delivered and transferred to maintenance. Some practices of the Supply Process are executed in

the tender phase where the SPC does not yet exist; other practices like AGR.2.BP.9: Transfer responsibility and knowledge, mark the end of the project phase. The best way to deal with this issue might be to check the availability of the process during a dry run.

- **Human Resource Management Process** - Depending on the project setup, it has to be decided whether the Human Resource Management Process can be assessed or must be placed out of scope.
  - *Option 1a:* The SPC does the HR. In this case, the capability requirement is valid and the process is part of the assessment scope.
  - *Option 1b:* The SPC does some HR (e.g. for administration staff) but most of the needed staff is delivered by the mother company. In this case, the capability requirement is valid and the process is in the assessment scope but reduced to the HR tasks performed by the SPC.
  - *Option 2a:* The mother company provides the complete HR and has committed to be involved in the assessment. In this case, the HR process is to be assessed at the mother company.
  - *Option 2b:* The mother company does the complete HR and has not committed to be involved in the assessment. In this case, the HR process should be placed out of scope and the delivery of HR is in scope of the acquisition process.
- **Infrastructure Management Process** - Depending on the project set up, the responsibility for the infrastructure (buildings, rooms, ICT, et cetera) is the responsibility of the SPC or of the mother company. If the SPC is responsible for the infrastructure, the capability requirement is valid and the Infrastructure Management Process is part of the assessment scope. In this case, even at the milestone 'Aanvangscertificaat' enough evidence is present for rating. If the SPC is not responsible for the infrastructure, i.e. the mother company delivers the infrastructure as a service, infrastructure is linked to acquisition and the infrastructure management process should be placed out of scope.
- **Project Assessment and Control Process** - Giving even a level 1 to this process indicates that technical tasks, in particular the implementation process, are executed. It is reasonable to assume that twelve month after 'Aanvangscertificaat', enough evidence is present to provide a meaningful rating for this process. But at the milestone 'Aanvangscertificaat', the SPC does not perform technical tasks. The SPC does the ramp up and the 'Aanvangscertificaat' shows that the ramp up is completed successfully. The best option is to postpone the level 2 requirement to the milestone 'Aanvangs-certificaat + 6 month'. At this milestone, there should be enough evidence to rate the process.
- **Measurement Process** - A good measurement approach is a cornerstone of successful project execution. It is likely to assume that the higher the capability level the better the process. As a result, RWS sometimes requires level 4 for the Measurement Process. It makes sense to require that the measurement process is managed properly. But it seems highly artificial to require KPI for the measurement process. But, for Level 4 there must be evidence that this KPI is defined, collected and used. The best option is to reduce the ambition to level 3
- **Stakeholder Requirements Definition Process** - PPP projects are contracted based on a tender. During the tender RWS gathers requirements from stakeholders and

defines the requirements for the requested infrastructure. As described above, the offer is normally delivered by the mother company. Consequently, there will be no additional evidence in the SPC regarding requirements definition. Several options are in place in this case: *Option 1:* Place this process out of scope; *Option 2:* Rate it using the offer as evidence; and *Option 3:* If other stakeholders than RWS are involved, rate handling of requirements of these stakeholders.

- **Transition Process** - For an SPC, transition marks the end of the only project the SPC is running. This means that at milestone 'Aanvangscertificaat' it is impossible to find evidence for the performance of this process. From the RWS perspective, this process is too important to be put out of scope of the assessment in the early phase of the project. Providing proof by a dry run will work.
- **Operation Process** - In many cases, the contractor will not operate some of the infrastructure assets delivered by the PPP contract. RWS does the operating. To the surprise of the assessor team, even if the delivered asset will be operated by RWS, the process is in scope of the assessment. Recommendation: if the contractor does not operate the infrastructure asset, it should not be part of the scope of the improvement roadmap.

5 Conclusion and Further Work

For the Dutch PPP projects, a mature management system is a cornerstone for project success. A consistent approach of describing the improvement roadmap and assessing its implementation is an effective support for the implementation and improvement of these management systems. ISO/IEC 15504 Part 6 and ISO/IEC 15288 deliver a toolset for the definition and the verification of improvement roadmaps.

When defining the roadmap for process improvement using process capability levels, proper attention must be placed on the question whether the process is relevant or not. If a process, (e.g. Operation) is not relevant it should not be in scope of the process improvement approach. In addition, careful analysis is needed at which milestone sufficient evidences will be available for rating. The authors are working on a standard roadmap that deals with the needs of RWS as a customer and the feasibility to implement and verify the roadmap.

References

1. Infrastructure PPP projects in the Netherlands. www.government.nl/topics/public-private-partnership-ppp-in-central-government/ppp-infrastructure-projects
2. Loon, H.: Concrete, steel and ISO 15288. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 269–272. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_29. ISBN 978-3-642-30438-5
3. Loon, H.: Using target process profiles in the real world. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 286–288. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_33. ISBN 978-3-642-30438-5

The Role of International Standards to Corroborate Artefact Development and Evaluation: Experiences from a Design Science Research Project in Process Assessment

Anup Shrestha[1(✉)], Aileen Cater-Steel[1], Mark Toleman[1], and Terry Rout[2]

[1] School of Management & Enterprise, Faculty of Business, Education, Law & Arts, University of Southern Queensland, Toowoomba, QLD, Australia
{anup.shrestha,aileen.cater-steel, mark.toleman}@usq.edu.au
[2] Software Quality Institute, Griffith University, Logan, QLD, Australia
t.rout@griffith.edu.au

Abstract. International standards were used to corroborate artefact development and evaluation in a Design Science Research (DSR) project within the context of Process Assessments in IT Service Management (ITSM). While there have been significant research efforts towards extending DSR guidelines and the development and revisions of the standards, reports of the application of International Standards to validate DSR artefacts are scant. DSR, akin to any academic research, is required to demonstrate rigour and relevance with the use of theories and prior knowledge. Moreover, DSR presents an artefact as a solution to a class of problems and reports how the artefact is developed and evaluated. Our DSR project demonstrated that concerns about the quality of artefacts can be addressed and thereby the utility and validity of the artefact can be verified with the use of International Standards. Using three International Standards, process assessment ISO/IEC 15504-33000 series, IT Service Management ISO/IEC 20000, and System and Software Quality Models ISO/IEC 25010, this paper presents an account of a real-life DSR project that demonstrates the significant role of International Standards to guide DSR researchers during artefact design, development and evaluation.

Keywords: International standards · Design science research · Process assessment · IT service management · Software quality evaluation

1 Introduction

The primary goal of a Design Science Research (DSR) method [1, 2] is to develop a new artefact. While DSR efforts focus on the features and functionalities of the artefact, research activities must be corroborated with some evidence that the artefact was built and evaluated rigorously. Without the validation of artefact design, development and

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 438–451, 2017.
DOI: 10.1007/978-3-319-67383-7_32

evaluation, the research contributions may not be highlighted and the artefacts could be viewed as merely unconfirmed propositions.

One of the key DSR requirements agreed by all schools of thought [3] is that the artefact development and evaluation must be validated using existing theories and guidelines. In a socio-technical context the artefact is influenced by the environment in which it operates. Previous DSR projects have used kernel theories [4, 5], case studies [6] and systematic literature reviews [7] for the corroboration of artefact design, development and evaluation. This paper advocates that guidance on how to validate the artefact build and evaluate cycles in DSR can be obtained from the standards belonging to the International Organization for Standardization (ISO) family, referred as the "International Standards" in the remainder of this paper. International Standards have been credited with facilitating communication in order to make information systems more consistent [8]. Since International Standards belong to the public domain and are universally applicable for transparent use [9], we assert that the use of International Standards promotes the validation of DSR artefacts during design, development and evaluation. Where applicable, DSR researchers may use available International Standards for transparency and consistency in the way research is conducted.

The role of International Standards in artefact validation can be demonstrated with its successful application to a real-life DSR project. Therefore, we report the design, development and evaluation of our artefact in which we used International Standards in a DSR project undertaken over four years (2011–2015). An iterative design process was followed to develop a research artefact for process assessment. Process assessment is a disciplined evaluation of an organisation unit's processes against a process assessment model (PAM) [10]. Our research artefact is named the "Software-mediated Process Assessment" (SMPA) approach that enables researchers and practitioners to assess ITSM processes in a transparent and efficient way. The four phases proposed in the SMPA approach include (a) assessment preparation; (b) online survey to collect assessment data; (c) measurement of process capability; and (d) reporting process improvement recommendations.

Three International Standards were implemented during the design, development and evaluation of the SMPA approach. The International Standard for ITSM ISO/IEC 20000 [11] provided the process reference model (PRM) for the processes to be assessed. The International Standard for process assessment ISO/IEC 15504 [12] provided support for a transparent assessment method. A decision support system (DSS) was implemented to demonstrate the use of the SMPA approach. The International Standard for Software Quality Evaluation ISO/IEC 25010 [13] provided the software quality in use model for the evaluation of the artefact that was conducted at two public-sector IT service providers in Australia. Evidence from the evaluation of the artefact indicated that the SMPA approach can be effective for process assessments [14].

The use of International Standards was a major driver in our DSR project to promote a transparent ITSM process assessment method. In this paper, we aim to report our research journey demonstrating how International Standards supported artefact design, development and evaluation, and thereby present a case for International Standards to be applied by relevant research communities for corroboration. The literature review of the DSR approach and the relevant International Standards is presented next. This is followed by a summary of our DSR project on ITSM process assessment.

In the following three sections, we present the SMPA artefact design, development and evaluation with key references to the International Standards used. We discuss our research experience highlighting the role that International Standards played in the successful execution of the project. Finally, we present the conclusion and direction for future work.

2 Literature Review

2.1 Design Science Research

Several DSR authorities have provided valuable guidelines related to the development and evaluation of artefacts that form the major activities and outcome of any DSR project. Baskerville [15] referred to the theory developed by design science as "theory discovery" where the theory is a by-product of the process of developing an artefact. In DSR the design process and resultant artefact have to be at least generalised to a class of problem domains [16]. This position corresponds to the definition of meta-requirements and meta-design provided by Walls et al. [5] in their proposed design theory. March & Smith [17] discussed design science and concluded that research artefacts may be constructed in the form of a construct, method, model, or instantiation. A major contribution of a DSR study should be to develop at least some components of a design theory. With the help of design theories, an artefact can address the identified research problems, present a novel solution to the problems and confirm the utility of the solution.

Over the past decade, the Information Systems research community has formalised DSR as an acceptable and rigorous research method. However, being a relatively emerging research method, DSR in information systems has limited resources that prompts the use of guidelines such as International Standards in order to validate artefact design, development and evaluation. Gregor & Hevner [18] proposed a DSR publication schema with guidance to present DSR projects. Their work provides significant insights in showcasing how DSR makes knowledge contribution and how to publish DSR work. Despite the significance of International Standards for policy making and in practice, there is a shortage of guidance on how to use International Standards for academic research, including for artefact development which is the major outcome of any DSR project [1]. Recent authors have suggested that future research must address the need for design principles that provide guidelines to identify the problem and proposed solutions relating to an artefact [19]. Consequently, current artefacts reported in DSR studies are variably validated [20]. Our work is motivated to address this challenge by demonstrating the use of International Standards to validate DSR artefact design, development and evaluation.

2.2 International Standard for Process Assessment

We used the International Standard for process assessment ISO/IEC 15504 during the design of our research artefact in 2013. This standard is currently being revised and transformed into a new standard family of ISO/IEC 33000 series [21]. Several parts of the ISO/IEC 15504 standard that were used in our research have now been withdrawn and new ISO/IEC 33000 standards have been published. Currently ten standards associated with the ISO/IEC 33000 series are published with many more under development [21]. The new standards present a generic view with a higher abstraction level for process assessment. Building new measurement frameworks and addressing quality characteristics other than process capability are two significant changes in ISO/IEC 330xx family. The ISO/IEC 33000 standard family also provides additional process assessment models and guidance in new areas. Readers can follow the Standards Catalogue on the ISO website for the latest standards update under the classification ICS code 35.080 IT > Software [22].

While there have been significant changes in ISO/IEC 330xx standard series (e.g. new concept of Process Quality Attribute in ISO/IEC 33001), the new standards correspond to related ISO/IEC 15504 content [21]. The measurement framework defined in ISO/IEC 15504-2 that was used in our research has been revised but remains similar to the new ISO/IEC 33020 standard. Moreover, the PAM used in our research – part 8 of the ISO/IEC 15504 – is still available and its transition into ISO/IEC 330xx family as ISO/IEC 33062 is expected to be straightforward [21]. The foundation of the ISO/IEC 15504 series has been subject of rigorous SPICE trials [23] and these have been published extensively [24]. In this light, we present an overview of the ISO/IEC 15504 standard to give a better perspective of the artefact since these standard parts were relevant during our project. We believe that the SMPA approach can be modified to meet the requirements of the new standards. In fact, the references made to ISO/IEC 15504 standards in our research can be viewed as a specific and valid instance of the ISO/IEC 330xx family in terms of the terminologies and the measurement framework. Therefore, the underlying concepts of the artefact, including the role of International Standards as highlighted in this paper, remain the same.

ISO/IEC 15504 defines six process capability levels: CL0 – Incomplete process; CL1 – Performed process; CL2 – Managed process; CL3 – Established process; CL4 – Predictable process; and CL5 – Optimising process. CL0 suggests a lack of effective performance of the process. At CL1, a single process attribute is defined. There are two specific process attributes defined for all the other process capability levels. Therefore, a total of nine process attributes (PA1.1 to PA5.2) exist in the measurement framework. At a more granular level, a number of explicit process indicators are defined for each process attribute. These process indicators provide criteria to assess process capability in finer detail. Process assessment is conducted in a standard manner when it is compliant with ISO/IEC 15504-2 requirements and where the assessors collect objective evidence against process indicators to determine capabilities of a process. ISO/IEC 15504 [12] suggests process assessment can be performed either as part of a process improvement activity or as part of a capability determination initiative.

2.3 International Standard for IT Service Management

The ITSM industry has defined a number of processes as best practices in the IT Infrastructure Library (ITIL®) framework. The British Standard BS15000 was developed based on ITIL in order to describe the ITIL processes in standard terms and more importantly to structure the ITIL processes in order to make them measurable and manageable [25]. Later, ISO/IEC 20000 based on the best practices of ITIL was published as the International Standard for ITSM. Since then it has undergone a number of updates and is currently synchronised with the latest ITIL 2011 edition [11]. ISO/IEC 20000 specifies requirements for IT service providers to develop and improve a service management system [26].

Part 1 of the ISO/IEC 20000 standard aims to support conformity assessment of the standard requirements in order to enable IT service providers to be certified based on a list of requirements that needs to be fulfilled [11]. This is valuable for a transparent method of an ITSM standard compliance audit. The ISO/IEC Standards Working Group responsible for ITSM (ISO/IEC JTC1/SC40) has also defined a PRM for the assessment of ITSM processes as Part 4 of the standard "that represents process elements in terms of purpose and outcomes" [27]. The PRM helps to identify activities required to check and maintain ISO/IEC 20000 compliance. In order to conduct standard-based process assessment, the PRM provides all the indicators to determine process performance at capability level 1 (CL1). The PRM for ITSM is scheduled to be renewed in line with ISO/IEC 330xx family in the coming years [21].

2.4 International Standard for Software Quality Evaluation

ISO/IEC 25010 is an International Standard that provides quality models for systems and software quality requirements and evaluation, also called SQuaRE, in the discipline of systems and software engineering [13]. Realising the growing adoption of software-as-a-service, the ISO/IEC 25010 standard was expanded in 2011 to include the quality in use dimension for software quality evaluation. A corresponding standard ISO/IEC 25040 [28] describes how the quality models from ISO/IEC 25010 can be used during the evaluation process. The quality in use is the degree to which software can be used by specific users to meet their needs to achieve goals in terms of effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use [13]. A standard definition of usability is provided in the quality in use model of ISO/IEC 25010, clause 4.2.4 [13]: "usability is defined as a subset of quality in use consisting of *effectiveness, efficiency and satisfaction*, [emphasised] for consistency with its established meaning". Furthermore, based on the standard, *satisfaction* is the user's response to interaction with the software and includes four sub-characteristics: *usefulness, trust, pleasure and comfort* [13].

3 Overview of the DSR Project

ITSM is an IT management framework that promotes service-oriented best practices to deliver value to organisations. The best practices are transformed into a summary of key requirements and guidelines for process improvement in the ISO/IEC 20000 standard. A major challenges in ITSM process assessment is the lack of transparency in the way ITSM processes are assessed. It has also been reported that existing process assessment methods are costly and time-consuming [29].

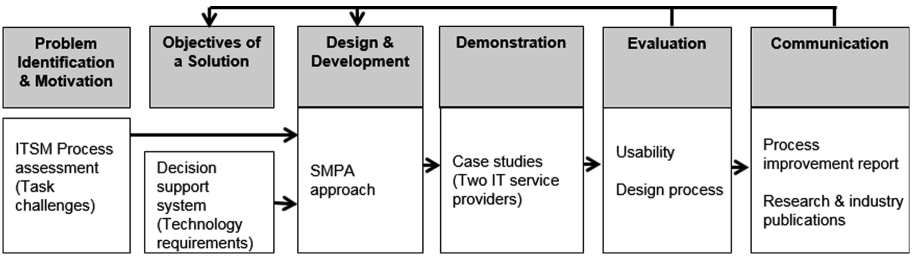

Fig. 1. DSR project methodology (adapted from [2])

Our DSR developed and evaluated the SMPA approach as the research artefact that is proposed to improve ITSM processes in a more transparent and efficient way than the current manual process assessment methods. The project draws on the DSR methodological guidelines for Information Systems (IS) research suggested by Peffers et al. [2]. Figure 1 presents our DSR project methodology.

The SMPA approach prescribes four phases to conduct ITSM process assessments. The first phase is preparation: information about the organisation profile, processes to assess and assessment participants along with their process roles are captured. The second and third phases survey the process stakeholders according to the ISO/IEC 15504 PAM and then measure process capability based on the survey responses. The final phase generates an assessment report that recommends process improvements for continual service improvement based on the ITIL framework. A comprehensive account of the SMPA approach has been reported previously [30].

The iterative nature of the artefact design process ensured that the final SMPA approach built after several "build-evaluate" cycles has utility and validity. Assessment goals were specified for each of the process capability levels. A number of assessment questions were related to specific assessment goals and the responses to the questions were calibrated with a metric of process knowledge. The SMPA approach addressed transparency issues in ITSM process assessment by following a goal-oriented measurement of ITSM processes using a standard PAM. With the background description of the DSR project, next we describe the artefact design, development and evaluation with reference to International Standards.

4 Artefact Design

In our research, the transparency issue with the ITSM process assessments is addressed with the use of International Standards. Using International Standards, the processes to be assessed are defined as structured activities in the PRM as Part 4 of the ISO/IEC 20000 standard [27]. ISO/IEC 15504 Part 8 was released as the PAM for ITSM [31]. These two International Standards have an interconnecting assessment framework and therefore they provide a transparent model for ITSM process assessments.

The ITSM environment is one where best practices and standards guide processes [11]. Therefore, introduction of a novel method that also conforms to International Standards plays a natural role in the acceptance of the artefact. Based on this premise, the SMPA approach is supported by the International Standards ISO/IEC 20000 and ISO/IEC 15504. Incorporation of widely accepted International Standards also provides justification of the iterative design of the SMPA approach.

Transparency can be demonstrated by aligning the assessment activities with the ISO/IEC 15504 standard that provides guidance on conducting the assessment process [10]. Part 2 of the ISO/IEC 15504 provides a measurement framework with capability rating metrics [12], however application of the framework to determine process capability is understandably not explicit in the standard. Perhaps this is because most of the assessment data analysis is largely dependent on the subjective judgment of the assessors which is based on their experience [32]. In cases where a software tool is used (e.g. SPICELite Assessment tool [33] and Appraisal Assistant [34]), the tools provide an interface to record evidence for standard indicators, rate process capabilities and produce assessment reports. There is limited discussion reported on how the collected assessment data is analysed, if it is done so, by any software tools.

Likewise, it is reasonable to assume that proprietary software tools and services in the ITSM industry such as PinkSCAN [35] and ITIL assessment services [36] also report alignment with the standard frameworks (ISO/IEC 15504, ITIL, CMMI) but they are silent about their data analysis approach due to their commercial value. During artefact design, International Standards were used for objective measurement as well as for the generation of the assessment results. The SMPA approach demonstrated transparency with the use of International Standards to not only collect data but also to perform gap analysis and determine process improvement guidelines.

5 Artefact Development

The SMPA approach uses ISO/IEC 15504 standard in order to exemplify a transparent method in ITSM process assessments. According to Part 2 of the standard that sets out the minimum requirements to perform an assessment, ITSM process assessment is based on a two-dimensional model: a process dimension and a capability dimension [12]. The process dimension is provided by an external PRM. Likewise, the capability dimension consists of a measurement framework comprising six process capability levels and their associated process attributes [12]. Process assessment is carried out utilising a conformant PAM that relates to the compliant PRM.

The base practices provided by ISO/IEC 20000-4 (process dimension) and the generic practices provided by ISO/IEC 15504-8 (capability dimension) were used to develop the questionnaire for each process. All the standard indicators, i.e. base practices for each process and the generic practices, were reviewed. Assessment questions for the survey were generated by analysing all standard indicators to construct singular, fine grained and close-ended assessment questions. The questions were then reviewed following the iterative design process to ensure industry relevance, standards alignment and academic rigour during their transformation.

The availability of the PAM for ITSM in ISO/IEC 15504 is one of the driving forces of this research. Although the combination of ISO/IEC 15504 and ISO/IEC 20000 was studied previously [37], there are few studies on the use of the combination for ITSM process assessment using the standard PAM. The standard PAM for ITSM [31] underpins the SMPA approach. Following this PAM, the SMPA approach provides a structured method to conduct process assessment in ITSM.

6 Artefact Evaluation

Artefact evaluation is necessary to confirm the validity of the contributions of the artefact. Evidence of utility of the artefact assures DSR researchers that the contributions of the artefact are applicable. The evaluation strategy advocated by Venable et al. [38] was used for evaluation. Using the quality models from the International Standard for software quality evaluation ISO/IEC 25010, the usability and outcomes of the SMPA approach were evaluated using quality factors for use of software.

The SMPA approach was evaluated with focus group discussions of SMPA survey participants and one-on-one interviews with the assessment facilitators at the two IT service providers. In order to assess if the SMPA approach has utility in a real organisation, it was essential to ensure that the survey approach was usable. Therefore, usability was determined as the key evaluation factor. The concept of usability as defined in ISO/IEC 25010 software quality in use model [13] was applied to evaluate five quality factors of the online survey. The standard definitions of the five software quality characteristics were transformed into operational definitions of usability characteristics to align their meaning to specific contexts of use. The data were analysed by reviewing focus group discussions and interview transcripts for themes or patterns related to the five software quality in use characteristics. The use of the International Standard for software quality in use model, ISO/IEC 25010, ensured that consistent terminologies were used during evaluation data collection and analysis.

As per the evaluation outcomes, participants reported that overall they found the online survey for assessment was trustworthy, comfortable and generally effective. Positive comments were also recorded regarding efficiency of conducting online surveys for assessments. However discussions led to a conclusion that a fully automated online survey that is strictly standards-based is not feasible and human input is critical for the facilitation of online assessment surveys. Regarding the use of International Standards, it was found that the PAM and guidelines based on ISO/IEC 15504 provided support to develop the SMPA approach that is more transparent than current ITSM process assessment methods.

7 Discussion

International standards provide requirements and guidelines that can be used consistently to ensure that processes are fit for their purpose. The International Standards referred to in this paper are developed and published on a voluntary but a fully consensus-based approach by independent bodies ISO and IEC that have national representatives of all United Nations member countries. International Standards, by their very nature, are powerful instruments of governance because of the effects their use can have on any activities undertaken. In terms of academic research, while

International Standards do not seek to guide any research activity, they can certainly provide valuable support towards the validation of research activities.

There are a number of important parallels between good research practice and good standardization practice. There is tremendous potential to use International Standards as part of good research practice, for instance, referencing International Standards in literature reviews and using standards to support research actions. Some of the key best practices of International Standards such as openness, transparency, effectiveness, global relevance, consensus and expert opinion [39] relate closely to good research practice. Therefore, researchers can understand and achieve the benefits of using International Standards to support their research activities. Moreover, International Standards embody universally agreed practices, drawing on the experience and expertise of all interested parties internationally. It is therefore plausible to assert that using International Standards promotes good research practice.

The two International Standards ISO/IEC 20000 and ISO/IEC 15504 are secondary data sources that were analysed in depth to extract information as input to develop the SMPA approach. The most relevant documents are the technical report Part 4 of ISO/IEC 20000, i.e. the PRM [27] and Part 8 of ISO/IEC 15504, i.e. an exemplar PAM for ITSM [31]. To the authors' knowledge, few researchers have studied the potential combined use of ISO/ IEC 15504 Part 8 and ISO/IEC 20000 Part 4, e.g. [40]. The choice of ISO/IEC 20000 and ISO/IEC 15504 is reinforced in this research in recognition of the credibility of the International Standards. It is logical to apply International Standard guidelines for evaluation after the experience of using International Standard guidelines for the development of the artefact. The International Standard ISO/IEC 25010 provides a software quality in use model [13] that was used to evaluate the usability of the SMPA approach.

The role of International Standards has been firmly established in greater adoption of ITSM process assessment [41]. For instance, Johnson et al. [42] demonstrated how consistent standards facilitate ITSM with an example of ITIL's configuration management process. Likewise, international IT standards can make the IT service transition less troublesome and help to streamline service operation [9]. It is therefore plausible to use a standard approach in process assessment (ISO/IEC 15504) and to apply such an approach to standard ITSM processes (ISO/IEC 20000) as both standards have been developed by the same organisations, ISO and IEC, thus fostering greater compatibility and global acceptance [43]. A standard and structured method provides the transparency required to compare outcomes and to measure improvements periodically. In addition, for multinational organisations a standards-based approach grounded on ISO and IEC specifications can make an assessment project feasible to conduct across global regions. The credibility of ISO and IEC is therefore one of the key drivers in this research.

Generally, standards provide statements of good professional practice, such as general principles rather than precise details of activities to be undertaken. Ironically, such an authoritative role of International Standards promotes transparency in the way activities are undertaken. The artefact in this research, the SMPA approach, provides prescriptive details of activities to be undertaken for ITSM process assessment. Nevertheless, since the artefact is scaffolded by the principles of International Standards, the support and validation of the prescribed activities is practical for industry use.

Two significant design issues were faced during the project. First, the process models of the International Standards for ITSM and process assessment were in a period of transition during the artefact development in this research. Therefore inconsistency was apparent in the way the process models were structured. The PRM for ITSM [27] was published as a technical report in 2010. This model was based on ITSM processes listed in the ISO/IEC 20000-1 standard published in December 2005. However ISO/IEC 20000-1:2005 was replaced with ISO/IEC 20000-1:2011 in June 2011 along with an updated set of requirements to maintain a service management system. A corresponding PRM based on the updated standard has not yet been published. A comprehensive review of PAM and PRM for ITSM process assessment has been planned within the standards community in the next few years [21].

Secondly, the measurement framework for process assessment is based on the International Standard ISO/IEC 15504-2 [12]. A new framework with updated metrics and assessment concepts is released in the ISO/IEC 33000 family [44]. As new sets of stable process models and standard guidelines are published, it is imperative that the research artefact is updated with changes to questions, calculations of process capability scores and recommendations for process improvement. However, we believe the overall SMPA approach is a valid method and the role of International Standards to ensure its validity remains.

With the expanding significance and reach of the newly published ISO/IEC 33000 standard series, the SMPA approach is expected to be a useful method for process assessments in any discipline that promotes a compliant assessment model. With our research experience in the process assessment and ITSM disciplines, we argue for the genuine contribution from International Standards towards the validity of DSR artefacts. In this light, we propose that International Standards can provide a suitable platform to validate the design, development and evaluation of a DSR artefact.

An overarching principle that governs the application of International Standards for DSR artefact corroboration is that all representations of artefacts (meta-artefacts) must be justified using prior knowledge. We advocate that International Standards are a reliable source of extant knowledge that can justify DSR meta-artefacts. Therefore, we present an application of International Standards from our DSR project, and illustrate its benefits for future DSR studies. Table 1 connects the relevant International Standards to our DSR project and thereby demonstrates how International Standards have justified our DSR initiatives.

The application of International Standards presented in this paper is aimed at promoting DSR transparency to guide researchers to demonstrate valid research work and the utility of research outcomes. It may seem that using International Standards to validate artefacts could place a burden on DSR researchers whose free- flowing innovation capabilities would be limited. However, we believe that the relevant International Standards can provide a solution to the majority of researchers who are concerned about developing and evaluating a worthy artefact. This resonates with one of the apparent causes of frustration in DSR that claims that DSR outcomes may not be derived from rigorous research work [20]. International standards may address some of the concerns about the quality of artefacts and thereby potentially increase the confidence DSR researchers have in the utility and validity of the final artefact.

Table 1. Mapping of International Standards to our DSR activities

International Standard	Key Role	DSR Activity	Mapping
ISO/IEC 20000	Address problems that the artefact can solve	Artefact Design	Provides a reference model of processes that needs to be assessed
ISO/IEC 15504 – 33000 series	State how the artefact was developed	Artefact Development	Provides the measurement framework and methodological guidance for process assessment
ISO/IEC 25010	Provide proof that the artefact is useful	Artefact Evaluation	Provides a software quality in use model to determine usability, based on ISO/IEC 25010 [13]

Another quality metric that relates to an International Standard's ability to validate artefacts is its ease of demonstration. Once researchers understand the structure of the International Standards, it is our expectation that they will find the standard sufficient and minimalistic to capture all information relating to an artefact that they must validate. For example, in our experience, building an assessment instrument from the ISO/IEC 20000 processes was simpler and more reliable than using the best practice guidelines from ITIL for assessment. Unlike ITIL as best practices for ITSM, ISO/IEC 20000 is minimalistic and tailor-made for assessment with a PRM. While ITSM assessments based on ITIL are certainly possible and more comprehensive, ITIL was designed to suggest improvements rather than assess quality levels. International standards provide a global, consensus-driven set of instruments for corroboration of research efforts, including DSR artefact validation.

In summary, a significant contributing factor to claim generalisation of the SMPA approach is the use of International Standards that provided a consistent structure to conduct process assessments and evaluate results. By developing clearer ways to assess ITSM processes based on International Standards, we hope that our research helps clarify unique challenges in process assessment activities and furthers our understanding of a consistent method to overcome such challenges.

8 Conclusion

Although artefacts represent the major deliverable in DSR projects, very little guidance and examples have been provided on how one can actually validate DSR artefact development and evaluation. This paper discusses the role of International Standards to validate an artefact in a real-life DSR project. The demonstration of this DSR project mapping with International Standards indicates that it is useful to validate artefacts.

From our experience of using International Standards for artefact design, development and evaluation, we believe that artefacts validated using universally- acceptable frameworks such as International Standards can potentially improve the way DSR projects are conducted. Future research can investigate how International Standards have been or could be applied in other DSR projects and whether this can promote validity in the way DSR projects are conducted. Another future research direction could examine previous DSR studies and catalogue International Standards that are used to validate DSR artefacts.

It can be argued that this example is only useful to very limited instances of DSR projects since it reports one artefact validated using three International Standards. However we argue that any DSR project that needs to validate the artefact can apply relevant International Standards where available and therefore, a more general view should be taken. Although we agree on the broader perspective, our intention for this paper is to give researchers an example of how a DSR project can be corroborated to showcase the validity of the artefact. The design knowledge developed in this research forms a base for subsequent research, implementation and evaluation that may contribute to such efforts as the trials for the International Standards for ITSM and process assessment. By trialling International Standards in industry, this research confirms that the standards are useful and supports the transition of new standards for effective industry use.

To conclude, this paper contributes to the IS community because prior work has not adequately addressed the role of International Standards in validating DSR artefacts. We have made a contribution to the growing body of guidelines for DSR with a practical example that demonstrates the role of International standards. The validation of carefully designed research artefacts has great potential to produce stronger IS design theories that may be valuable to both researchers and practitioners within and beyond the IS discipline.

References

1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
2. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. **24**(3), 45–77 (2008)
3. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on design-oriented information systems research. Eur. J. Inf. Syst. **20**(1), 7–10 (2011)
4. Iivari, J.: A paradigmatic analysis of information systems as a design science. Scand. J. Inf. Syst. **19**(2), 39–64 (2007)
5. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing information system design theory in perspective: how useful was our 1992 initial rendition. J. Inf. Technol. Theor. Appl. **6**(2), 43–58 (2004)
6. Van Aken, J.E.: The Nature of Organizing Design: Both Like and Unlike Material Object Design. Eindhoven Center for Innovation Studies: Technische Universiteit Eindhoven. The Netherlands (2006)
7. Carlsson, S.A., Henningsson, S., Hrastinski, S., Keller, C.: Socio-technical IS design science research: developing design theory for IS integration management. IseB **9**(1), 109–131 (2011)
8. Getronics: Implementing Leading Standards for IT Management. 1st edn. Van Haren Publishing, Zaltmobbel (2006)

9. Kumbakara, N.: Managed IT services: the role of IT standards. Inf. Manag. Comput. Secur. **16**(4), 336–359 (2008)
10. ISO/IEC: ISO/IEC 15504-1:2005 – Information Technology – Process Assessment – Part 1: Concepts and Vocabulary. International Organisation for Standardisation (2005)
11. ISO/IEC: ISO/IEC 20000-1:2011 – Information Technology – Service Management – Part 1: Service Management System Requirements. International Organisation for Standardisation (2011)
12. ISO/IEC: ISO/IEC 15504-2:2004 – Information Technology – Process Assessment – Part 2: Performing an Assessment. International Organisation for Standardisation (2004)
13. ISO/IEC: ISO/IEC 25010:2011 – Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models. International Organisation for Standardisation (2011)
14. Shrestha, A., Cater-Steel, A., Toleman, M., Rout, T.: Evaluation of Software Mediated Process Assessments for IT Service Management. In: Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2015. CCIS, vol. 526, pp. 72–84. Springer, Cham (2015). doi:10.1007/978-3-319-19860-6_7
15. Baskerville, R.: What design science is not. Eur. J. Inf. Syst. **17**(5), 441–443 (2008)
16. Winter, R.: Design science research in Europe. Eur. J. Inf. Syst. **17**(5), 470–475 (2008)
17. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decis. Support Syst. **15**(4), 251–266 (1995)
18. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS Q. **37**(2), 337–355 (2013)
19. Mandviwalla, M.: Generating and justifying design theory. J. Assoc. Inf. Syst. **16**(5), 314–344 (2015)
20. Weedman, J.: Client as designer in collaborative design science research projects: what does social science design theory tell us? Eur. J. Inf. Syst. **17**(5), 476–488 (2008)
21. ISO/IEC JTC1/SC7 WG10: Transition from ISO/IEC 15504 to ISO/IEC 330xx. Standing Document (2017)
22. ISO: Software Standards Catalogue. https://www.iso.org/ics/35.080/x/. Accessed 01 June 2017
23. Jung, H.W., Hunter, R., Goldenson, D.R., El-Emam, K.: Findings from phase 2 of the SPICE trials. Softw. Process: Improv. Pract. **6**(4), 205–242 (2001)
24. Rout, T.P., El Emam, K., Fusani, M., Goldenson, D., Jung, H.W.: SPICE in retrospect: developing a standard for process assessment. J. Syst. Softw. **80**(9), 1483–1493 (2007)
25. Malzahn, D.: Integrated product life cycle management for software: CMMI, SPICE, and ISO/IEC 20000. In: Cater-Steel, A. (ed.) Information Technology Governance and Service Management: Frameworks and Adaptations, pp. 423–442. Information Science Reference, Hershey (2007)
26. ISO/IEC: ISO/IEC 20000-2:2012 – Information Technology – Service Management – Part 2: Guidance on the Application of Service Management Systems. International Organisation for Standardisation (2012)
27. ISO/IEC: ISO/IEC TR 20000-4:2010 – Information Technology – Service Management – Part 4: Process Reference Model. International Organisation for Standardisation (2010)
28. ISO/IEC: ISO/IEC 25040:2011 – Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) - Evaluation Process. International Organisation for Standardisation (2011)
29. Peldzius, S., Ragaisis, S.: Usage of Multiple Process Assessment Models. In: Woronowicz, T., Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 223–234. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38833-0_20

30. Shrestha, A., Cater-Steel, A., Toleman, M., Rout, T.: Towards Transparent and Efficient Process Assessments for IT Service Management. In: Mitasiunas, A., Rout, T., O'Connor, Rory V., Dorling, A. (eds.) SPICE 2014. CCIS, vol. 477, pp. 165–176. Springer, Cham (2014). doi:10.1007/978-3-319-13036-1_15
31. ISO/IEC: ISO/IEC TS 15504-8:2012 - Information Technology - Process Assessment - Part 8: An Exemplar Process Assessment Model for IT Service Management. International Organisation for Standardisation (2012)
32. Barafort, B., Rousseau, A., Dubois, E.: How to Design an Innovative Framework for Process Improvement? The TIPA for ITIL Case. In: Barafort, B., O'Connor, Rory V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 48–59. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43896-1_5
33. HM&S. What is SPiCE-Lite? http://www.spicelite.com/cms/en/about-spice-lite.html. Accessed 01 June 2017
34. Rout, T.: Studies on the assessment process: usage of objective evidence in assessing process capability. J. Soft.: Evol. Process **22**(4), 297–305 (2010)
35. PinkElephant: PinkVERIFY™ - ITSM Tool Suite Assessment Service. https://www.pinkelephant.com/en-US/PinkVERIFY/PinkVERIFYCertification. Accessed 01 June 2017
36. Rudd, C., Sansbury, J.: ITIL® Maturity Model and Self-assessment Service: User Guide. AXELOS Limited, Norwich (2013)
37. Nehfort, A.: SPICE Assessments for IT Service Management according to ISO/IEC 20000-1. In: International SPICE 2007 Conference, Frankfurt, Germany (2007)
38. Venable, J., Pries-Heje, J., Baskerville, R.: A Comprehensive Framework for Evaluation in Design Science Research. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 423–438. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29863-9_31
39. ISO/IEC: Using and Referencing ISO and IEC Standards to Support Public Policy. International Organization for Standardization (2014)
40. Barafort, B., Di Renzo, B., Lejeune, V., Prime, S., Simon, J.: ITIL based service management measurement and ISO/IEC 15504 process assessment: a win-win opportunity. In: 5th International SPICE 2005 Conference, Klagenfurt, Austria (2005)
41. Hilbert, R., Renault, A.: Assessing IT service management processes with AIDA– experience feedback. In: 14th European Conference for Software Process Improvement, Potsdam, Germany (2007)
42. Johnson, M.W., Hately, A., Miller, B., Orr, R.: Evolving standards for IT service management. IBM Syst. J. **46**(3), 583–597 (2007)
43. Lepmets, M., Mesquida, A.L., Cater-Steel, A., Mas, A., Ras, E.: The evaluation of the IT service quality measurement framework in industry. Glob. J. Flex. Syst. Manag. **15**(1), 39–57 (2014)
44. ISO/IEC: ISO/IEC 33001:2015 – Information Technology - Process Assessment - Concepts and Terminology. International Organisation for Standardisation (2015)

Strategic and Knowledge Issues in SPI

The Impact of Situational Context on the Software Development Process – A Case Study of a Highly Innovative Start-up Organization

Gerard Marks[1], Rory V. O'Connor[2,3], and Paul M. Clarke[2,3](✉)

[1] Optimality Technologies, Invent, DCU, Dublin, Ireland
Gerard.Marks@optimalitytech.com
[2] Dublin City University, Dublin, Ireland
{Rory.OConnor,Paul.M.Clarke}@dcu.ie
[3] Lero, The Irish Software Research Centre, Dublin, Ireland

Abstract. Over the past six years, we have examined the impact of situational context of the software development process. Our early work involved the systematic development of a comprehensive situational factors reference framework. More recently, our efforts have focused on the application of this reference framework to different types of situational context. In this latest in a series of case studies, we examine the case of a small start-up organization, exploring in detail the process adopted. We also undertook a detailed evaluation of the situational context, carefully identifying the situational factors of greatest importance and how these factors have influenced the process design. The outcome of our case study confirms our earlier finding that a software development process is highly dependent on the organizational context. We also discovered some interesting new themes in this start-up environment, including the difficulty associated with prioritizing situational factors and the complexity that surrounds software process design. The role of organizational learning and feedback into improved development processes is also presented as a critical feature.

Keywords: Software development process · Software development context · Agile · Lean · Process selection

1 Introduction

Although many valuable software development models, methods and standards have been created, it remains the case that attempts to identify a universally optimal approach to software development have been frustrated by the variation that presents in software development contexts [1]. Added to the challenge introduced by this variation, we also find that no situational context is unchanging [2], with the result that process adaptation is inevitably required. These observations in relation to the software development process seem likely to meet with the agreement of seasoned software development researchers and practitioners. However, while agreement on the position may arise, the authors have suggested that the solution to the problem of harmonizing a

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 455–466, 2017.
DOI: 10.1007/978-3-319-67383-7_33

process with a context is highly complex, in fact it would appear to be an instance of a complex adaptive system [3]. In pursuit of an improved understanding of this complex interplay between a software development process and its situational context, we have assigned high importance to the evaluation of individual situational contexts and corresponding processes [4]. Accordingly, some of our related work has examined the problem in a high-growth small to medium sized organization applying a microservices architecture for rapid product evolution [5], and also in safety critical software development environments, including medical device and nuclear power domains [6].

In the case study reported upon in this paper, we focus our examination on a further development setting. This time, we examine the software development process adopted by a high potential growth firm that operates in the specialized database performance and interoperability niche. This firm has grappled with the challenge of satisfying the predictability demands of mission-critical data-intensive systems while simultaneously battling the survival concerns which are all too often a reality of small start-up organizations. Through examining the situational context and software development process in the case study organization, we identify the key constraints that have focused the software development process enactment. Together with earlier findings, this knowledge is helpful in building up a portfolio of context-to-process relationships.

While our work has proven to be iterative and slow in nature, it has a number of important benefits. Firstly, it can help us to better appreciate the relationships and dimensions that comprise this complex challenge. Individual organizations seeking an objective reflection on their software development process can tap this resource as an aid to self-evaluation. Secondly, over time, the development of a suite of case studies can identify commonalities and differences in different types of settings (and the impact this has on the development process), thereby collectively helping to reduce the process-to-situational-context harmonization challenge.

Section 2 outlines the situational factors framework; Sect. 3 presents an overview of the company studied, including its software development process; Sect. 4 examines the role of situational context; and finally, Sect. 5 presents a discussion and conclusion.

2 Situational Factors

Since at least 1992 (and probably much earlier) the importance of situational context as an informant of the software development process has been acknowledged [7]. Although published resources advocate that an "organization's processes operate in a business context that should be understood" [8] and that a "life cycle model… [should be] appropriate for the project's scope, magnitude, complexity, changing needs and opportunities" [9], we suggest that there remains a significant lack of guidance on exactly how companies might adapt their process to their (changing) situational context. Software development necessarily occurs in a development context, which includes a large number of concerns and factors [10, 11] with this context being pivotal in understanding what works for whom, where, when, and why [12]. In support of the importance of understanding the instructional function of situational contexts, authors such as Dyba [13] highlight that the dependence on a potentially large number of situational factors is of itself an important reason for why software engineering is so hard.

Despite the various references to the importance of situational context in the literature, it was the lack of a comprehensive situational factors framework for software development that led two of the authors to produce and publish an initial reference framework [14], itself an amalgamation of earlier important contributions, from multiple areas such as software risk estimation, cost models for software engineering, capability maturity frameworks, etc.

The framework incorporates 44 individual factors (refer to Fig. 1) classified under 8 categories (refer to Table 1), which are further elaborated as 170 underlying sub-factors. A sample listing of the sub-factors in the *Personnel* classification is presented in Table 2, with comprehensive details of the framework available in previously published material [14].

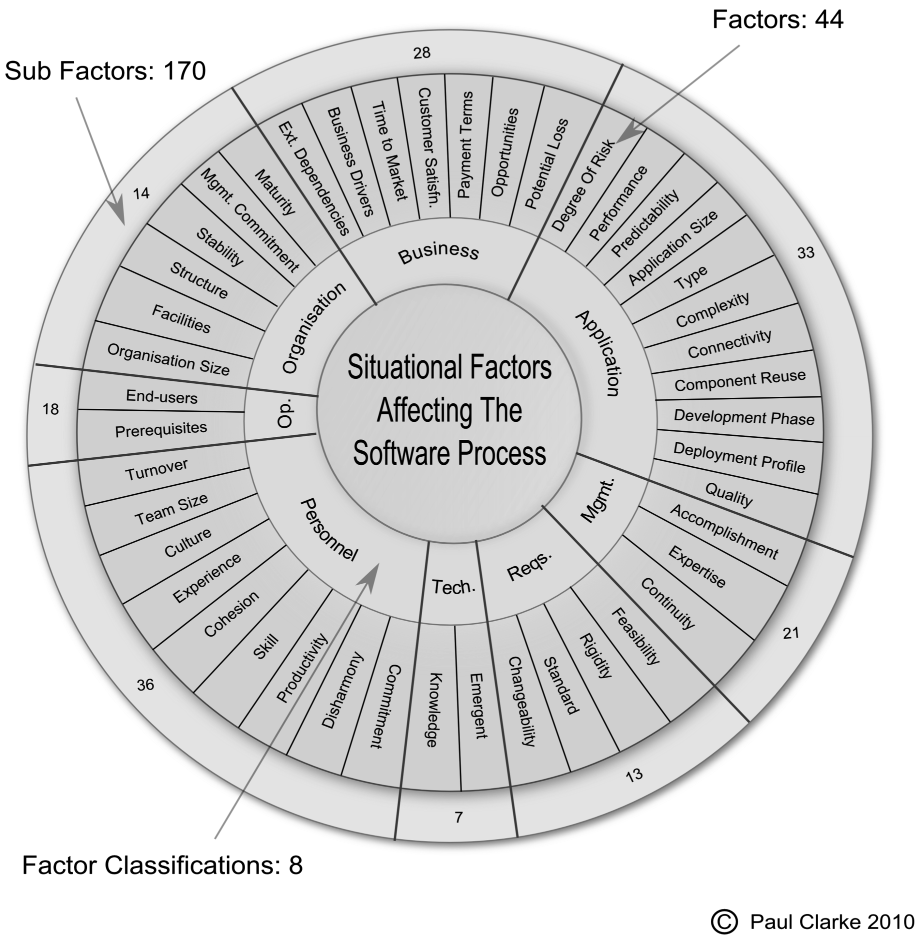

Fig. 1. Situational factors reference framework

Table 1. Situational factors classification

Classification	Description
Personnel	Constitution and characteristics of the non-managerial personnel involved in the software development efforts
Application	Characteristics of the application(s)
Technology	Profile of the technology being used for the software development effort
Organization	Profile of the organization
Operation	Operational considerations and constraints
Management	Constitution and characteristics of the development management team
Business	Strategic/tactical business considerations
Requirements	Characteristics of the requirements

Table 2. Personnel factors & sub-factors

Factor	Sub-factor
Turnover	Turnover of personnel
Team size	(Relative) team size
Culture	Team culture/resistance to change
Experience	General team experience/diversity/ability to understand the human implications of a new information system/team ability to work with management/application experience/analyst experience/programmer experience/tester experience/experience with development methodology/platform experience
Cohesion	General cohesion/team members who have not worked for you/team not having worked together in the past/team ability to successfully complete a task/team ability to work with undefined elements and uncertain objectives/overdependence on team members/distributed team/team geographically distant
Skill	Operational knowledge/team expertise (task)/team ability to work with undefined elements and uncertain objectives/training development
Productivity	Team ability to carry out tasks productively
Commitment	Commitment to project among team members
Disharmony	Interpersonal conflicts

The authors consider the situational factors reference framework to be a stepping stone towards greater understanding of the complexity of software development settings, and the systematic approach adopted in its creation from a rich and detailed set of sources has given rise to a framework that we consider to outline a broadly informed reference for the software development community [4]. Using the framework, the situational factors affecting the software process were investigated in practice in the case study start-up organization, details of which are presented in the following sections.

3 Case Study Company

Optimality Technologies is a spin-out company from Dublin City University (DCU), founded in July 2015. Prior to the formation of Optimality, a sustained research effort into XML query performance (focusing largely on data indexing and retrieval techniques and data processing algorithms) was achieved through the DCU-based Database Performance & Migration Group (DPMG). From its earliest roots as the DPMG, Optimality prioritized industrial engagement as a means to test assumptions and potential solutions.

During the lifetime of the DPMG, the sustained focus was the development of data-related products that reduced the complexity and risk associated with modernizing database infrastructure. Ultimately, a series of industrial and research engagements pointed to single, significant challenge: Could the database layer be optimized whilst preserving the application code base? If this challenge could be overcome in an economically-viable manner, there was a major immediate benefit for organizations: it would be possible to modernize the database infrastructure with zero application code rewrites. This became the focus of Optimality's efforts and over time, resulted in the development of a sophisticated tool set and associated interfacing software product, which has been applied to the task of database modernization. Perhaps the two most important advantages of this approach can be summarised as:

- **Reduced solution development time.** The result of zero application code rewrites is a significant reduction in the development effort.
- **Reduced risk of software failures.** Since the application-level test suites remain valid, it is possible to quickly detect any issues with the database modernization/migration effort.

In the view of Optimality, these two advantages increase the viability of database modernization for organizations, and this attracted the attention of leading global financial software product and service providers. In some cases, these organizations have harnessed the Optimality tool set to evaluate alternative or emerging database technologies prior to making large investment decisions for production-grade products and services. No matter the size of the organization with which Optimality engages, the constant challenge is to harmonize the objectives of strict correctness (i.e. data integrity is unaffected) and reliability (in terms of solution stability) with the need for high-paced innovation (sometimes involving emerging technology). To address this context, Optimality has adopted and refined a software development process that is particular to their organization.

3.1 Process Overview

Figure 2 illustrates the process lifecycle from the initial customer engagement through to an iterative system elaboration process, with further details of the individual steps being as follows:

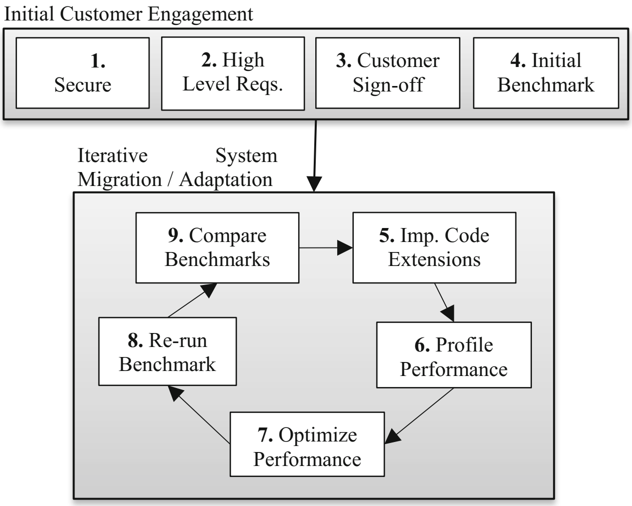

Fig. 2. Optimality – high level software process lifecycle

<u>**Initial Customer Engagement**</u>

- **Secure Contract.** New business acquisition.
- **High Level Requirements.** Evaluate the client's high level requirements and formulate a specification document along with projected milestones, deliverables and payment terms (which may be time and materials based, or fixed price). Since there is high variability in existing client systems and objectives related to innovation, it is not possible to fully elaborate requirements at this stage.
- **Customer Sign-off.** Once the customer has signed-off on the requirements and terms, work can begin.
- **Establish Initial Benchmark.** Performance considerations are key aspect of the work. Therefore, a specified benchmark system captures performance metrics prior to the implementation of any solution implementation effort.

<u>**Iterative System Migration/Adaptation**</u>

- **Implement Code Extensions.** If required, extensions to the Optimality tool set are implemented to enable the migration process (e.g. providing coverage for a new query language).
- **Profile Performance.** Evaluate the performance constraints and targets. Where appropriate, identify the most attractive cost-benefit work packages.
- **Optimize Performance.** Involves tuning the target database and Optimality's processing engine to ensure that performance is maximized.
- **Re-run Benchmark.** Rerun the benchmark to examine impact on performance and if required, confirm (using application-level tests) that migration effort has been successful.
- **Compare Benchmarks.** Evaluate the results of the benchmark, and liaise with the customer to determine if subsequent migration/adaptation iterations would be beneficial.

Table 3 provides an overview of the typical durations for each step of the process. Note that there is variance for each step duration, which allows for some small rapid changes to be introduced into a formal evaluation cycle if required.

Table 3. Estimated process duration overview

Process name	Duration (Days)
Implement code extensions	0–60
Profile performance	2–10
Optimize performance	2–60
Re-run benchmark	1–5
Compare benchmarks	1–3

3.2 Testing and Quality

In the earliest stages, the objective was to simply morph the Optimality tool set into whatever the client demanded. Gradually, this led to the development of an automated Extract, Transform, Load (ETL) process, whereby once a query is received (for example, from a user interface), it is redirected to the new data model (or database) and will reconstruct the result set into the format expected by the application layer. However, in enabling this automated interaction, constraints in relation to coverage and quality must also be satisfied.

3.2.1 Coverage

Optimality provides an *SQL-to-X* service where *X* can be: (1) a new data model within the same database, (2) another relational database, or (3) a *NoSQL* or *NewSQL* database. As a result, multiple dialects of SQL (e.g. Oracle, SQL Server, MySQL) must be supported which necessitates the need for a dual/hybrid database.

In the dual database scenario, a runtime *query routing* service enables a subset of tables to be migrated to the new database/model, while all others are routed to the original system, thereby allowing the fully functional software application to be redeployed in a very short time period. Since this approach can quickly isolate critical solution viability information, it has proven to be very effective in supporting the type of proof of concept required by many clients.

3.2.2 Quality

High data quality is a critical requirement for many database intensive systems, especially in sectors such as Finance. To satisfy this constraint, a number of quality related techniques were injected or emphasised in the software development process, including:

- Core algorithms formally verified at the theoretical level.
- Core functionality subject to robust unit testing.
- Continuous integration is adopted to protect against overall quality degradation.

Collectively, and although costly, the insistence on the adoption of these three techniques adequately addressed quality considerations. With very few exceptions, unit tests are written prior to the code itself being written. In the early stages, standalone unit tests were written for each core piece of functionality (a query transformation, for example). However, it became apparent that continuous integration (whereby test data is re-generated each time and queries are tested against each of the supported databases) was required.

3.2.3 Automating Continuous Integration

As a final degree of integrity checking, an automated *Integrity Checker* was developed. Given that the dual database approach was adopted to allow for iterative migration lifecycles, it is possible to execute the 'original' query against the 'original' database and the 'translated' query against the 'new' database and byte-compare the results at runtime (i.e. the process is entirely automated). Therefore, the Integrity Checker provides (1) a way for end users to validate the correctness of the system against multiple sources of *test data*, and (2) a means for end users validate the system against actual *production data* (at runtime). Together with other innovations such as the automated ETL process, the Integrity Checker effectively automates the creation of continuous integration tests. Were it not for this advanced form of automation, it would not be possible to sustain the pace of development while also satisfying the quality constraints.

4 Applying the Situational Factors Reference Framework

Two researchers in association with the Managing Director from Optimality analyzed the company's situational factors, the outcome of which was a listing of the dominant contextual factors affecting Optimality's software development process (refer to Table 4).

Table 4. Situational factors identified in case study

Category	Factors identified in case study
Personnel	**Skill:** Given the very high application and programming skill of both primary engineers, the team had a high velocity while also maintaining high quality – plus the start-up cost in terms of personnel on-boarding was low
Requirements	**Changeability:** Many requirements would only became clear through a sustained prototyping-type effort. Therefore, an agile/rapid prototyping approach was well suited to the nature of requirements
Application	**Quality:** There is a strict requirement for accuracy (i.e. high quality) of query-related tasks. This factor was a motivator for adopting test driven development (TDD) and continuous integration (CI);
Application	**Performance:** There was a significant requirement for very high performance from the Optimality software and as a result, regular investments in refactoring were needed in order to streamline performance;

(*continued*)

Table 4. (*continued*)

Category	Factors identified in case study
Application	**Complexity:** The high volume and complexity of data queries raised the complexity of the application overall. TDD and CI were instrumental in raising confidence that the complexity did not compromise the application quality;
Application	**Predictability:** Given the sometimes rapid pace of functional deliveries, a lean/agile software development philosophy was adopted. As the extent of recent changes could be high, the need for a process offering both robust refactoring and TDD/CI was very high;
Application	**Type:** A low tolerance for data inaccuracy influenced the decision to implement a robust TDD and CI infrastructure. The factor also had a direct impact on the software architecture. To permit 3rd parties to address different aspects of overall system functionality, parallelization allowed other systems to handle certain concerns
Operational	**End-Users:** End-users in this case were expecting responsiveness from their software supplier in pursuit of competitive advantages in a fast moving market. This fact is key in shaping much of the process design – which is capable of addressing rapidly changing requirements
Technical	**Emergent:** Aspects of the technology stack were emergent (e.g. the Datomic and MongoDB databases). A responsive/agile software process was desirable
Organizational	**Size:** Given that the organization comprised (on a full time basis) of between one and two highly specialized, post-Doctoral and close-working engineers, the need for documentation as a means for internal communication was very low
Business	**Business Drivers:** Being a small start-up organization, the pressure to manage finances and minimize costs was high. As a result, the use of technology solutions for quality (e.g. TDD and CI) was preferred to human solutions (which also serviced the demand to quickly deliver high quality software on a continual basis);
Business	**Payment Arrangements:** In many cases, fixed price contracts were secured with the result that the motivation to adopt a minimal scope delivery was increased;
Business	**Magnitude of Potential Loss:** Since inaccurate queries can result in inaccurate calculations and information, the magnitude of potential loss for low quality software was potentially financially very high. To address this factor, large investments in TDD/CI. Plus, the architectural decision to adopt a dual/hybrid database solution had a major impact in de-risking potential software issues;
Business	**Customer Satisfaction:** Given the profile of clients as large financial services IT provided, the quality of the application had to consistently very high. TDD and CI in the software process contributed to realizing this confidence and quality

5 Discussion

In the case of Optimality, we witness a common theme in software development process decision making: a complex set of dominant and sometimes interrelated situational factors need to be accommodated in the software development process, thus any individual software development process decision may (and perhaps ideally should) deliver positive benefits for multiple situational constraints (ref. to Table 4). This type of approach can be considered favorable in the context of complex adaptive systems, wherein a cocktail of interrelated concerns continually interact and we seek to derive the optimal holistic outcome across all concerns when adapting a process. As we have argued in the past, the relationship between a software development process and its situational context would appear to be an instance of a complex adaptive system [3] and therefore, discovering the type of process thinking that we have revealed in Optimality further supports this observation. However, Optimality did not conduct their process adaptation through application of the situational factors framework utilised in this retrospective study, rather they modified whatever aspect of the process they felt justified change at any point in time (and only to the extent that it was economically defensible). Here we see some evidence that perhaps each SME is unique in their process, fighting for survival with only limited resources they may be highly constrained to make just those changes which can deliver a quick return. But perhaps this uniqueness is not just an SME phenomenon as in a related study into the role of situational context in SMEs and medium sized organisations, we have witnessed some considerable variability in reported process enactment [15].

Evidence in support of the role of learning in buttressing improved process adaption was observed in the Optimality case study. While the company had a tendency for aggressive product innovation with an ultra-lean approach to feature delivery, the experience gained with this approach demonstrated that in practice, the cost of refactoring (which was an absolute necessity given the product quality and cost-base constraints) mounted over time. Accordingly, the company had to adapt their process so as to improve their ability to implement better product architecture and design early in development cycles so as to strike an improved economic balance. This is interesting as it represents a regression from lean/agile thinking back to more traditional approaches.

With the variability in iteration durations quite high and the need to reduce risk in relation to the impact of low quality, it is perhaps the case that the Optimality process, while being agile, is also resonant with Boehm's spiral model [16]. We also see evidence in Optimality that would appear to confirm a trend towards increased automation in software development which we have witnessed in other case studies [17, 18]. And while Optimality may notional consider their process to be agile/lean, the significant variation in iteration durations (from a total of 5 days to >130 days) and the heavy burden that can be placed on a small team in a start-up environment, may run contrary to the agile principle: "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely" [19]. Clearly, a constant pace of development is difficult to establish where there is not a constant pace of requirements identification, and while a sustainable pace is a worthy goal, there remain segments of the software development community who continue endure long and unpredictable working hours. Perhaps when economics and human nature collide with worthy ideals, there will always be a battle to be waged.

Acknowledgments. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 to Lero, the Irish Software Engineering Research Centre (www.lero.ie) and Enterprise Ireland grant CF20133605.

References

1. Clarke, P., O'Connor, R., Leavy, B., Yilmaz, M.: Exploring the relationship between software process adaptive capability and organisational performance. IEEE Trans. Softw. Eng. **41**(12), 1169–1183 (2015)
2. O'Connor, R.V., Clarke, P.: Software process reflexivity and business performance: initial results from an empirical study. In: Proceedings of the 2015 International Conference on Software and System Process, pp. 142–146 (2015)
3. Clarke, P., O'Connor, R.V., Leavy, B.: A complexity theory viewpoint on the software development process and situational context. In: Proceedings of the 2016 International Conference on Software and System Process (ICSSP 2016) 2016
4. Clarke, P., O'Connor, R.V.: Changing situational contexts present a constant challenge to software developers. In: O'Connor, R., Umay Akkaya, M., Kemaneci, K., Yilmaz, M., Poth, A., Messnarz, R. (eds.) Systems, Software and Services Process Improvement. CCIS, vol. 543, pp. 100–111. Springer, Cham (2015). doi:10.1007/978-3-319-24647-5_9
5. O'Connor, R.V., Elger, P., Clarke, P.: Exploring the impact of situational context: a case study of a software development process for a microservices architecture. In: Proceedings of the International Conference on Software and Systems Process (ICSSP 2016), pp. 6–10 (2016)
6. Nevalainen, R., Clarke, P., McCaffery, F., O'Connor, R.V., Varkoi, T.: Situational factors in safety critical software development. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2016. CCIS, vol. 633, pp. 132–147. Springer, Cham (2016). doi:10.1007/978-3-319-44817-6_11
7. Feiler, P., Humphrey, W.: Software process development and enactment: concepts and definitions. CMU/SEI-92-TR-004. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA (1992)
8. SEI, CMMI for Development, Version 1.3. CMU/SEI-2006-TR-008. Software Engineering Institute, Pittsburgh, PA, USA (2010)
9. Clarke, P., O'Connor, R.V., Yilmaz, M.: A hierarchy of SPI activities for software SMEs: results from ISO/IEC 12207-based SPI assessments. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 62–74. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30439-2_6
10. McLeod, L., MacDonell, S.: Factors that affect software systems development project outcomes: a survey of research. ACM Comput. Surv. **43**(4), 24:1–24:56 (2011)
11. Orlikowski, W.J., Baroudi, J.J.: Studying information technology in organizations: research approaches and assumptions. Inf. Syst. Res. **2**(1), 1–28 (1991)
12. Dyba, T.: Contextualizing empirical evidence. IEEE Softw. **30**(1), 81–83 (2013)
13. Dyba, T., Sjoberg, D.I.K., Cruzes, D.S.: What works for whom, where, when, and why? on the role of context in empirical software engineering. In: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 19–28 (2012)

14. Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: towards a comprehensive reference framework. J. Inf. Softw. Technol. **54**(5), 433–447 (2012)
15. Clarke, P., O'Connor, R.V., Solan, D., Elger, P., Yilmaz, M., Ennis, A., Gerrity, M., McGrath, S., Treanor, R.: Exploring software process variation arising from differences in situational context. In: Proceedings of the 24th European and Asian Conference on Systems, Software and Services Process Improvement (EuroSPI 2017), 5–8 September 2017, Ostrava, Czech Republic (2017). (Accepted for publication)
16. Boehm, B.: A spiral model of software development and enhancement. IEEE Comput. **21**(5), 61–72 (1988)
17. Clarke, P., Elger, P., O'Connor, R.V.: Technology-enabled continuous software development. In: Proceedings of the International Conference on Software Engineering (ICSE) Workshop on Continuous Software Evolution and Delivery (CSED) 2016
18. O'Connor, R.V., Elger, P., Clarke, P.: Exploring the impact of situational context: a case study of a software development process for a microservices architecture. In: proceedings of the International Conference on Software and Systems Process (ICSSP), Co-Located with the International Conference on Software Engineering (ICSE), pp. 6–10 (2016). doi:10.1145/2904354.2904368
19. Fowler, M., Highsmith, J.: The agile manifesto. Softw. Dev. **9**, 28–32 (2001)

Aspects You Should Consider in Your Action Plan When Implementing an Improvement Strategy

Peter H. Carstensen[1] and Otto Vinter[2(✉)]

[1] Department of Computer Science,
University of Copenhagen, 2200 København N, Denmark
pehe@di.ku.dk

[2] Department of People and Technology,
Roskilde University, 4000 Roskilde, Denmark
otv@ruc.dk, vinter@ottovinter.dk

Abstract. Both ISO/IEC 15504 (SPICE) and ISO/IEC 33014 include a step in their improvement process called: Develop action plan. But which actions should you include, and are you sure that these actions cover all aspects? We have performed a thorough study of the change strategy literature that is the foundation for the ten overall change strategies defined in ISO/IEC 33014. We have extracted statements from this material that represent generic actions recommended by the authors for each strategy. Through analytic induction we have then identified and validated eight aspects that you should consider when choosing your concrete actions for executing the strategy.

Keywords: Organizational change · Change strategy · Action plan · Process improvement · ISO/IEC 33014 · ISO/IEC 15504 · SPICE

1 Introduction

Process improvement is a challenge irrespective of whether you seek formal assessment of your general maturity through one of the prominent frameworks like CMMI [1], ISO/IEC 15504 (SPICE) [2], or merely want to improve those processes you feel are in most urgent need. Process improvement is, however, only one type of organizational change. They are all about changing people's way of working and consequently inherently complex, difficult and often prone to failure.

You must therefore carefully select your change strategy before you develop a detailed action plan. These can never be standardized, because they are heavily dependent on your context and the nature of change you want to achieve.

So, where can you look for guidance when you need to decide on your change strategy? ISO/IEC 33014 [3] defines ten distinctly different overall change strategies for you to choose from. The ten change strategies are listed in Table 1. Each strategy is based on a comprehensive survey of management literature, the most encompassing of these are by Mintzberg et al. [4, 5], Senge [6] and Huy [7].

The ten strategies were originally identified during the Danish research project Talent@IT (2003-2006) and the results were presented at conferences (e.g. [24]), in

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 467–480, 2017.
DOI: 10.1007/978-3-319-67383-7_34

Table 1. The ten overall change strategies defined in ISO/IEC 33014 (re. Table 4 in [3]). The literature examples are those where we found most recommended actions related to the change strategy.

Change strategy	Definition and literature examples
Attitude-driven (Learning-driven)	Change is driven by a focus on organizational learning, individual learning and what creates new attitudes and behavior
	Lit.ex: Senge [6], Huy [7]
Commanding	Change is driven and dictated by (top) management. Management takes on the roles as owner, sponsor and change agents
	Lit.ex: Huy [7], Mintzberg [4, 5]
Employee-driven	Change is driven from the bottom of the organizational hierarchy when needs for change arise among employees
	Lit.ex: Kensing [8, 9], Andersen et al. [10]
Exploration	Change is driven by the need for flexibility, agility, or a need to explore new markets, technology or customer groups
	Lit.ex: Benner and Tushman [11], Senge [6]
Measures-driven	Change is driven by measures and measurements
	Lit.ex: Oakland [12]
Optionality	Change is driven by the motivation and need of the individual or group. It is to a large degree optional whether the individual takes the innovation into use
	Lit.ex: Rogers [13]
Production-organized	Change is driven by the need for optimization and/or cost reduction.
	Lit.ex: Benner and Tushman [11], Liker [14], Huy [7]
Reengineering	Change is driven by fundamentally rethinking and redesigning the organization to achieve dramatic improvements
	Lit.ex: Bashein et al. [15], Davenport [16], Hammer and Champy [17], Huy [7]
Socializing	Change in organizational capabilities is driven by working through social relationships. Diffusion of innovations happens through personal contacts rather than through plans and dictates
	Lit.ex: Huy [7], Snowden and Boone [18], Gittel et al. [19, 20]
Specialist-driven	Change is driven by specialists, either with professional, technical, or domain knowledge
	Lit.ex: Ciborra [21], Simon [22, 23]

journals (e.g. [25]) and in a book [26]. A questionnaire-based tool was also developed to identify which (combination) of the strategies would be most suitable in a specific context ([26] p. 173).

However, once the change strategy (combination) has been selected (see Fig. 1, ISO/IEC 33014 clause 6 step 4), little help has previously been available to assist management or process consultants in developing the detailed action plan as required in clause 5 step 4 of ISO/IEC 33014 (see Fig. 1), or the identical step 4 in Fig. 4 of ISO/IEC 15504-4 (SPICE) [2].

To alleviate this, we have performed a thorough study of the change management strategy literature that is the foundation for the ten overall change strategies defined in ISO/IEC 33014. We have extracted statements from this material that represent recommended actions by the authors for each strategy. We have primarily focused on the (top) management perspective.

During this process we have identified and validated eight aspects that can be seen as overall headings or categories for the actions. We believe that these eight aspects can be supportive for the management or process consultant and change agent when developing and executing a specific action plan for a change in a particular context. The eight aspects are listed in Table 2.

In this paper we first present the process of extracting relevant recommended actions from the change management literature and how we arrived at identifying the eight aspects. Then we discuss the usability of these aspects and to what extent they cover the ten change strategies in ISO/IEC 33014.

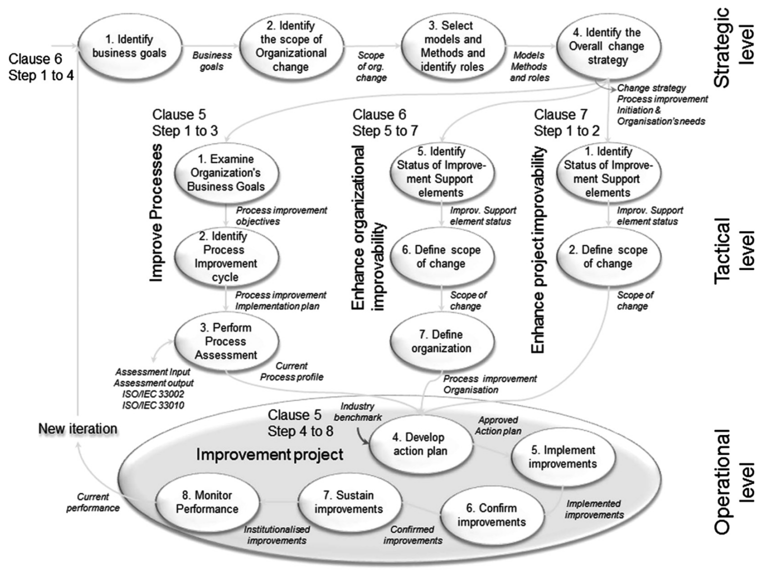

Fig. 1. Overview of the ISO/IEC 33014 model (Fig. 1 in [3]). This paper focuses on clause 6 step 4 (Identify Overall change strategy) and clause 5 step 4 (Develop action plan).

2 What Have Others Done

There are huge amounts of literature on change strategies and general recommendations on how to organize and conduct strategic changes. Among the widely known and broadly recognized can be mentioned: Kotter focusing on an eight point plan for a change [27]; Hammer and Champy arguing for reengineering the whole corporation [17]; Mintzberg's overall conceptual frameworks for understanding and changing different organization structures [4]; Rogers' deep analyses of adoption and diffusion of changes [13]; Senge's approach to change organizations through socializing and learning [6]; and Huy focusing on structuring and sequencing of strategies over time [7]. Most of these have been central in setting the scene for the research reported in this paper. Finally, the ISO/IEC standards mentioned above [2, 3] have also contributed to defining the steps in planning and executing a change.

In his influential book on strategy safaris [5] Mintzberg and his co-authors suggest a set of 'strategy schools' approaching the strategy formation process as: a conception, a negotiation, a transformation, or as being a formal, analytical, visionary, mental, emergent, collective or reactive process. These ten distinct schools might be supportive for characterizing a specific change process or categorizing an approach or methodology.

Balogun and her co-authors [30, 31] provide a diagnostic framework called the 'change kaleidoscope' for identifying appropriate 'design choices'. This framework consists of three layers: The organizational strategic change context referring to the broader strategic analysis conducted; eight essential features of the change context (time, scope, preservation, diversity, capability, capacity, readiness, and power); and six dimensions of choices open to the change agent (path, start-point, style, target, levers and roles).

Next to the academic literature lots of consultants, bloggers, consultancy companies etc. have provided their lists of suggestions and recommendations. An example could be strategy+business that provides ten principles for leading change management [28] focusing on culture, how to involve all layers in the organization, how to engage and lead etc.; or Forbes [29] that suggest ten recommendations that are derived from science.

However, most of the literature presents specific strategies and recommendations on how to plan and perform a change. It focuses on the processes to be employed in the planning and execution of the change. A multitude of three to seven stage models are promoted by many authors as presented in the overview paper by Al-Haddad and Kotnour ([32] p. 246).

Throughout the literature are mentioned aspects like: motivating people and sharing the vision. But the main focus in the literature remains on how to structure and plan the change and the processes to be followed. The aspects mentioned are not turned into a framework of common aspects to be considered by managers, process consultants and change agents in their planning and design of the actions to be performed. The common aspects we have identified are: Communication, Culture, Competences and training, Decision-making, Knowledge acquisition, Methods and techniques, Organizing, Processes and plans (see Table 2).

3 Our Research Approach

Our research approach is based on analytic induction. Znaniecki in his book [33] outlines this approach in a number of steps:

> "Begin by studying a small number of cases of the phenomenon to be explained, searching for similarities that could point to common factors. Once a hypothetical explanation has been developed further cases are examined. If any one of these does not fit the hypothesis, either the

hypothesis is reformulated so as to match the features of all the cases so far studied, or the original definition of the type of phenomenon to be explained is redefined, on the grounds that it does not represent a causally homogeneous category. Further cases are then studied until no more anomalies seem to be emerging." (quoted from Wikipedia)

We began by extracting recommended actions from the literature behind two of the ten ISO/IEC 33014 strategies (Optionality and Specialist-driven, see Table 1). We read the texts from end-to-end and extracted statements by the author(s) that seemed characteristic for the strategy. One of us was reading the texts and the other was reviewing the extracted recommendations for action.

We kept the recommendations as close to the original statements in the texts as possible. Our main intervention was to make the statements actionable i.e. put them into imperative form. An example of such a reformulation of a statement is: "Leadership of change belongs to one small group of people, typically located at the top of the formal hierarchy" ([7] p. 605), which was reformulated to a recommended action: "Ensure that leadership of the change belongs to one small group of people typically located at the top of the formal hierarchy".

When we looked at the resulting recommended actions we could see that some of the recommendations were addressing the same aspect of the change (e.g. culture). We identified eight such aspects. Most of the recommended actions could be allocated to one aspect only. However, in some cases a recommended action was allocated to two or at most three aspects. This was no surprise to us as we had deliberately kept the original formulation of the statements in the texts, which therefore could include several aspects.

We then hypothesized that these eight aspects would be applicable also to the recommendations for the remaining eight strategies. We discussed each aspect and defined them properly. We quickly realized that the aspects were not completely orthogonal, which was never our intension. We regard the aspects as a kind of check-list for management or process consultant, who develops and executes a change process. Their applicability in practice is more important to us than whether they are 100% precise. The eight aspects are listed in Table 2.

To validate the hypothesis we repeated the same process for two more of the ten strategies (Production-organized and Socializing, see Table 1). The extracted statements from the corresponding literature were now analyzed whether they could be allocated to the previously identified aspects. We found this rather easy, which seemed to confirm our hypothesis.

However, to make sure that this was not the result of our bias we presented our results to the authors of [25], who had also been involved in the Talent@IT project and one of them our partner when the ten strategies were originally identified. They proposed three more aspects based on their knowledge from improvement projects. We analyzed their proposals and found that two were already covered by our aspects, and the third required an addition to the description of one of our aspects. The review thus resulted in a consolidation of the eight aspects.

With these changes in place we went on to repeat our process for the remaining six strategies. During this process we did not find any need to update our definition of the aspects, and the recommended actions could again be allocated to no more than three.

Table 2. The eight aspects of recommended actions for executing a change, with examples.

Aspect	Short explanation and examples
Communication	Types of information that should be communicated, to/by whom, when/through which channels - "Communicate often and focus on the opportunities" ([15] p. 6) - "Develop and publish clear documented corporate beliefs and purpose - a mission statement" ([12] p. 36)
Culture	The culture that should be established/changed/supported in the organization to secure the success - "Encourage people to challenge and examine alternatives" ([7] p. 619) - "Encourage individuals to establish improvement goals for themselves and their groups" ([12] p. 19)
Competences and training	The competences management and change-team should have before the work on the change is initiated, or which should be built up on the way - "Ensure that the local adopters are sufficiently educated/trained to understand the big picture about the available innovation/change" ([13] p. 399) - "Develop leadership skills among middle managers to install new values in them" ([7] p. 620)
Decision-making	Decisions and commitments it is important to make and achieve before and during the work on the change, who should make them, when and within what scope - "Shift power to the experts (specialists) by virtue of their knowledge" ([4] p. 106) - "Base your management decisions on a long term philosophy, even at the expense of short term financial goals" ([14] p. 5)
Knowledge acquisition	The knowledge that should be obtained/gained/collected before and during the work on the change - "Ensure that the impact of existing initiatives are evaluated and considered carefully before starting another series of improvement initiatives" ([12] p. 159) - "Analyze constraints and evaluate specifications" ([22] p. 189)
Methods and techniques	Concrete methods and techniques that should be used in order to execute the change - "Co-ordinate, advise and manage instead of keeping control centrally" ([21] p. 39) - "Forget the past, break rules and traditions in order to create variation" ([11] p. 251)
Organizing	How the change-team as well as all involved or affected by it should be organized/structured - "Establish and cultivate communities of practice" ([21] p. 22) - "Form quality improvement teams with representatives from each department" ([12] p. 19)
Processes and plans	Concrete plans and processes that management and change-team establish for the work - "Overhaul processes which create a vicious cycle of overload, stress, burnout and low morale" ([7] p. 612) - "Ensure control at all stages of planning and operationalization of the strategy" ([5] p. 55)

As a final review we presented the recommended actions and the aspects to a team of master students at Roskilde University specializing in process improvement, and they found them applicable to their own company contexts.

Through our application of analytic induction as described in the beginning of this section, we believe that we have validated our hypothesis and that the eight aspects we have identified have relevance and applicability to practitioners in the change management field. But of course further use in practical contexts must be performed to prove this.

4 Results

As mentioned in the previous section we went carefully through a number of books (17) and papers/articles (15) aiming at identifying recommendations for (top) management, process and change consultants on how to organize and carry out strategic changes.

The first rough list contained a total of 665 recommendations distributed over the ten different change strategies mentioned in ISO/IEC 33014 [3]. For most of the strategies we identified between 50 and 104 recommendations. However, the strategies of Commanding and Optionality were considerably lower with respectively 33 and 28 recommendations each. We also identified a number of recommendations with such general intent that they could be relevant for almost all strategies. We will not discuss these further here, except report that they also can be allocated to our eight aspects.

We reviewed all 665 recommendations and selected those recommendations that were most clearly indicative of each the ten strategies. This selection resulted in a total of 233 recommendations for the ten strategies distributed as listed in Table 3.

Our analytic induction then identified eight generic aspects that managers, process consultants and change agents need to consider when planning and executing changes. We consider these eight aspects to be useful when developing a context specific change plan. The eight aspects are listed in Table 2.

As mentioned in the previous chapter we evaluated, verified and validated the relevance and applicability of these eight aspects. One of the ways of testing our hypothesis was to map the distribution of recommendations on the eight aspects and ten strategies. This resulted in the picture shown in Fig. 2. As it can be seen the coverage of aspects for each strategy differs a lot. We will reflect on this in a following Sect. 5.1.

It should be noted that the total number of Xs in the cells is greater than the 233 recommendations we have selected. The reason is that some recommendations could be allocated to more than one aspect. For example the recommendation for the Specialist-driven strategy: "Design organizational units where, to the greatest degree possible, local decision makers confront the full range of issues and dilemmas" ([6] p. 287) can be allocated to both the Organizing and Decision-making aspect. In total we had 133 recommendations with a singular aspect-categorization, 91 with tuple, and only 9 with a triple categorization distributed on the strategies as shown in Table 3.

	Communication	Culture	Competences and Training	Decision-making	Knowledge acquisition	Methods and Techniques	Organizing	Processes and plans
Attitude-driven	XXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXXXX	XX	XX	XXXXXXXXXXXXX	XXX	X
Commanding	XXXXXXX			XXXX		XXXXXXX	XXX	XXXX
Employee-driven	X	XXXXXXX	XXXXX	XXXXXX	X	XXXXXXX	XXXXX	
Exploration	X	XXXX	XXX	XXXX	X	XXXXXXXXXXXX	XXXXXXXXXXX	XXX
Measures-driven	XX	XXXX	XXX	XX	XXX	XXXXXXX	XXXXX	XXXXXX
Optionality	XXX	XXXX	XX	XXX		XXXXXX	XXXXX	X
Production-organized	X	XXX	XXX	XXXXXX	XX	XXXXXXXXXX	XXXXX	XXXXXXX
Reengineering	XXXXXXX		XX	XXXXXX	XX	XXXXXXX	XXX	XXXXXXX
Socializing	XX	XXXXXXX	XXX	XXX	XX	XXXXXXXXXXXXXXX	XXXXXX	X
Specialist-driven	XXX	XX		XX	XXXXX	XXXXXXXXX	XXXX	XX

Fig. 2. Mapping the selected recommendations onto aspects and strategies. Strategies are listed by rows, aspects by columns and number of recommendation as cell contents (Xs).

Table 3. Overview of the number of selected recommendations per strategy, and whether they were allocated to one or more aspects.

Change strategy	Total	Single aspect allocation	Tuple aspect allocation	Triple aspect allocation
Attitude-driven	35	20	12	3
Commanding	19	13	6	0
Employee-driven	22	12	10	0
Exploration	25	11	14	0
Measures-driven	21	13	5	3
Optionality	16	8	8	0
Production-organized	27	18	8	1
Reengineering	21	9	11	1
Socializing	25	12	12	1
Specialist-driven	22	17	5	0

5 Discussion

As mentioned previously the eight aspects we have identified are to some extent recognizable in many other authors' recommendations and reflections on 'general' issues to consider when planning and executing change activities. Most influential authors argue for establishing a vision for the change, motivating the involved actors, and they provide specific recommendations for how the change work should be organized and which specific methods and approaches to apply. In this light we could say that they indirectly address all our eight aspects in various amount of detail.

The aspects: Organizing, Communication, Methods and techniques, Processes and plans are addressed by almost all authors writing about change and change strategies. The aspects: Decision-making, Culture, and Competences and training are also reflected upon by many authors. However, we have found fewer authors explicitly recommending actions related to the Knowledge acquisition aspect, both with respect to information collection before the change process is initiated and during the execution of the change (i.e. feedback).

Even though all eight aspects can be found in various wrappings and detail very few of the classic publications on change and change strategies aim at providing a comprehensive set of all aspects that could be relevant to consider when developing recommendations for concrete context-specific action.

Since our aspects were derived from a study of this literature looking for recommended actions for an action plan, it is of no surprise that the aspects we have identified are reflected in much of the prominent change management literature.

Therefore, we will not claim that the individual aspects we have identified constitute something new. The interesting result of our work is that these eight aspects together seem to constitute a complete framework of relevant aspects to consider when developing change action plans. And we have found only a few attempts at providing a framework for supporting change strategies with specific recommendations.

Mintzberg and his co-authors [5] suggest a set of 10 ‘strategy schools’ approaching the strategy formation process along different dimensions (schools). These strategy schools provide a framework for overall characterization of the change as: a conception, a negotiation, a transformation, etc. It will, however, be difficult to characterize specific recommendations by means of these schools.

Balogun and her co-authors [30, 31] with their ‘change kaleidoscope’ provide eight features of the change context, e.g.: capability, capacity, power and readiness. Furthermore they suggest a list of dimensions which the change agent can act on, e.g.: style, levers and roles. Again, this framework is more directed towards characterizing a specific change or strategy, but is not suited for characterizing specific recommendations.

Finally, the ISO/IEC standards [2, 3] issue guidelines for how to plan and manage process improvements in organizations (see Fig. 1). They mention all of our eight aspects, but they keep the guidelines at an overall level with very few detailed recommendations for action.

5.1 Reflections on Aspect Coverage of Strategies

An interesting thing to observe and consider has to do with the coverage of aspects in the recommendations we have selected for each of the ten strategies. Figure 2 provides an overview of this coverage and Table 3 presents a little more detail.

The first observation is the fairly uneven distribution of the number of recommended actions for a strategy (Xs in Fig. 2) related to each aspect. The aspect of Methods and techniques is by far the most common across all ten strategies i.e. 93 recommendations and only 18 for Knowledge acquisition. This could be explained by the fact that when authors provide recommendations on how to execute changes, it is quite natural that they suggest specific methods and techniques. Another explanation could be that our definition of the Methods and techniques aspect is too broad. However, we have yet found it hard to split it without creating a rather artificial distinction for this aspect.

If we look into the specific distribution of aspects for each strategy we also see significant differences. E.g. we have found no recommendations for three aspects in the case of the Commanding strategy. And in the case of the Employee-driven strategy another set of three aspects are covered with only 0-1 recommendations.

In a number of Danish organizations the authors of [25] have found that Optionality, Specialist-driven, and Socializing strategies come out as best strategy fit. Furthermore, due to external requirements dictating the change a Commanding strategy was also often a good fit. For illustrating the discussion we’ll look a bit more into these four.

For the Optionality strategy the most frequent recommendations belong to the aspects Methods and techniques (6) and Organizing (5), whereas Knowledge acquisition has not been mentioned at all in any of the 16 recommendations selected for this strategy. An example of a recommendation in Methods and Techniques for the Optionality strategy is: “Allow locally perceived needs and problems decide which innovations/changes they should adopt” ([13] p. 396).

In the Specialist-driven strategy the most frequent recommendations again belong to the aspect Methods and techniques (9), whereas Culture (2), Processes and plans (2) and Competences and training (0) are the least common aspects allocated to the 22 recommendations. An example of a recommendation in Methods and Techniques for the Specialist-driven strategy is: “Co-ordinate, advise and manage instead of keeping control centrally” ([21] p. 39).

In the Socializing strategy the most frequent recommendations again belong to the aspect Methods and techniques (15), Culture (7) and Organizing (6), whereas Processes and plans (1) is the least common aspect allocated to the 25 recommendations. An example of a recommendation in Methods and Techniques for the Socializing strategy is: “Allow people to network and experiment with new social-work relations and power rearrangements” ([7] p. 615).

For the Commanding strategy the most frequent recommendations are allocated to the aspects Communication (7) and Methods and techniques (7), whereas Knowledge acquisition, Culture, and Competences and training are not covered at all in the 21 recommendations. An example of a recommendation on Communication for the Commanding strategy is: “Meet employees face to face and communicate your vision” ([7] p. 612).

If we look at the eight aspects one by one we observe the following (see Fig. 2):

- The Communication aspect is strongly represented in the Commanding and Reengineering strategies. This appears quite natural due to the strong top management influence on the change process. The Communication aspect is almost absent in the Employee-driven, Exploration and Production-organized strategies. The first two can be explained because of the direct involvement of the actors in the target group.
- The Culture aspect is very frequent in Attitude-driven (Learning-driven) and well represented in Employee-driven and Socializing strategies. Again, this seems natural when having the nature of these strategies in mind. The Culture aspect is absent in the Commanding and Reengineering strategies. Considering the nature of these forceful types of changes it appears quite natural that the culture of the organization is considered of less importance.
- The aspect of Competences and training is by far the most prominent aspect in recommendations for the Attitude-driven strategy. This again seems natural as teaching and learning is the focus for this strategy. The original name for this strategy was Learning-driven which also indicates this. The Competences and training aspect is absent in the Commanding and Specialist-driven strategies. Again it seems intuitively explainable. In the Commanding strategy the changes are dictated and must take place, and in the Specialist-driven strategy the relevant competences are already in place (experts).
- Decision-making is present in recommendations for all ten strategies with some variations. But there is no clear absence or strong representation for any of the ten strategies.
- Knowledge acquisition is weakly represented in all strategies, except the Measures-driven and Specialist-driven strategies. It seems quite natural that for these two strategies the acquisition and use of information is essential. For the other eight we can only observe that the authors in the change strategy literature evidently have not addressed this aspect often in their recommendations.

- The Methods and techniques aspect is very frequent in recommendations for all ten strategies. It is clearly an area that the authors of the change strategy literature find highly relevant and natural to provide recommendations for.
- The Organizing aspect is present in recommendations for all ten strategies. It is interesting that the number of recommendations for organizing is very high for the Exploration strategy. The reason for this could be that very often separate organizational units for experimentation and exploration are established.
- The aspect of Processes and plans is most common in recommendations for the Production-organized and Reengineering strategies. This seems to be quite natural having the nature of the strategies in mind. On the other hand Processes and plans are absent or very weakly represented in recommendations for the Optionality, Socializing, Employee-driven and Attitude-driven (Learning-driven) strategies. This could indicate that for strategies having a very active involvement and participation of the actors working in the setting being changed, it is not considered important to give specific recommendations on how to plan, structure and execute the change activities.

During the categorization of the individual recommendations we found many recommendations that covered more than one aspect. In total we had, as mentioned previously, 133 recommendations with a singular aspect-categorization, 91 with tuple, and 9 with a triple categorization. The main explanation for the tuple and triple categorizations is that many authors, quite naturally, provide recommendations that both cover a general statement of what is important (e.g. concerning communication, culture or decision-making) and indicate how it should be planned, organized or conducted.

It is still too early to discuss the usefulness and applicability of our eight aspects for developing concrete context-specific actions for a change. Such discussion requires further experiments where the aspects are used in specific cases when real-life changes are being planned, prepared and carried out. However, we are quite convinced that the eight aspects will function as a good checklist for management, process consultants and change agents when planning and executing changes.

In the literature we have found many indications that the eight aspects cover relevant facets and ingredients in change management. Careful consideration of the eight aspects when designing a specific action plan will therefore constitute a good platform for ensuring that all relevant facets have been addressed. When we start using the aspects on specific real-life cases it is quite possible that we will find a need for detailing and subdividing each of the aspects further in order to provide much more advice at a level where it can be of further assistance to practitioners.

In a research context we expect the eight aspects to provide a conceptual framework for identifying and characterizing strategies, approaches and recommendations for change and change activities. The aspects might potentially be used as a set of dimensions for comparing different strategies and compare different approaches or descriptions of a specific strategy, e.g. to compare the approaches of Senge [6] and Huy [7] to the Attitude-driven strategy. For research purposes further detailing and subdividing of the aspects will probably also be required.

5.2 Methodological Reflections and Future Research

The study presented in this paper is, of course, biased for a number of reasons. Because the work was a follow-up on the Talent@IT project, our starting point for relevant literature was [24, 25]. Therefore a number of other relevant references were not initially considered. During the literature survey process we added literature we found relevant, but whether the starting point still biased our focus and therefore what we have addressed is unclear.

Furthermore, we are the only two persons who have (1) identified the recommendations from the literature, (2) selected the recommendations to be included in the final list for each strategy, and (3) conducted the analytic induction and abstraction leading to the eight aspects.

In our research approach we have continuously aimed at getting our findings and selections validated, but there is still a risk of a methodological weakness and a source for bias in our results. Thus, future research should aim at investigating other relevant sources for recommendations on change strategies; seek to get the aspects tested and validated on other sets of recommendations; and possibly validate whether other readers of the same literature would extract a similar set of recommendations, and select a similar set as the most important.

6 Conclusion

Based on the literature behind the ten strategies defined in ISO/IEC 33014 [3] (Table 1) we have extracted statements representing the authors' recommended actions for change for each strategy in order to develop a catalogue of these to assist the (top) management or process consultant in developing their own context-specific action plan (clause 5 step 4 in Fig. 1).

During this search we discovered that the recommended actions could be grouped according to eight aspects of the change to be undertaken (Table 2). We verified this hypothesis across all the recommendations we extracted, and validated the applicability of the actions and aspects by exhibiting a number of master students to them with a confirmative result.

We also verified that the distribution of the selected actions across strategies and aspects (Fig. 2) correspond to what you would expect given the nature of the literature examined (see Sect. 5.1).

We realize that we have not included all relevant literature on change management and that the real validation of the usefulness of the aspects can only take place on real-life cases. However, we only see this as a possibility for further testing of the validity of the aspects.

Given the reservations above, we are confident of the usefulness of the aspects we have found. We firmly believe that when planning and executing changes in your specific context/case, you should carefully consider these eight aspects (Table 2) when developing your action plan and accompanying list of recommended actions.

Acknowledgments. We wish to thank Professor Jan Pries-Heje (Roskilde University) and Jørn Johansen (Whitebox) for thoughtful discussions and review of the eight aspects. We also wish to thank the master students at Roskilde University for their contribution in the validation process.

References

1. Chrissis, M.B., Conrad, M., Shrum, S.: CMMI for Development: Guidelines for Process Integration and Product Improvement, 3rd edn. CMU/Software Engineering Institute, Westford (2012). ISBN-13: 978-0321711502
2. ISO: ISO/IEC/TR 15504-4. Information technology–Process assessment–Guidance on use for process improvement and process capability determination, Geneva, Switzerland (2013)
3. ISO: ISO/IEC/TR 33014. Information technology–Process assessment–Guide for process improvement, Geneva, Switzerland (2013)
4. Mintzberg, H.: Structure in Fives - Designing Effective Organizations. Prentice-Hall, Englewood Cliffs (1983)
5. Mintzberg, H., Ahlstrand, B., Lampel, J.: Strategy Safari: A Guided Tour Through the Wilds of Strategic Management. Financial Times. Prentice Hall, London (2002)
6. Senge, P.M.: The Fifth Discipline: The Art and Practice of the Learning Organization. Doubleday, New York (1990)
7. Huy, Q.N.: Time, temporal capability, and planned change. Acad. Manag. Rev. **26**(4), 601–623 (2001)
8. Kensing, F.: Methods and Practices in Participatory Design. ITU Press, Copenhagen (2003)
9. Kensing, F., Blomberg, J.: Participatory design: issues and concerns. Comput. Supported Coop. Work **7**(3–4), 167–185 (1998)
10. Andersen, C.V., Krath, F., Krukow, L., Mathiasssen L., Pries-Heje, J.: The grass root effort. In: Mathiassen, et al. (eds.) Improving Software Organizations - From Principles to Practice. Addison-Wesley (2001)
11. Benner, M., Tushman, M.: Exploitation, exploration, and process management: the productivity dilemma revisited. Acad. Manag. Rev. **28**(2), 238–256 (2003)
12. Oakland, J.S.: TQM – Text with Cases, 3rd edn. Butterworth-Heinemann, Oxford (2003)
13. Rogers, E.M.: Diffusion of Innovations, 5th edn. Free Press, New York (2003)
14. Liker, J.K.: The Toyota Way. Tata McGraw-Hill, New York (2004)
15. Bashein, B.J., Markus, M.L., Riley, P.: Preconditions for BPR success: and how to prevent failures. Inf. Syst. Manag. **11**(2), 7–13 (1994)
16. Davenport, T.H.: Process Innovation: Re-engineering Work through Information Technology. Harvard Business School Press, Boston (1993)
17. Hammer, M., Champy, J.: Reengineering the Corporation. A Manifesto For Business Revolution. Harper Business, New York (1993)
18. Snowden, D.J., Boone, M.E.: A Leader's Framework for Decision Making. Harv. Bus. Rev. **85**, 68–76 (2007)
19. Gittell, J.H., Seidner, R., Wimbush, J.: A Social Capital Model of High Performance Work Systems. Industry Studies Association, Working Papers Series (2007)
20. Gittel, J.H.: new directions for relational coordination theory. In: Spreitzer, G.M., Cameron, K.S. (eds.) The Oxford Handbook of Positive Organizational Scholarship, Chap. 30 (2011)
21. Ciborra, C.U.: From Control to Drift. The Dynamics of Corporate Information Infrastructures. Oxford University Press, Oxford (2000)
22. Simon, H.A.: The structure of ill structured problems. Artif. Intell. **4**, 181–201 (1973)
23. Simon, H.A.: Search and reasoning in problem solving. Artif. Intell. **21**, 7–29 (1983)

24. Pries-Heje, J., Vinter, O.: A framework for selecting change strategies in IT organizations. In: Münch, J., Vierimaa, M. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 408–414. Springer, Heidelberg (2006). doi:10.1007/11767718_36
25. Pries-Heje, J., Johansen, J.: Choosing change strategy for ISO/IEC 33014. J. Softw. Evol. Process. **27**, 573–583 (2015). doi:10.1002/smr.1724. Published online in Wiley Online Library (wileyonlinelibrary.com)
26. Pries-Heje, J., Johansen, J. (eds.): ImprovAbility: Success with Process Improvement. DELTA, Hørsholm (2013)
27. Kotter, J.P.: Leading Change. Harward Business Review Press, Boston (1996)
28. www.strategy-business.com. 12 May 2017
29. www.forbes.com. 12 May 2017
30. Balogun, J., Hailey, V.H., Gustafsson, S.: Exploring Strategic Change. Pearson, Harlow (2015)
31. Balogun, J., Hailey, V.H.: Devising context sensitive approaches to change: the example of glaxo wellcome. Long Range Plann. J. **35**, 153–178 (2002)
32. Al-Haddad, S., Kotnour, T.: Integrating the organizational change literature: a model for successful change. J. Organ. Change Manag. **28**(2), 234–262 (2015)
33. Znaniecki, F.: The Method of Sociology. Farrar & Rinehart, New York (1934)

Exploring Knowledge Loss in Open Source Software (OSS) Projects

Mehvish Rashid[1,2(✉)], Paul M. Clarke[1,2], and Rory V. O'Connor[1,2]

[1] Dublin City University, Dublin, Ireland
Mehvish.Rashid2@mail.dcu.ie,
{Paul.M.Clarke,Rory.OConnor}@dcu.ie
[2] Lero, the Irish Software Research Centre, Limerick, Ireland

Abstract. Open Source Software is a term used to identify software developed and released under an "open source" license, meaning that under certain conditions; it is openly available for use, inspection, modification, and for redistribution free of cost (or with cost based on the license agreement). Incorporation of OSS while developing software can reduce time and cost of development. The nature of the work force (volunteers and paid) in OSS projects is transient and results in high turnover leading to knowledge loss. In this work, we explore the phenomenon of knowledge loss in OSS projects. Maintenance of OSS projects requires knowledge, typically shared asynchronously using technology-mediated channels. Knowledge sought in this manner is reactive in the sense that a developer will consult these channels looking for possible solutions or supporting information. We follow the backward snowballing to study the relevant literature on knowledge loss in OSS. Our work suggests that proactive knowledge exchange mechanisms may bring some benefits to OSS projects. Further integration of knowledge management practices with the established OSS practices can minimise knowledge loss.

Keywords: Software Development Process · Open Source Software · Knowledge Loss · Knowledge Retention Process · Open Source Software Contributor

1 Introduction

Open Source Software (OSS) is a term used to identify software developed and released under an "open source" license that complies with Open Source Definition (OSD). The OSD uses either a short definition based on four criteria as in the Free Software Foundation (FSF) or a longer version based on ten criteria as in the Open Source Initiative (OSI). The difference between these two definitions is only of language while underlying meaning and outcome is the same. "The freedom to use, change, sell, or give away the software, the availability of source code, and the protection of authors' intellectual property rights are the central tenets of the OSD" [1]. Users with technical inclination can use, freely access the code, inspect, modify and redistribute the software [2]. However, the freedom to use source code from an OSS project and its distribution varies based on which category of OSS license agreement is applied. There are three

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 481–495, 2017.
DOI: 10.1007/978-3-319-67383-7_35

main categories of OSS licenses based on their degree of restrictiveness: Strong-copyleft, weak-copyleft and non-copyleft [3]. A strong-copyleft or restrictive license requires that any derivatives of the original software are also licensed in a similar manner. Weak-copyleft licenses allow the derivatives of the software to be released under different license. Non-copyleft licenses allow the software including derivatives to be redistributed under a different license than the original one. While free software mostly identifies with GNU Public License (GPL), OSS license agreements may vary based on the incorporation of the software that can be either propriety or free. Another term to represent free software is Free Open Source Software (FOSS). The term "free" in FOSS was not considered by some to adequately express the notion of freedom and consequently, in 2001, the European Commission (EC) introduced the term Free/Libre Open Source Software (FLOSS), to avoid taking sides in the debate and to stay neutral on the distinction between free software and open-source software. OSS projects are of varying sizes and at times involve commercial firms who heavily depend on OSS system [4]. A survey conducted in 2015 reported that almost 78% of companies run operations on open source software and 66% of companies have incorporated open source software to create products for customers [5].

In OSS projects, maintenance and development are not considered as two separate phases of software development lifecycle [6]. Software maintenance is the field which is concerned with the evolution of a software system after its initial release [6]. In Closed Source Software (CSS) or traditional software development, the maintenance phase starts after the software is complete, authorised and running [7]. Whereas in OSS projects, the source code may be released before it is complete or in workable form; and the maintenance activities in OSS start when system is still in the initial stage of development [6]. The OSS system, already in phase-out stage, experience a rebirth when other contributors start contributing with new enhancements [7]. Maintenance in an OSS system is the source of continuous evolution and requires knowledge in various forms. In order to solve a problem, a software developer has to understand existing source codes, design a solution, program the solution, and test it. The nature of the OSS work force (volunteers and paid) is transient and results in high turnover on projects. This turnover leads to knowledge loss on OSS projects [8]. As a potential solution to the knowledge loss situation in OSS projects we introduce the concept of Knowledge Management (KM).

1.1 Knowledge Management

It is important to clarify the distinction between the terms *data*, *information*, and *knowledge*. *Data* represents observations and facts without any contextual meaning, whereas *information* is the result of associating data with a meaningful context [9]. In order to convert data into information, it must be contextualized, categorised, calculated and condensed [9]. *Knowledge* is driven from information [9], it is the product of an individual's experience and accumulates as a result of communication or inference [10]. In a general sense, knowledge may be categorised as either *explicit* or *tacit* (or implicit) [11]. Tacit knowledge comprises of skills learned due to the personal capabilities of contributors and if not documented, remains confined to an individual, whereas explicit knowledge is available in documented form [11]. At an organisational level, knowledge is created as a result of the interaction between the tacit and explicit knowledge [11]. Accumulated tacit knowledge is lost when contributors leave projects. Knowledge loss is a problem constantly faced due to employees leaving an organisation [12–14] and it is reported in OSS projects [15–17], where the majority of contributors are typically volunteers. The duration of volunteer participation in OSS projects is considered to be unpredictable [18], with the phenomenon of volunteers joining and leaving at their own discretion being more common in OSS projects when compared with employee-based arrangements that are typical in CSS [18].

In order to reduce the impact of knowledge loss on the organisation's productivity and on product's quality, organisations invest in KM processes. KM is defined as the approach adopted by an organisation to engage workers in relevant activities of creating, managing, sharing and reusing knowledge [19].

The purpose of this work is to explore the problem of knowledge loss in OSS due to the transient profile of contributors and to examine the affect this may have on productivity and quality of the project. In Sect. 2, we will explore the literature related to OSS knowledge loss and further inspect KM activities in OSS projects. In Sect. 3, we interpret the findings from section. We conclude this work in Sect. 4 by proposing directions for future research.

2 Related Literature

In this section, we explore the existing literature relevant to the problem of knowledge loss. In order to identify the literature, the initial step was to find the key set of papers related to the topic. Different search strings were used on Google Scholar such as "knowledge loss in open source software", "knowledge loss in free libre open source software", and "knowledge loss in free open source software".

Key papers of interest were identified using this approach and these formed the initial review corpus. While evaluating the core area of interest, backward snowballing was employed on the initial set of papers. Backward snowballing refers to the process of extending the literature review on the basis of following the trail offered up by the initial set of paper identified [20]. Accordingly, further works of interest were selected from the reference list of initially selected papers. In combination with the term "open source software", other terms relating to knowledge management were also searched on Google Scholar consisting of "knowledge sharing", "knowledge creation", "knowledge reuse", "knowledge retention", "knowledge acquisition", and "knowledge capture". The themes consolidated from the collected papers are depicted in Fig. 1. Each rectangle represents the theme found in the papers and line represents a connection between two themes.

Section 2.1 explains the problem of knowledge loss in OSS, followed by Sect. 2.2 that describes the contributor profile as observed in OSS projects and its implication for knowledge loss. Finally, Sect. 2.3 elaborates on KM related activities in OSS project.

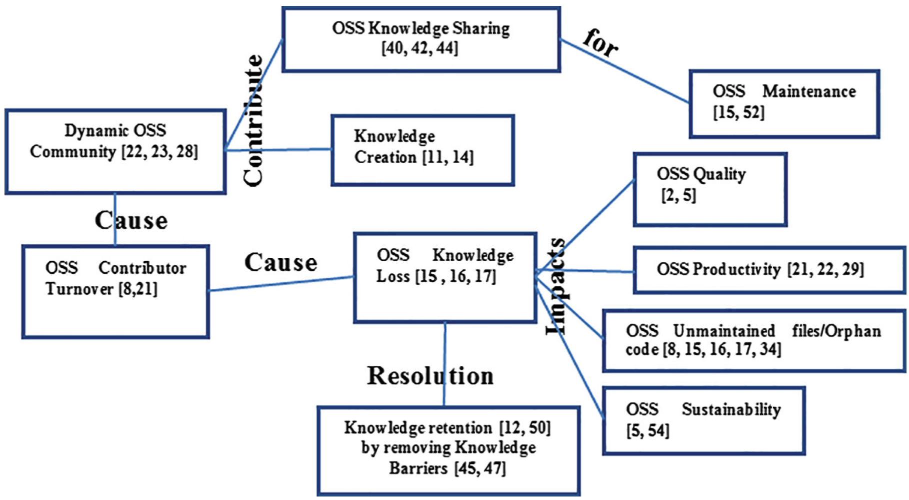

Fig. 1. Mind map of related literature on OSS knowledge loss

2.1 Knowledge Loss in Open Source Software Projects

Evolution of OSS projects result in evolving teams of contributors who are constantly joining, leaving, or changing their role in the project. The phenomenon of resources joining and leaving in this fashion is referred as ‘turnover’ [21]. The contributor turnover leads to knowledge loss in OSS projects. In many large OSS projects, a high turnover has been observed leading to the formation of the new development teams [8]. Knowledge loss impacts the productivity of the OSS projects in two ways: (1) The effort required to acquire knowledge to perform the maintenance tasks; and (2) The loss of effort when code is orphaned and removed from the project.

In order to write quality software code knowledgeable contributors are required. Searching knowledge is argued to be time consuming and costly [22]. The search efforts can vary depending on the source and the level of details. A post or a query on the project mailing list require less efforts while searching through the results of search engine or examining the clues into source code documentation is time consuming [22]. A study on the GNOME[1] project reported that 30 months’ time is needed for the contributor to understand the software code and to make a contribution [23]. Developers gradually become productive taking more than a year’s time on a project to reach productivity plateau [24]. The time to complete distributed tasks is estimated to be three times longer than for co-located tasks [25].

The time required by a new person to learn the inner workings of the project when experienced contributors leave, causes considerable productivity loss [17]. In-depth understanding of software code and interconnecting file structure is not required to complete simple tasks. On the other hand, contributors may have difficulty performing

[1] GNOME is a well-known large libre project sponsored by several companies. https://www.gnome.org/foundation/.

non-trivial tasks due to ‘information blocking’, unavailability of the relevant information to complete a task [17]. The productivity of the contributor and overall project suffers due to the information blocking and a lack of understanding of the code base. According to estimates, information blocking consumes 60% of developers efforts [26].

During the preparation of a release, contributors make changes to align their work with the goals of the release [6]. As abandoned code increases on the project, the numbers of reported defects increase as well [27]. The maintenance of abandoned code is difficult because the team lacks knowledge of its creation and structure [15, 25]. The source code that remains unmaintained (unless a legacy system) has an element of uncertainty for the development team since the contributors who wrote it have left the project [17]. Removal of unmaintained code results in loss of existing functionality and may impact users of the system [6].

2.2 Contributors in Open Source Software Projects

Contributors are the knowledge workers in OSS projects. The development style in OSS projects is distinguished from CSS by the term: ‘cathedral and bazaar’ [28]. ‘Cathedral’ refers to closed approach of software development with a smaller group of developers having an access to the source code. On the contrary, ‘Bazaar’ refers to an open approach of software development with a large number of volunteers having an access to the source code to contribute on new requirements, bug fixes, and defect reporting. It is argued that a typical OSS project starts with a cathedral development and then transitions to bazaar development style [28]. In a large community of contributors, “the bazaar”, the code is under review by many, which has an effect similar to self-corrective mechanism as in peer-review process [28]. Even though the OSS code is openly accessible, the code review is conducted by limited number of contributors [29], who have earned their status by meritocracy and have proven their skills, experience and expertise while working on the project.

In OSS, each project is an equivalent of an organization in traditional software industry or CSS. The development in OSS is completed in independent, self-assigned, and parallel streams without much coordination due to geographical dispersion [6]. There are two main roles of contributors in OSS projects, developers, and users. Developers contribute in the open source community in a distributed virtual environment and users in parallel provide their feedback. In OSS, developer and the user can be the same person who may contribute the code and test the software in user’s role.

The layered structure called an onion model represents contributors in the OSS community [30]. The teams in OSS community consist of core, co-developer, active users, and passive users. The core is a small group of highly skilled and experienced members, responsible for most of the code development and ensures the design and evolution of the project. Co-developers contribute by reviewing or modifying the code or by bug fixes. Active users contribute bug reports or feature request but do not contribute any code. Passive users are the users of the code and do not make any contibution and their number is difficult to predict. However, in Linux developer organise themselves into two groups core and periphery [31]. The core in Linux project consists of project leader and hundreds of maintainers. Periphery is a large group of developers further divided into two teams: development and bug reporting.

Based on the demonstration of skills and abilities on the project, the users transcend in onion model towards becoming a core member. A contributor can simultaneously perform more than one role in the OSS project. For example, a contributor can be a core member responsible for code commits and at the same time tester of the code.

The onion model is used in the literature to assess the difference in the progress of volunteers and commercially involved developers [23]. Volunteers joining the OSS project follow the onion model and contribute based on the meritocracy, while hired developers get integrated into the project faster [23]. The reason for the difference in the progression level of volunteers and hired members on GNOME project is due to variance in knowledge accessible to both kinds of contributors.

OSS project collaborations can be of three types: community-based, non-profit organisation and commercially based. Community-based open source projects take their organisational form from an Internet-based community, and the developers are mostly the volunteers [31]. Volunteers collaborate in OSS projects in their free time and do not directly profit economically in any way from their efforts [18]. The intention of the volunteer to participate in OSS projects is to learn new skills, contribute code and develop a reputation within the OSS community that may in the future result in career opportunities [32]. Another motivation for the volunteers is related to the feeling of satisfaction, competence, and fulfilment from code writing called intrinsic motivation [33]. Managing volunteer contributors can cause certain problems not evident in traditional software development [34]. Apache project is managed by volunteers, who are otherwise full time developers. Debian is another project with 100% volunteers who are responsible for tasks including maintaining software packages, supporting the server infrastructure, developing Debian-specific software.

In non-profit organisation OSS projects, developers are either paid workers or volunteers. The project is mature enough and is funded similar to a formal organisation. There is still some element of community projects maintained in such projects, for example, Apache Software Foundation [35]. In commercially involved OSS projects, a software company sponsors projects and employs majority of contributors. A commercial company, Netscape, managed Mozilla project in the past. Companies like IBM, HP, SUN (now acquired by Oracle), sponsor OSS projects in which major contributors are paid developers [36].

Such a vast community of OSS project contributors and diverse collaborations raise concerns on acquiring distributed knowledge on software development. Software development knowledge is said to be distributed among developers [37]. In OSS projects, a small subset of contributors called core members make major code contributions (80%) [38]. Knowledge when distributed among a small group of contributors in OSS projects, one person leaving can cause the loss of 80% files in the system [15]. On the contrary, when knowledge distributes across a larger group of contributors, one person leaving causes minimum loss of files, as seen in the case of Linux project [15]. OSS projects require uniform distribution of knowledge with a mechanism that resonates with its dynamic work structure.

2.3 Knowledge Management in Open Source Software

Knowledge Management is one of the social processes and a major area of research in Open Source Software (OSS). OSS development is a knowledge-intensive activity and managing knowledge is a challenging task [4]. In this section we identify the knowledge related activities in OSS projects namely knowledge creation and knowledge sharing. Further, we discuss the knowledge barriers faced in OSS projects and details on knowledge retention process used in organisations.

2.3.1 Knowledge Creation in OSS Projects

Knowledge creation in OSS differs from the CSS [31]. A comparison of knowledge creation in OSS and CSS is given based on the five organising principles: Intellectual property ownership, membership restrictions, authority and incentives, knowledge distribution across organisational and geographical boundaries, and dominant mode of communication [31]. In case of CSS, the knowledge is owned by the organisation with an access given to employees hired. The employees are paid for their work, and the knowledge distribution is within the boundaries of the firm, mostly with face-to-face communication. While in the case of community based model the knowledge and membership is open to public and contribution is from members (mostly volunteers). Distribution of knowledge in community based models extends outside the community, and the dominant mode of communication is technology based (similar to CSS distributed development).

In OSS, the knowledge creation follows community based model and involves interaction of contributors on a larger scale than in CSS. Knowledge creation takes place when individuals are collectively working and interacting on a task and are constantly acquiring relevant information. Knowledge creation is through social interaction among individuals and organisations, and it is dynamic in nature [11].

Nonaka et al. proposed knowledge creation process, they explain conversion of tacit knowledge to explicit knowledge, which is then "crystalized." Explicit knowledge is retained by the relevant organisation and becomes the basis of new knowledge [11]. The process of knowledge creation is based on four modes of knowledge conversion: *Socialisation, Externalisation, Internalisation,* and *Combination* are coined as *SECI*. *Socialisation* is the sharing of experience and results in the creation of new tacit knowledge from the existing tacit knowledge. *Externalisation* is the conversion of tacit knowledge to explicit knowledge. Externalisation results into the articulated knowledge. *Combination* is the addition of the new explicit knowledge to the existing explicit knowledge in the knowledge system. *Internalisation* is the conversion of explicit knowledge to tacit knowledge. In internalisation, knowledge is acquired from artifacts in explicit form, and new mental models are created again resulting in tacit knowledge.

Nonaka's knowledge assets are produced as the results of inputs and outputs of the knowledge creation process SECI [11]. Knowledge created from socialisation among project members, results in intangible knowledge based on skills and expertise. Knowledge created from socialisation is made explicit through externalisation and results in conceptual knowledge assets such as product's concepts and design. Knowledge integrated with the existing explicit knowledge through combination results in systemic knowledge assets. Examples of systemic knowledge assets are documentations, specifications, and manuals. The explicit knowledge, when acquired by an individual converts to tacit knowledge by internalisation results in routine knowledge assets. The examples of routine knowledge assets are know-how in daily operations, organisation routines and operations.

The process of knowledge creation as detailed through SECI, can be used to understand knowledge creation in OSS projects. In OSS projects, contributors acquire knowledge from communication channels like Internet Relay Chat (IRC), mailing lists, posting on blogs and online resources. As a result, the new tacit knowledge is created similar to socialisation mode in SECI. The resulting communication is in explicit form but not very well structured. A Conversion to externalisation mode will apply to OSS projects, if the unstructured information is formally documented and made available to OSS community. Even though tacit knowledge to some extent is converted to explicit but it remains in unstructured form and is not readily available for reuse. Further, it is also time consuming for the contributors to search for the required information through unstructured communication archives. The combination and internalisation mode of SECI are not traceable in OSS projects. Knowledge loss occurs when during the process of knowledge creation tacit knowledge is not made explicit and is not retained for future reuse [39].

2.3.2 Knowledge Sharing in OSS Projects

In OSS projects, knowledge sharing is an ongoing activity in an intensely people-oriented and self-organised community [40]. As we shall demonstrate, this activity might also be considered to be characterised as both reactive and somewhat disorganised. In such a setup, knowledge is dispersed in the community of contributors interacting on a project and is not limited within a small group [41]. Knowledge sharing is through asynchronous means of communication and with a collection of artifacts, which are publicly available for reuse.

Knowledge is stored in repositories namely Concurrent Versions Systems (CVS), Subversion (SVN), Frequently Asked Questions (FAQs), project websites, blogs, bug reporting and bug tracking databases (e.g. Bug Tracking System BTS). Knowledge is also believed to be archived in the artefacts available to public such as mailing lists and knowledge sharing can be quantified by analysing the mailing lists exchange among the listed members in OSS projects [40]. The contributors in CSS share software coding knowledge as a part of their job, while contributors in OSS share knowledge voluntarily [42]. It is argued that contributors in OSS communities involve in free advice and tacit knowledge sharing to a larger extent than formal CSS organisations [42]. In OSS communities knowledge sharing can be associated to social motivation [43]. Social motivation such as supportive behaviour influences the behaviour of contributors and their performance. There is also intrinsic motivation for the knowledge provider such as altruism or learning, by helping others solve problems.

Connecting contributors in a social network also enhances mutual knowledge and skills among them [32]. A strong social network and without any extrinsic reward system may result in effective knowledge sharing [42]. A formal coordination mechanism can provide better visibility of contributions from other team members. Contributors can be more informed about the contributions made by others members working in OSS projects.

Gamification is another emerging form of knowledge sharing in OSS communities [44]. The community members vote upon the questions and answers posted on a site, the numbers of votes reflect on the poster's reputation and seen as a measure of their expertise by the potential employer. Gamification element on sites is found to have increased the engagement of the participants and popularity of the site. In OSS community gamification element is argued to provide a better visibility of contributors activities [44].

The social media sites also serve for contributors to learn, collaborate, share knowledge and interact with users of software [44]. Contributors contribute on software development social media sites such as GitHub for coding, Jira to track issues, StackExchange network for question answer website, StackOverflow for professional programmers, and CrossValidated for statisticians and data miners. Although knowledge sharing activates are taking place in OSS projects, the mechanisms to articulate tacit knowledge are non-existing.

2.3.3 Knowledge Barriers and Knowledge Retention in OSS

While *knowledge barriers* cause inhibition in the innovation and learning process of organisations, knowledge retention (KR) is the ultimate goal of an organisation striving to innovate and improve performance. The inaccessibility to a certain kind of knowledge can delay the contributions on development activities by a contributor [45]. We focus on two kinds of barriers namely *contribution barriers* and *knowledge sharing barriers*. The limited knowledge of programming language, difficulty level of algorithms, complexity of technologies and source code used in OSS, can cause contribution barriers [46]. Computer languages are complex and difficult to learn with intertwined modules, so an understanding of existing architecture is required to contribute to an inter-dependent module [46]. The barriers for the newcomers to contribute in a OSS project are the lack of knowledge on project practices, lack of documentation, understanding information flow, unclear comments, and outdated documentation [45].

Knowledge sharing barriers are categorised into three levels: individual, organisational and technological [47]. Individual level barriers that limit knowledge sharing are a lack of time, lack of trust, a person who is unconsciously not aware of the possessed knowledge and lack of social network. While discussing distributed global communities to facilitate knowledge, language barriers, lack of common terminology, and lack of trust all inhibit knowledge sharing [12]. On the organisational level, barriers including non-supporting environment and culture lead to unsuccessful knowledge sharing. On the technological level, barriers to knowledge sharing are a lack of training, lack of communication on the benefits of technology, unsuitable technology, and reluctance to use technology.

The removal of obstacles due to knowledge flow in projects has the potential for a decrease in labour cost, improved schedule observance, and better final product quality [48]. The top five problematic knowledge flows were divided into two categories: difficulties with the online storage and retrieval of documents, and intra-team communication. The first category relates to explicit knowledge flow problems, while the second relates to tacit knowledge [48]. In addition to the identification of knowledge barriers, KR processes are also required within an organisation for knowledge to be accessible for the future reuse. In OSS projects, KR processes do not exist as in CSS organisations. Knowledge retention relates to capturing knowledge in an organisation and is an important aspect of KM. Knowledge retention mainly comes into focus when an employee is leaving. Three things indicate the need of a KR mechanism in an organisation: Lack of knowledge and an overly long time to acquire it is due to steep learning curves; People repeating mistakes and performing rework because they forget what they learned from previous projects; Chances of individuals owning key knowledge becoming unavailable [49].

Knowledge retention can be seen as a way of embedding and enabling knowledge within an organisation and a critical factor for sustainable performance [50]. It is an effort-demanding task to identify potential knowledge for the organisation. The structure of the organisation in the context of how well it supports knowledge retention is of importance. Once the person who has the potential knowledge leaves the organisation, it is hard to retain this knowledge.

Codification and personalisation are considered useful strategies for knowledge bases to be further used in knowledge intensive activities like software development [51]. In knowledge bases, codification captures electronic information and personalisation deals with the ways humans' use and process knowledge. Organizations implement codification strategy to encourage the reuse of explicit knowledge. The core techniques designed to retain knowledge in an organisation are mainly dependent on its knowledge-sharing practices. The techniques that facilitate knowledge capture, sharing, and reapplication are after-action reviews, communities of practice, face-to-face meetings, mentoring programs, expert referral services, video conferencing, interviews, written reports, use of training and technology-based systems to transfer the knowledge [12].

3 Discussion

In a large, geographically dispersed and dynamic OSS community, contributors vary in their skills and experiences. The quality of contributions (mostly the source code) on the projects reflects a contributor's expertise and skills [15]. Knowledge sharing in OSS communities is mainly by asynchronous communication and typically involves mailing lists, blogs, forums, and Internet Relay Chat (IRC). Researchers have utilised OSS project mailing list data in various studies and it is thought to be one of the primary communication mechanisms in OSS projects [52]. However, the knowledge shared suffers from only partial coverage [17] and it can lack effective levels of organisation. OSS project knowledge may be abruptly lost when volunteers cease to contribute, and with knowledge not shared (existing in tacit form), the impact on the overall health of the project can be very damaging [17]. In effect, the stability of OSS projects and their success are dependent on contributor retention [53], or perhaps more precisely on the retention of knowledge contributor [54] either through directly sustaining contributors on the project or by co-opting individual knowledge into the collective knowledge sphere.

The removal of knowledge flow obstacles in projects has the potential for a decrease in labour cost, improved schedule observance, and better final product quality [48]. We propose that certain proactive knowledge acquisition practices will reduce the total cost of knowledge exchange in OSS projects, thereby improving the project productivity.

Techniques to identify the critical knowledge will be a necessary first step to improving the current position, though we expect that there will be a challenge in striking the appropriate balance between proactive and reactive knowledge management, and this must somehow take account of the preference of contributors for these two different styles of knowledge exchange.

As we have demonstrated, OSS communities are mainly composed of volunteers who cannot be constrained to work permanently on the project [18] or to share their knowledge. The challenging task is how to orchestrate knowledge management in such a dynamic and dispersed community as OSS, especially as open source projects become larger and more widely adopted. This we suggest is not just a concern for the custodians of OSS projects but also for the consumers of the OSS itself. A private company may be motivated not just by the immediate cost saving in adopting an OSS project, but they may also be concerned with the maintainability of the OSS into the future as a strategic product development decision. In this respect, we envisage that a set of OSS knowledge management principles may be a product of our research and we have already undertaken some limited work in this direction.

Having established the absence of research on knowledge loss in the OSS project space, we propose to undertake a sustained investigation of this problem and to aid this exercise; we have established the following two research questions. We expect that further research questions will be identified as our research evolves.

- **RQ1.** Which knowledge management practices enable an effective knowledge management strategy for OSS projects?
- **RQ2.** How can knowledge management practices be integrated with established work practices in OSS projects?

4 Conclusion

From our review of the related literature, we conclude that knowledge management in OSS projects has received only indirect or superficial treatment, and we have found no single substantial examination of the reactive and proactive knowledge strategy for OSS projects. In OSS projects, contributors are not obliged to notify the project community when they leave. The general mechanism of knowledge retention in software firms may sometimes be reactive in nature, triggered when an employee is leaving but even then, the opportunity for knowledge repatriation into the organisation will endure at least to the extent that the employee is cooperative and within the notice period that is typical in contemporary employment contracts. Conversely, in OSS projects a contributor may simply fall off the project radar – without notice and perhaps also unnoticed by the project – thereby eliminating any opportunity for reactive knowledge repatriation. Therefore, a proactive approach to retain knowledge is instinctively appealing for OSS projects.

In summary, we have investigated the published literature into knowledge loss in OSS projects, finding that there has been insufficient treatment of this concern to date. We also find that given the nature of OSS projects, proactive knowledge management mechanisms may be especially important, for example, because of the highly fragmented and transient nature of OSS project contributors. Given the popularity of OSS and its widespread and growing adoption, we believe that there is benefit in examining mechanisms to promote proactive knowledge management in OSS projects, and that these benefits can be shared by both contributors to and consumers of OSS.

Acknowledgments. This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 to Lero, the Irish Software Research Centre (www.lero.ie).

References

1. Feller, J., Fitzgerald, B.: Understanding Open Source Software Development. Addison-Wesley, London (2002)
2. Crowston, K., Howison, J., Annabi, H.: Information systems success in free and open source software development: theory and measures. Softw. Process Improv. Pract. **11**, 123–148 (2006)
3. Subramaniam, C., Sen, R., Nelson, M.L.: Determinants of open source software project success: a longitudinal study. Decis. Support Syst. **46**, 576–585 (2009)
4. Crowston, K., Wei, K., Howison, J., Wiggins, A.: Free/libre open-source software development: what we know and what we do not know. ACM Comput. Surv. **44**, 7 (2012)
5. 78% of Companies Run On Open Source Yet Lack Formal Policies|Black Duck Software. Black Duck Software, N.p. Web, 8 June 2017
6. Michlmayr, M.: Quality Improvement in Volunteer Free and Open Source Software Projects: Exploring the Impact of Release Management. University of Cambridge (2007)
7. Capiluppi, A., Gonzalez-Barahona, J.M., Herraiz, I., Robles, G.: Adapting the “staged model for software evolution” to free/libre/open source software. In: Ninth International Workshop on Principles of Software Evolution: in Conjunction with the 6th ESEC/FSE Joint Meeting, pp. 79–82. ACM, Dubrovnik (2007)
8. Robles, G., Gonzalez-Barahona, J.M.: Contributor turnover in libre software projects. IFIP International Federation for Information Processing, vol. 203, pp. 273–286 (2006)
9. Davenport, T.H., Prusak, L.: Working Knowledge: How Organizations Manage What They Know. Harvard Business Press, Boston (1998)
10. Zack, M.H.: Managing codified knowledge. Sloan Manage. Rev. **40**, 45–58 (1999)
11. Nonaka, I., Toyama, R., Konno, N.: SECI, Ba and leadership: a unified model of dynamic knowledge creation. Long Range Plan. **33**, 5–34 (2000)
12. De Long, D.W., Davenport, T.: Better practices for retaining organizational knowledge: lessons from the leading edge. Employ. Relat. Today **30**, 51–63 (2003)
13. Jennex, M.E., Durcikova, A.: Assessing knowledge loss risk. In: 2013 46th Hawaii International Conference on System Sciences (HICSS), pp. 3478–3487. IEEE (2013)
14. Viana, D., Conte, T., Marczak, S., Ferreira, R., Souza, C.D.: Knowledge creation and loss within a software organization: an exploratory case study. In: 2015 48th Hawaii International Conference on System Sciences (HICSS), pp. 3980–3989 (2015)
15. Donadelli, S.M.: The impact of knowledge loss on software projects: turnover, customer found defects, and dormant files. Software Engineering, p. 85. Concordia University (2015)
16. Rigby, P.C., Zhu, Y.C., Donadelli, S.M., Mockus, A.: Quantifying and mitigating turnover-induced knowledge loss: case studies of Chrome and a project at Avaya. In: Proceedings of the 2016 International Conference on Software Engineering (2016)

17. Izquierdo-Cortazar, D., Robles, G., Ortega, F., Gonzalez-Barahona, J.M.: Using software archaeology to measure knowledge loss in software projects due to developer turnover. In: 42nd Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2009)
18. Robles, G., Gonzalez-Barahona, J.M., Michlmayr, M.: Evolution of volunteer participation in libre software projects: evidence from Debian. In: Proceedings of the 1st International Conference on Open Source Systems, pp. 100–107 (2005)
19. Dingsoyr, T., Bjornson, F.O., Shull, F.: What do we know about knowledge management? Practical implications for software engineering. IEEE Softw. **26**, 100–103 (2009)
20. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, p. 38. ACM (2014)
21. Foucault, M., Palyart, M., Blanc, X., Murphy, G.C., Falleri, J.-R.: Impact of developer turnover on quality in open-source software. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, pp. 829–841. ACM, Bergamo (2015)
22. von Krogh, G., Spaeth, S., Haefliger, S.: Knowledge reuse in open source software: an exploratory study of 15 open source projects. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, p. 198b. IEEE (2005)
23. Herraiz, I., Robles, G., Amor, J.J., Romera, T., González Barahona, J.M.: The processes of joining in global distributed software projects. In: Proceedings of the 2006 International Workshop on Global Software Development for the Practitioner, pp. 27–33. ACM, Shanghai (2006)
24. Zhou, M., Mockus, A.: Developer fluency: achieving true mastery in software projects. In: Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 137–146. ACM, Santa Fe (2010)
25. Herbsleb, J.D., Mockus, A.: An empirical study of speed and communication in globally distributed software development. IEEE Trans. Soft. Eng. **29**, 481–494 (2003)
26. Liu, W., Chen, C.L., Lakshminarayanan, V., Perry, D.E.: A design for evidence - based soft research. SIGSOFT Softw. Eng. Notes **30**, 1–7 (2005)
27. Otte, T., Moreton, R., Knoell, H.D.: Applied quality assurance methods under the open source development model. In: 32nd Annual IEEE International Computer Software and Applications, COMPSAC 2008, pp. 1247–1252 (2008)
28. Capiluppi, A., Michlmayr, M.: From the cathedral to the bazaar: an empirical study of the lifecycle of volunteer community projects. In: Feller, J., Fitzgerald, B., Scacchi, W., Sillitti, A. (eds.) OSS 2007. ITIFIP, vol. 234, pp. 31–44. Springer, Boston, MA (2007). doi:10.1007/978-0-387-72486-7_3
29. Rigby, P.C., German, D.M., Cowen, L., Storey, M.-A.: Peer review on open-source software projects: parameters, statistical models, and theory. ACM Trans. Softw. Eng. Methodol. **23**, 1–33 (2014)
30. Crowston, K., Howison, J.: The social structure of free and open source software development. First Monday **10**(2) (2005)
31. Lee, G.K., Cole, R.E.: From a firm-based to a community-based model of knowledge creation: the case of the Linux Kernel development. Organ. Sci. **14**, 633–649 (2003)
32. Crowston, K.: Lessons from volunteering and free/libre open source software development for the future of work. In: Chiasson, M., Henfridsson, O., Karsten, H., DeGross, J.I. (eds.) Researching the Future in Information Systems, pp. 215–229. Springer, Heidelberg (2011)
33. Schilling, A., Laumer, S., Weitzel, T.: Is the source strong with you? A fit perspective to predict sustained participation of FLOSS developers. In: International Conference on Information Systems 2011, ICIS 2011, pp. 1620–1630 (2011)

34. Michlmayr, M., Robles, G., Gonzalez-Barahona, J.M.: Volunteers in large libre software projects: a quantitative analysis over time. In: Emerging Free and Open Source Software Practices, pp. 1–24 (2007)
35. Xu, B.: Volunteers' Participative Behaviors in Open Source Software Development: The Role of Extrinsic Incentive, Intrinsic Motivation and Relational Social Capital. Texas Tech University (2006)
36. Fitzgerald, B.: The transformation of open source software. MIS Q. **30**, 587–598 (2006)
37. Yunwen, Y., Yamamoto, Y., Kishida, K.: Dynamic community: a new conceptual framework for supporting knowledge collaboration in software development. In: 11th Asia-Pacific Software Engineering Conference, pp. 472–481 (2004)
38. Mockus, A., Fielding, R.T., Herbsleb, J.: A case study of open source software development: the Apache server. In: Proceedings of the 22nd International Conference on Software Engineering, pp. 263–272. ACM, Limerick (2000)
39. Assimakopoulos, D., Yan, J.: Sources of knowledge acquisition for Chinese software engineers. R&D Manage. **36**, 97–106 (2006)
40. Sowe, S.K., Stamelos, I., Angelis, L.: Understanding knowledge sharing activities in free/open source software projects: an empirical study. J. Syst. Softw. **81**, 431–446 (2008)
41. Ye, Y., Kishida, K.: Toward an understanding of the motivation open source software developers. In: Proceedings of the 25th International Conference on Software Engineering, pp. 419–429. IEEE Computer Society, Portland (2003)
42. Endres, M.L., Endres, S.P., Chowdhury, S.K., Alam, I.: Tacit knowledge sharing, self-efficacy theory, and application to the open source community. J. Knowl. Manage. **11**, 92–103 (2007)
43. Licorish, S.A., MacDonell, S.G.: Understanding the attitudes, knowledge sharing behaviors and task performance of core developers: A longitudinal study. Inf. Softw. Technol. **56**, 1578–1596 (2014)
44. Vasilescu, B., Serebrenik, A., Devanbu, P., Filkov, V.: How social Q&A sites are changing knowledge sharing in open source software communities. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 342–354. ACM, Baltimore (2014)
45. Steinmacher, I., Silva, M.A.G., Gerosa, M.A., Redmiles, D.F.: A systematic literature review on the barriers faced by newcomers to open source software projects. Inf. Softw. Technol. **59**, 67–85 (2015)
46. von Krogh, G., Spaeth, S., Lakhani, K.R.: Community, joining, and specialization in open source software innovation: a case study. Res. Policy **32**, 1217–1241 (2003)
47. Kukko, M., Helander, N.: Knowledge sharing barriers in growing software companies. In: 45th Hawaii International Conference on System Science, pp. 3756–3765 (2012)
48. Mitchell, S.M., Seaman, C.B.: Software process improvement through the identification and removal of project-level knowledge flow obstacles. In: Proceedings of the 34th International Conference on Software Engineering, pp. 1265–1268. IEEE (2012)
49. Lindvall, M., Rus, I.: Knowledge Management for Software Organizations. In: Aurum, A., Jeffery, R., Wohlin, C., Handzic, M. (eds.) Managing Software Engineering Knowledge, pp. 73–94. Springer, Heidelberg (2003)
50. Doan, Q.M., Rosenthal-Sabroux, C., Grundstein, M.: A reference model for knowledge retention within small and medium-sized enterprises. In: KMIS, pp. 306–311 (2011)
51. Donnellan, B., Fitzgerald, B., Lake, B., Sturdy, J.: Implementing an open source knowledge base. IEEE Softw. **22**, 92–95 (2005)

52. Sharif, K.Y., English, M., Ali, N., Exton, C., Collins, J.J., Buckley, J.: An empirically-based characterization and quantification of information seeking through mailing lists during Open Source developers' software evolution. Inf. Softw. Technol. **57**, 77–94 (2015)
53. Schilling, A., Laumer, S., Weitzel, T.: Who will remain? An evaluation of actual person-job and person-team fit to predict developer retention in FLOSS projects. In: 45th Hawaii International Conference on System Science, pp. 3446–3455 (2012)
54. Ayushi, R., Ashish, S.: What community contribution pattern says about stability of software project? In: 21st Asia-Pacific Software Engineering Conference, pp. 31–34 (2014)

Education Issues in SPI

Relating Student, Teacher and Third-Party Assessments in a Bachelor Capstone Project

Vincent Ribaud[1(✉)] and Vincent Leilde[2]

[1] Lab-STICC, Team MOCS, Université de Brest, Brest, France
ribaud@univ-brest.fr
[2] Lab-STICC, Team MOCS, ENSTA Bretagne,
3 Rue François Verny, Brest, France

Abstract. The capstone is arguably the most important course in any engineering program because it provides a culminating experience and is often the only course intended to develop non-technical, but essential skills. In a software development, the capstone runs from requirements to qualification testing. Indeed, the project progress is sustained by software processes. This paper yields different settings where students, teachers and third-party assessors performed [self-] assessment and the paper analyses corresponding correlation coefficients. The paper presents also some aspects of the bachelor capstone. A research question aims to seek if an external process assessment can be replaced or completed with students' self-assessment. Our initial findings were presented at the International Workshop on Software Process Education Training and Professionalism (IWSPETP) 2015 in Gothenburg, Sweden and we aimed to improve the assessment using teacher and third-party assessments. Revised findings show that, if they are related to curriculum topics, students and teacher assessments are correlated but that external assessment is not suitable in an academic context.

Keywords: Process assessment · Competencies model · Capstone project

1 Introduction

Project experience for graduates of computer science programs has the following characteristic in the ACM Computer Science Curricula [1]: "*To ensure that graduates can successfully apply the knowledge they have gained, all graduates of computer science programs should have been involved in at least one substantial project. […] Such projects should challenge students by being integrative, requiring evaluation of potential solutions, and requiring work on a larger scale than typical course projects. Students should have opportunities to develop their interpersonal communication skills as part of their project experience.*" The capstone is arguably the most important course in any engineering program because it provides a culminating experience and is often the only course used to develop non-technical, but essential skills [2]. Many programs run capstone projects in different settings [3–8]. The capstone project is intended to provide students with a learning by doing approach about software development, from requirements to qualification testing. Indeed, the project progress is

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 499–506, 2017.
DOI: 10.1007/978-3-319-67383-7_36

sustained by software processes. Within the ISO/IEC 15504 series and the ISO/IEC 330xx family of standards, process assessment is used for process improvement and/or process capability determination. Process assessment helps students to be conscious about and improve what they are doing. Hence, a capstone teacher's activity is to assist students with appreciation and guidance, a task that relies on the assessment of students' practices and students' products. This paper yields different settings where students, teachers and third-party assessors performed [self-] assessment and analyses correlation coefficients. Incidentally, the paper presents some aspects of the bachelor capstone project at Brest University. Data collection started 3 years ago. Initial findings were presented in [9].

The paper structure is: Sect. 2 overviews process assessment, Sect. 3 presents different settings we carried process assessments; we finish with a conclusion.

2 Process Assessment

Most software engineering educators will agree that the main goal of the capstone project is to learn by doing a simplified cycle of software development through a somewhat realistic project. For instance, Dascalu et al. use a "streamlined" version of a traditional software development process [3]. Umphress et al. state that using software processes in the classroom helps in three ways: 1 - processes describe the tasks that students must accomplish to build software; 2 - processes can give the instructor visibility into the project; 3 - processes can provide continuity and corporate memory across academic terms [4]. Consequently, the exposition to some kind of process assessment is considered as a side-effect goal of the capstone project. It is a conventional assertion that assessment drives learning [10]; hence process assessment drives processes learning. Conventionally, a process is seen as a set of activities or tasks, converting inputs into outputs [11]. This definition is not suited for process assessment. Rout states that "*it is of more value to explore the purpose for which the process is employed. Implementing a process results in the achievement of a number of observable outcomes, which together demonstrate achievement of the process purpose* [12]." This approach is used to specify processes in a Process Reference Model (PRM). We use a small subset of the ISO/IEC 15504-5:2012 Exemplar Process Assessment Model that includes a PRM (replicated from the ISO/IEC 12207:2008), mainly the Software Processes of the ENG Process Group [13]: ENG.3 System architectural design, ENG.4 Software requirements analysis, ENG.5 Software design, ENG.6 Software construction, ENG.7 Software integration, ENG.8 Software testing. Process Purpose, Process Outcomes, Base Practices (BP) have been kept without any modification; Input and Outputs Work Products (WP) have been set to main products.

From an individual perspective, the ISO/IEC 15504 Exemplar Process Assessment Model (PAM) is seen as a competencies model related to the knowledge, skills and attitudes involved in a software project. A competencies model defines and organizes the elements of a curriculum (or a professional baseline) and their relationships. During the capstone project, all the students use the model and self-assess their progress.

A hierarchical model is easy to manage and use. We kept the hierarchical decomposition issued from the ISO/IEC 15504 Exemplar PAM: process groups –

process – base practices and products. A competency model is decomposed into competency areas (mapping to process groups); each area corresponding to one of the main division of the profession or of a curriculum. Each area organizes the competencies into families (mapping to processes). A family corresponds to main activities of the area. Each family is made of a set of knowledge and abilities (mapping to base practices), called competencies; each of these entities is represented by a designation and a description. The ability model and its associated tool eCompas have been presented in [14].

2.1 Process Assessment

The technique of process assessment is essentially a measurement activity. Within ISO/IEC 15504, process assessment has been applied to a characteristic termed process capability, defined as "*a characterization of the ability of a process to meet current or projected business goals*" [13]. It is now replaced in the 330xx family of standards by the larger concept of process quality, defined as "*ability of a process to satisfy stated and implied stakeholders needs when used in a specific context* [15]. In ISO/IEC 33020:2015, process capability is defined on a six point ordinal scale that enables capability to be assessed from the bottom of the scale, *Incomplete*, through the top end of the scale, *Innovating* [16]. We see Capability Level 1, *Performed*, as an achievement: through the performance of necessary actions and the presence of appropriate input and output work products, the process achieves its process purpose and outcomes. Hence, Capability Level 1 will be the goal and the assessment focus.

If students are able to perform a process, it denotes a successful learning of software processes, and teachers' assessments rate this capability. Because we believe that learning is sustained by continuous, self-directed assessment, done by teachers or a third-party, the research question aims to state how students' self-assessment and teacher's assessment are correlated and if self-assessment of BPs and WPs is an alternative to external assessment about ISO/IEC 15504 Capability Level 1. Obviously, the main goal of assessment is students' ability to perform the selected processes set.

3 The Capstone Project

3.1 Overview

Schedule: The curriculum is a 3-year Bachelor of Computer Science. The project is performed during two periods. The first period is dispatched all the semester along and homework is required. The second period (2 weeks) happens after the final exams and before students' internship. Students are familiar with the Author-Reader cycle: each deliverable can be reviewed as much as needed by the teacher that provides students with comments and suggestions. It is called Continuous Assessment in [5, 6].

System Architecture: The system is made of 2 sub-systems: PocketAgenda (PA) for address books and agenda management and interface with a central directory; WhoIsWho (WIW) for managing the directory and a social network. PocketAgenda is implemented with Java, JSF relying on an Oracle RDBMS. WhoIsWho is implemented in Java using a RDBMS. Both sub-systems communicate with a protocol to establish using UDP. The system is delivered in 3 batches. Batch 0 established and analyzed requirements. Batch 1 performed collaborative architectural design, separate client and server development, integration. Batch 2 is focused on information system development.

Students consent: Students were advised that they can freely participate to the experiment described in this paper. The class contains 29 students, all agreed to participate; 4 did not complete the project and do not take part to the study. Students have to regularly update the competencies model consisting in the ENG process group, the 6 processes above and their Base Practices and main Work Products and self-assess on an achievement scale: Not - Partially - Largely - Full. There will be also teacher and third-party assessments that will be anonymously joined to self-assessments by volunteer students.

3.2 Batch 0: Writing and Analyzing Requirements

Batch 0 is intended to capture, write and manage requirements through use cases. It is a non-technical task not familiar to students. In [7], requirements are discussed as one of the four challenges for capstone projects. Students use an iterative process of writing and reviewing by the teacher. Usually, 3 cycles are required to achieve the task. Table 1 presents the correlation coefficient r between student and teacher assessment for the ENG.4 Software requirements analysis. It relies on 3 BPs and 2 WPs. Table 2 presents also the average assessment for each assessed item. The overall correlation coefficient relates 25 * 6 = 150 self-assessment measures with the corresponding teacher assessment measures, its value `r = 0.64` indicates a correlation.

Table 1. ENG.4 assessment (self and teacher)

	Stud. avg	Tch. avg	r
BP1: Specify software requirements	2.12	1.84	0.31
BP3: Develop criteria for software testing	1.76	1.76	1.00
BP4: Ensure consistency	1.92	0.88	0.29
17-8 Interface requirements	1.88	1.88	1.00
17-11 Software requirements	2.08	2.08	1.00

Table 2. ENG.3 (self, teacher and third-party)

	Stud. avg	Tch. avg	3-party avg.	r Std-Tch	r Std-3party
BP1: Describe system architecture	2.24	2.02	1.68	–0.22	0.18
BP3. Define interfaces	1.96	2.16	1.56	0.48	0.36
BP4. Ensure consistency	2	1.72	0.88	0	0.44
04-01 Database design	2.48	2.2	1.88	0.49	0.35
04-04 High level design	2.12	1.84	1.64	0.37	–0.11

Thanks to the Author-Reader cycle, specification writing iterates several time during the semester and the final mark given to almost 17-8 Interface requirements and 17-11 Software requirement documents was Fully Achieved. Hence correlation between students and teacher assessments is complete. However, students mistake documents assessment for the BP1: Specify software requirements. Documents were improved through the author-reader cycle, but only reflective students improve their practices accordingly. Also, students did not understand the ENG.4. BP4: Ensure consistency and failed the self-assessment. Most students did not take any interest in traceability and self-assessed at a much higher level that the teacher did.

A special set of values can bias a correlation coefficient; if we remove the BP4: Ensure consistency assessment, we get `r = 0.89`, indicating an effective correlation. However, a bias still exists because students are mostly self-assessing using the continuous feedback they got from the teacher during the Author-Reader cycle. Students reported that they wrote use cases from a statement of work for the first time and that they could not have succeeded without the Author-Reader cycle.

3.3 Batch 1: A Client-Server Endeavor

For the batch 1, students have to work closely in pairs, to produce architectural design and interface specification and to integrate the client and server sub-systems, each sub-system being designed, developed and tested by one student. Defining the high-level architecture, producing the medium and low-level design are typical activities of the design phase [3]. 4 pairs failed to work together and split, consequently lonesome students worked alone and have to develop both sub-systems. We were aware of two biases: 1 - students interpret the teacher's feedback to self-assess accordingly; 2 - relationship issues might prevent teachers to assess students to their effective level. Hence, for ENG.3 System architectural design process and ENG.7 Software integration process, in addition to teachers' assessment, another teacher, experienced in ISO/IEC 15504 assessments, acted as a third-party assessor.

Architectural design: For the ENG.3 System architectural design, Table 2 presents the correlation coefficient between student and teacher assessments and the correlation coefficient between student and third-party assessments. Assessment relies on 3 BPs and 2 WPs. Table 2 presents also the average assessment for each assessed item. The correlation coefficient between self-assessment and teacher assessment measures is `r1 = 0.28` and the correlation coefficient between self-assessment and third-party assessment measures is `r2 = 0.24`. There is no real indication for a correlation.

Detailed correlation is poor, except maybe for database design and interface design, but these technical topics are deeply addressed in the curriculum. An half of students perform a very superficial architectural work because they are eager to jump to the code. They believe that the work is fair enough but teachers do not. The BP4. Ensure consistency is a traceability matter that suffers the same problem described above. A similar concern to requirements arose: most students took Work Products (Design Documents) assessment as an indication of their achievement.

Students reported that requirement analysis greatly helped to figure out the system behavior and facilitated the design phase and interface specification. However, students had never really learnt architectural design and interface between sub-systems, indeed it explains the low third-party assessment average for BPS and WPs.

Integration: ENG.7 Software integration is assessed with 6 main Base Practices and 2 Work Products. The correlation coefficient between self and teacher assessments is `r1 = -0.03` and the correlation coefficient between self and third-party assessments is `r2 = 0.31`. However, several BPs or WPs were assessed by the third-party assessor with the same mark for all students (N or P): the standard deviation is zero and the correlation coefficient is biased and was not used. Table 3 presents the assessment average for the third types of assessment.

Table 3. ENG.7 indicators

	Stud. avg	Tch. avg	3-party avg.
BP1: Develop software integration strategy	1.56	1.20	0.40
BP2: Develop tests for integrated software items	2.08	1.08	0.52
BP3: Integrate software item	2.00	2.12	1.76
BP4: Test integrated software items	2.00	1.80	1.16
BP5. Ensure consistency	1.76	1.20	0.72
BP6: Regression test integrated software items	1.64	0.52	0.2
08-10 Software integration test plan	1.44	0.88	0.00
11-01 Software product	2.04	2.12	1.48

All BPs and WPs related to integration and test are weakly third-party assessed, indicating that students are not really aware of these topics, a common hole in a Bachelor curriculum. Some students were aware of the poor maturity of the integrated product, partly due to the lack of testing. Although the Junit framework has been taught during the first semester, some students did not see the point to use it while some others did not see how to use it for the project. As mentioned by [4], we came to doubt the veracity of process data we collected. Students reported that they appreciated the high-level discipline that the capstone imposed, but they balked at the details.

3.4 Batch 2: Information System Development

For the batch 2, students have to work loosely in pairs; each of the two has developed different components of the information system and has been assessed individually. Table 4 presents the correlation coefficient `r` between student and teacher assessment for the ENG.6 Software construction process. It relies on 4 Base Practices and 2 Work Products. Table 4 presents also the average assessment for each assessed item. The correlation coefficient is `r = 0.10` and there is no indication for a correlation.

However, BPs and WPs related to unit testing were assessed by the teacher with almost the same mark for all students (N or P), biasing the correlation coefficient. If we remove BPs and WPs related to unit testing (17-14 Test cases specification; 15-10 Test incidents report; BP1: Develop unit verification procedures), we get `r = 0.49`, indicating a possible correlation.

Table 4. ENG.6 assessment (self and teacher)

	Stud. avg	Tch. avg	r
BP1: Develop unit verification procedures	1.84	0.40	0.05
BP2: Develop software units	1.92	1.84	0.37
BP3: Ensure consistency	1.92	0.92	0.25
BP4: Verify software units	1.96	1.00	–0.2
17-14 Test cases specification	1.80	0.36	0.07
15-10 Test incidents report	1.52	0.12	–0.45

Our bachelor students have little awareness of the importance of testing, including test specification and bugs reporting. This issue has been raised by professional tutors many times during the internships but no effective solution has been found until yet. Students reported that the ENG.6 Software construction process raised a certain anxiety because students had doubt about their ability to develop a stand-alone server inter-operating with a JDeveloper application and two databases but most students succeeded. For some students, a poor Java literacy compromised the project progress. It is one problem reported by Goold: the lack of technical skills in some teams [5].

4 Conclusion

The research question aims to see how students' self-assessment and external assessment [by a teacher or a third-party] are correlated. This is not true for topics not addressed in the curriculum or unknown by students. For well-known topics, assessments are correlated roughly for the half of the study population. It might indicate that in a professional setting, where employees are skilled for the required tasks, self-assessment might be a good replacement to external assessment. Using a third-party assessment instead of coaches' assessment was not convincing. Third-party assessment is too harsh and tends to assess almost all students with the same mark. Self-knowledge or teacher's understanding tempers this rough assessment towards a finer appreciation.

The interest of a competencies model (process/BPs/WPs) is to supply a reference framework for doing the job. Software professionals may benefit from self-assessment using a competencies model in order to record abilities gained through different projects, to store annotations related to new skills, to establish snapshots in order to evaluate and recognize knowledge, skills and experience gained over long periods and in diverse contexts, including in non-formal and informal settings.

Acknowledgements. We thank all the students of the 2016-2017 final year of Bachelor in Computer Science for their agreement to participate to this study, and especially Maxens Manach and Killian Monot who collected and anonymized the assessments. We thank Laurence Duval, a teacher that coached and assessed half of the students during batch 1.

References

1. ACM: 2013 Computer Science Curricula - Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, http://www.acm.org/education/CS2013-final-report.pdf. Accessed 16 Jun 2017
2. Abbott, S. (ed.): Capstone project. The glossary of education reform. http://edglossary.org/capstone-project. Accessed 23 Mar 2016
3. Dascalu, S.M., Varol, Y.L., Harris, F.C., Westphal, B.T.: Computer science capstone course senior projects: from project idea to prototype implementation. In: Proceedings 35th Conference on Frontiers in Education, pp. S3J–1. IEEE, Indianapolis (2005)
4. Umphress, D.A., Hendrix, T.D., Cross, J.H.: Software process in the classroom: the Capstone project experience. IEEE Softw. **19**(5), 78–81 (2002)
5. Karunasekera, S., Bedse, K.: Preparing software engineering graduates for an industry career. In: Proceedings of the 30th Conference on Software Engineering Education & Training (CSEE & T), pp. 97–106. IEEE, Dublin (2007)
6. Vasilevskaya, M., Broman, D., Sandahl, K.: Assessing large-project courses: model, activities, and lessons learned. Trans. Comput. Educ. **15**(4), 30 pages (2015)
7. Bloomfield, A., Sherriff, M., Williams, K.: A service learning practicum capstone. In: 45th Technical Symposium on Computer Science Education (SIGCSE), pp. 265–270 (2014)
8. Goold, A.: Providing process for projects in capstone courses. In: Proceedings of the 8th Conference on Innovation and Technology in Computer Science Education (ITiCSE), pp. 26–29. ACM, Thessaloniki (2003)
9. Ribaud, V., Matthieu, A.B., O'Connor, R.V.: Process assessment issues in a bachelor capstone project. In: Proceedings of the 1st International Workshop on Software Process Education, Training and Professionalism (SPEPT 2015), pp. 25–33. CEUR Workshop Proceedings (2015)
10. Dollard, J., Miller, N.E.: Personality and psychotherapy; an analysis in terms of learning, thinking, and culture. McGraw-Hill, New York (1950)
11. ISO/IEC 12207:2008, Systems and software engineering–Software life cycle processes. ISO, Geneva (2008)
12. Rout, T.: The evolving picture of standardisation and certification for process assessment. In Proceedings of the 7th Conference on Quality of Information and Communications Technology (QUATIC), pp. 63–72. IEEE, Portugal (2010)
13. ISO/IEC 15504-5:2012. Information technology–Process assessment–Part 5: An exemplar software life cycle process assessment model. ISO, Geneva (2012)
14. Ribaud, V., Saliou, P.: Towards an ability model for SE apprenticeship. Innov. Teach. Learn. Inf. Comput. Sci. **6**(3), 7–107 (2007)
15. ISO/IEC 33001:2015. Information technology–Process assessment–Concepts and terminology. ISO, Geneva (2015)
16. ISO/IEC 33020:2015. Information technology–Process assessment–Process measurement framework for assessment of process capability. ISO, Geneva (2015)

Evaluation Model of PRO2PI-WORK4E Method for Teaching Software Process Improvement

Clenio F. Salviano(✉)

CTI: Centro de Tecnologia da Informação Renato Archer,
Rodovia D. Pedro I, km 143.6, Campinas, SP 13069-901, Brazil
Clenio.Salviano@cti.gov.br, Clenio.Salviano@gmail.com

Abstract. Methods to guide introduction to Software Process Improvement (SPI) courses to potentiate "learning SPI by doing SPI" and their systematic evaluations are relevant to both practice and research. An evaluation model for an educational method to teach SPI was developed. This model is based on a model for the evaluation of educational games. The developed model is composed of model design; evaluation process, objective and questionnaire; documentation model, data compilation and analysis model and spreadsheet; and example of use. The model was used to evaluate a course. Results of this evaluation provide initial validation of this model and also indicate that the evaluated method is effective in potentiate "learning SPI by doing SPI".

Keywords: Software process improvement · Teaching SPI · Evaluation model

1 Introduction

Teaching Software Process Improvement (SPI) is a challenging effort. It is an important dimension of SPI. A recent international workshop, for example, focused on the new challenges for and best practices in software process education, training and professionalism [20]. In this workshop, a systematic mapping study on SPI education concludes: "in spite of its [SPI] importance, increasing its coverage in educational settings is still challenging" [20, pp. 7–17]. Another article introduces a research to understand SPI education oriented to software industry needs [20, pp. 70–74]. Consequently, methods to teach SPI and their systematic evaluations are relevant to both SPI practice and SPI research.

To guide courses on introduction to SPI, an educational method (PRO2PI-WORK4E or simply WORK4E) has been developed, applied and evolved during the last 12 years [1]. This year EM4E, an Evaluation Model for WORK4E, was developed and applied. This article introduces EM4E. The article is organized in six sections. This Sect. 1 is a brief introduction of the article. Section 2 provides an overview of the educational method. Section 3 presents related work on evaluation of educational games and methods. Section 4 introduces the evaluation model. Section 5 presents results from an evaluation of both the educational method and the evaluation model itself. Finally, Sect. 6 presents conclusions.

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 507–521, 2017.
DOI: 10.1007/978-3-319-67383-7_37

2 WORK4E Method to Teach Introduction to SPI

The method is part of an innovative process improvement methodology: PRO2PI (Process Modeling Profile to drive Process Improvement) [2]. PRO2PI evolves current model based SPI towards a modeling driven SPI. PRO2PI-WORK method (or simply WORK) is a methodological element of PRO2PI. This method guides a workshop to start a process improvement cycle with the establishment of a Process Modeling Profile to drive improvements [22]. PRO2PI-WORK4E ("for (*four* **4**) **E**ducation") (or simply WORK4E) is a customized version of this method to teach SPI. Both WORK and WORK4E guide the first phases of a SPI cycle and the learning process of SPI. These phases are oriented by business needs, objective and goals, with process assessment, capability profile and improvement plan. While WORK focuses on starting a SPI cycle with learning SPI ("doing SPI with learning SPI"), WORK4E focuses on learning SPI by starting a SPI cycle ("learning SPI by doing SPI").

In a WORK4E course, groups of students identify a known working environment, usually where they work, describe it as an organizational unit and then do a series of activities to start a business oriented process improvement cycle. After that, they write additional material on introduction, motivation, and conclusion to produce a technical article describing these first phases of a SPI cycle.

A previous version of WORK4E method was introduced in another article [1]. The current version has few adjustments. WORK4E is defined with four phases and twenty activities (Fig. 1).

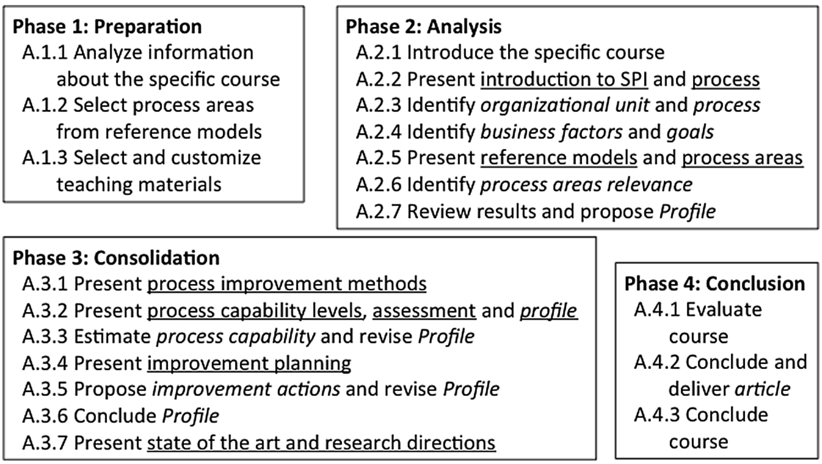

Fig. 1. Phases and activities of WORK4E

First phase of WORK4E guides a preparation to teach a specific course. Phases 2 and 3 guide actual teaching and learning activities. Phase 2 has two groups of activities. One related with introducing SPI and identifying an organizational unit (OU) and business motivation for a SPI cycle (A.2.2 to A.2.4). The second group is related with introducing selected reference models and process areas, identifying which ones are more relevant for a SPI cycle on that OU under that business motivation, and proposing a Process Capability Profile to guide the SPI cycle (A.2.5 to A.2.7). Phase 3 has three groups of activities. One related with introducing process improvement methods and process capability levels, assessment and profile, and estimating capability level of selected process areas in that OU (A.3.1 to A.3.3). The second group is related with introducing improvement planning and proposing improvement actions for the SPI cycle (A.3.4 to A.3.6). Third group is related with presenting state of the art and research directions on SPI, including ISO/IEC 330xx family of standards. During phases 2 and 3 activities, each group of students executes the first phases of the SPI cycle as a means to learn SPI ("SPI learning by doing SPI"). Then they consolidate these results as articles about starting a SPI cycle. Fourth phase guides a conclusion of the specific course.

The subject matter of a course using this method can be explained using the Framework for Describing Process Improvement Subject Matter [20, p. 80–85, 21]. The framework organizes a subject matter into KNOW (Know process and process improvement concepts), DO (Apply concepts, make choices for process and process improvement management and culture change) and USE (Use tools and techniques and pervasive supporting skills). Eight process and process improvement concepts (KNOW) are presented in activities A.2.2, A.2.5, A.3.1, A.3.2 and A.3.4. Applications of these concepts using tools and techniques and pervasive supporting skills (DO and USE) are presented by the instructor and used by the students in activities A.2.3, A.2.4, A.2.7, A.3.3, A.3.5 and A.3.6.

In A.2.2 to A.2.4, for example, together with the presentation of "introduction to process and SPI" (KNOW), there are presentations of how to identify and define "organizational unit", "macro process", "business factors" and "improvement goals", and techniques for "process documentation", "problem identification" and "objective compelling goals" (DO). Next the students apply these techniques (USE).

After previous edition of this course, some adjustments were implemented, including: (a) reduction of slides on concepts; (b) reduction of the number of process areas presented and analyzed in A.2.5 and A.2.6 from 11 to 5; (c) development of an article as a reference example [17]; and (d) increase the number of examples on slides.

3 Evaluation of Educational Games and Methods

In order to identify related work to evaluate WORK4E, first I analyzed the state of art of how to systematically evaluate educational games. WORK4E is an educational method not an educational game. However, it shares some aspects of educational games and I plan to automate its usage and include more educational games characteristics on it. Hence, I decided to investigate educational games evaluation.

In a recent systematic literature review on this subject, Petri and von Wangenheim [6] identified 21,291 articles and selected and analyzed 11 relevant ones. These eleven articles describe seven approaches (models, methods, scales or frameworks) to systematically evaluate educational games.

Petri and von Wangenheim identified that "most of them are frameworks rather than comprehensive evaluation methods, indicating a lack of support on how to conduct such evaluations" and "also seem to be developed in a rather ad-hoc manner, not providing an explicit definition of the objective, measures or data collection instruments" [6, p. 1012]. They identified two approaches that "have been systematically developed by explicitly decomposing evaluation goals into measures and defining a questionnaire, evaluated through series of case studies": MEEGA and EGameFlow. Finally, they concluded: "currently, MEEGA seems to be used more widely in practice being reported by several studies from different authors evaluating different games and contexts, […]. On the other hand, EGameFlow seems to have been applied so far only by the authors of the model themselves".

I already knew MEEGA, a Model for the Evaluation of Educational Games [3–5], and decided to review it to verify whether it could be used to evaluate WORK4E. After a subjective review I concluded that an adapted version of MEEGA, as a new more specific model, could be developed and used. Then I studied MEEGA and its main references, specially Kirkpatrick Model [7], ARCS Model [8], revised Bloom's taxonomy [9], Wohlin's book on experimentation in software engineering [10], GQM goal definition template [12, p. 4], and Jedlitschka and Pfahl's guidelines for reporting controlled experiments in software engineering [11]. I also identified grounded theory techniques [13] to interpret text data with comments on WORK4E.

MEEGA's conceptual model with its main elements, MEEGA's questionnaire and their relationships, are illustrated in Fig. 2, redrew, adapted and translated from [4, p. 128], together with EM4E's conceptual model, which is basically the same of MEEGA, EM4E's questionnaire and its relationship with MEEGA's questionnaire.

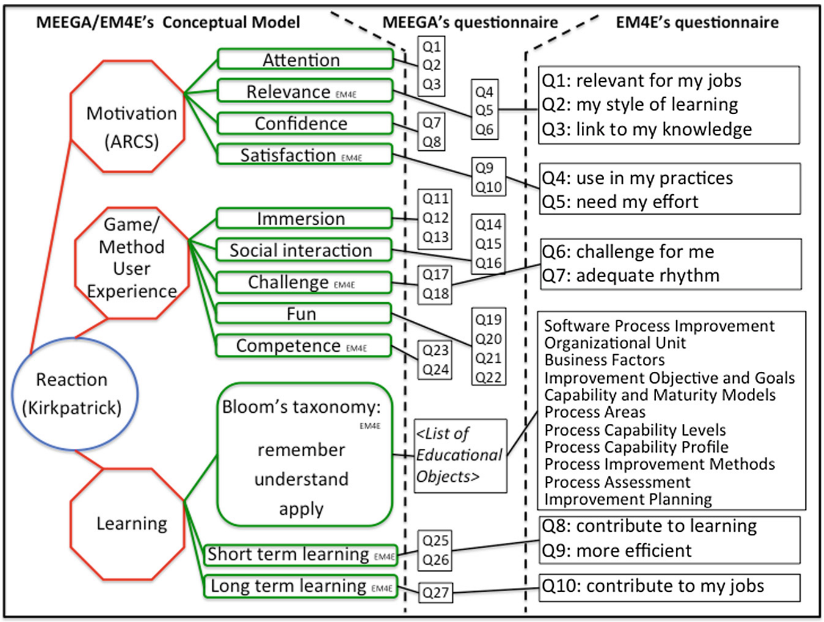

Fig. 2. MEEGA and EM4E's conceptual model and questionnaire

Donald Kirkpatrick developed a training program evaluation model with four sequential levels: Reaction, Learning, Behavior and Results [7]. MEEGA's conceptual model uses only level 1: Reaction. The Reaction level is the measuring of the reaction of the participants in the training program. Reaction is composed of three quality factors: motivation, game user experience and learning. Motivation is Attention, Relevance, Confidence and Satisfaction (ARCS) [8]. Game user experience is Immersion, Social interaction, Challenge, Fun and Competence. Learning is a list of Educational Objects, with the first three levels of revised Bloom's taxonomy: remember, understand and apply [9], and Short and Long term learning.

MEEGA's conceptual model is implemented in a questionnaire with 27 affirmations organized in the eleven dimensions and questions for each educational object. MEEGA is described with four elements: Conceptual model, questionnaire, evaluation process and a spreadsheet [4, p. 140].

4 A Model for the Evaluation of WORK4E Method

The Evaluation Model of WORK4E (EM4E) is based on MEEGA and its main references. The objective of EM4E is to guide evaluations of the effectiveness of WORK4E to potentiate "learning SPI by doing SPI". The objective of each evaluation is defined using the GQM template [12, p. 4]. The general evaluation objective, which should be adapted for each evaluation, and its template are:

Analyze {the name of activity or attribute}	WORK4E method
for the purpose of {overall goal}	Evaluate its effectiveness, as the degree to which it is successful in potentiate "learning SPI by doing SPI"
with respect to {the aspect to be considered}	Motivation, experience using the method and learning
from the viewpoint of {interested people}	Student perception
in the context of {environment}	"Introduction to SPI" course with WORK4E

Based on MEEGA's requirements and research strategy, EM4E's requirements are related with its applicability and usefulness. In terms of applicability, EM4E needs to achieve its objective with a "quick evaluation with minimal time and minimizing the interruption of the instructional unit and be easy to use and not require advanced knowledge of its users in the area of education, measurement or statistics" [3, p. 9]. The general study design is a "one-shot post-test only design" in order to achieve these requirements. In terms of usefulness, the results of an evaluation need to identify strengths and improvement opportunities from WORK4E author viewpoint.

EM4E is composed of eight components (Fig. 3). The **Evaluation objective** component is already described. The **Model design** component is a Technical Report

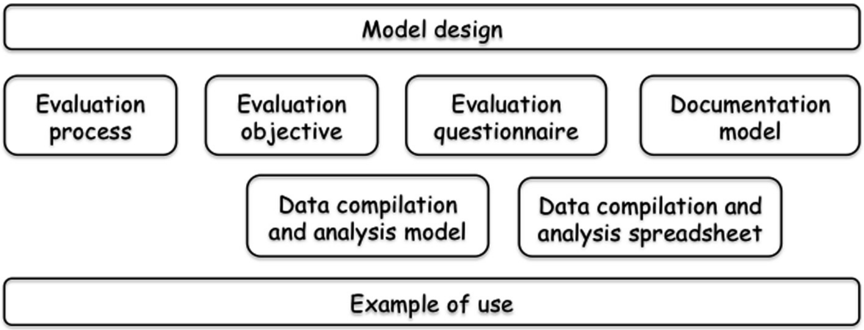

Fig. 3. Components of EM4E model

that presents the model design [14]. This component describes WORK4E method overview, MEEGA model overview, EM4E model's requirements, architecture and conceptual model, four other components (Evaluation process, Evaluation objective, Data compilation and analysis model and Documentation model) and an overview of three other components (Evaluation questionnaire, Data compilation and analysis spreadsheet and Example of use). This section is a summary of that Technical Report.

The description of EM4E design includes how MEEGA was adapted to produce EM4E. This adaptation was derived from MEEGA's questionnaire to develop EM4E's conceptual model and evaluation questionnaire. The **Evaluation questionnaire** component is an instrument to collect data from the students. It has three main elements: affirmations, educational objects and comments.

To develop evaluation questionnaire, first I analyzed each affirmation from MEEGA's questionnaire and selected seven as the most relevant ones for EM4E. These affirmations were Q4, Q5, Q9, Q17, Q25, Q26 and Q27. Then I identified the dimensions of these affirmations: Relevance, Satisfaction, Challenge, Short term learning and Long term learning. After I confirmed that these dimensions were the most relevant to EM4E. Then I included other affirmations to complete the dimensions: Q6, Q10 and Q18. The final set has 10 affirmations that are renumbered from $Q_{EM4E}1$ to $Q_{EM4E}10$. These affirmations were reviewed to better communicate their meanings and to change the word game to method. So EM4E has five dimensions out of original eleven. These EM4E's dimensions address all three quality factors: motivation, game/method user experience and learning. Figure 2 illustrates EM4E's conceptual model, EM4E's questionnaire and its relationships with MEEGA's questionnaire. An English version of EM4E's ten affirmations, their dimensions and correspondent original MEEGA questions are listed in Table 1.

The answer for each affirmation is given in a five Likert-type scale (Fig. 4).

For the second element of evaluation questionnaire, student perception of the learning effect on the three levels of the revised Bloom's taxonomy (remembering, understanding and applying) based on Kirkpatrick's level 1 (reaction) are evaluated using a set of educational objects. The set is composed of eight key concepts of SPI (Introduction to Process and SPI, Capability and Maturity Models, Process Areas, Process Improvement Methods, Process Capability Levels, Process Capability Profile,

Table 1. Affirmations of EM4E's questionnaire

EM4E Id.	EM4E's Affirmation (English version)	Dimension	MEEGA Id.
$Q_{EM4E}1$	The method content is relevant to my interests	Relevance	Q4
$Q_{EM4E}2$	The way the method works suits my way of learning	Relevance	Q5
$Q_{EM4E}3$	The method content is connected to other knowledge I already had	Relevance	Q6
$Q_{EM4E}4$	I am satisfied because I know I will have opportunities to use in practice things I learned using this method	Satisfaction	Q9
$Q_{EM4E}5$	It is due to my personal effort that I manage to advance in the method	Satisfaction	Q10
$Q_{EM4E}6$	This method is appropriately challenging for me, the tasks are neither too easy nor too difficult	Challenge	Q17
$Q_{EM4E}7$	The method progresses at an adequate pace and does not become monotonous - offers new situations or variations in its tasks	Challenge	Q18
$Q_{EM4E}8$	The method contributed to my learning in this course	Short term Learning	Q25
$Q_{EM4E}9$	The method ("learning by doing") was efficient for my learning, comparing it with other "learning by listening" activities	Long term learning	Q26
$Q_{EM4E}10$	The experience with the method will contribute to my professional performance in practice	Long term learning	Q27

1. Affirmations: Please circle a number (-2, -1, 0, +1, or +2) according to how much you disagree or agree with each statement below, considering				
strongly disagree	disagree	neither agree nor disagree	agree	strongly agree
-2	-1	0	+1	+2

Id.	**Affirmation**	**Rating**	**Optional comments**
1	The method content is relevant to my interests.	-2 -1 0 +1 +2	
	(...)		
10	The experience with the method will contribute to my professional performance in practice.	-2 -1 0 +1 +2	

Fig. 4. Excerpt of affirmations element of EM4E's questionnaire

Process Assessment, and Improvement Planning) and three activities to initiate a SPI cycle on how to identify Organizational Unit, Business Factors and Improvement Objective and Goals, as listed in Fig. 2. For each educational object, each student rates his/her perceived level of knowledge before and after using the method on a 5-point interval scale, ranging from 1 (very low) to 5 (very high) [Fig. 5].

2. Learning: Assign a score from 1 to 5 to your level of knowledge of the concepts listed in the table below (1-very low; 5-very high), before and after using the method, in terms of:						
Educational object	**Remembering what it is**		**Understanding how it works**		**Applying it in practice**	
	Before	After	Before	After	Before	After
1. Introduction to Process and SPI						
(…)						
11. Improvement Planning						

Fig. 5. Excerpt of learning element of EM4E's evaluation questionnaire

Third and final evaluation questionnaire element (Comments) asks for general comments organized into strengths, weakness, improvement opportunities and other type of comment (one to three comments for each).

The **Evaluation process** is the same of MEEGA, whose design is based on the five steps of experimentation process by Wohlin et al. [10, pp. 85–160]: (a) scoping, (b) planning, (c) operation, (d) analysis and interpretation, and (e) presentation and package.

The **Data compilation and analysis model** guides how the collected data are compiled and analyzed. This model provides a connection between the evaluation questionnaire and the data compilation and analysis spreadsheet. Data collected from questionnaire's three elements (affirmations, educational objects and comments) are recorded in the spreadsheet. Then affirmations and educational objects data are processed, statistics results are calculated (percentage, average, standard deviation, p-value, etc.) and graphics are generated (frequency, box-plot, etc.). Comments are analyzed in iterative cycles to identify concepts and categories as generic labels using grounded theory style of coding [13]. Then graphics are generated to show number of occurrences of generic labels. The **Data compilation and analysis spreadsheet** is implemented as a Microsoft® Excel® spreadsheet.

The **Documentation model** is based on a scheme for the academic reporting of experiments proposed by Jedlitschka and Pfahl [11] and used in MEEGA [4, pp. 138–140] and Wohlin et al. [10, pp. 153–157]. The document should have five sections: Introduction, Objective and Planning, Execution, Analysis and Interpretation and Conclusions.

Introduction section sets the scope of the work and encourages readers to read the rest of the documentation. Objective and Planning (Experimental Design) section describes the objective and the outcome of the evaluation planning. Execution describes how the plan was implemented. Analysis and Interpretation section summarizes the collected data, describes how it was analyzed and interprets of the findings from analysis. Conclusions section concludes the document, including threats to validity and further work. The **Example of use** is a technical report with an evaluation results documentation example.

5 Results from an Evaluation

In order to validate the evaluation model and produce the Example of use component of EM4E, I performed an evaluation. Following the Documentation model, the results are described in a Technical Report [16]. This section presents an overview of it.

During four consecutive Saturdays, from March 18th till April 8th 2017, I presented an Introduction to SPI course unit using WORK4E. The course unit was taught in Portuguese as part of a postgraduate course in Information Technology (IT) Governance at the Technological Faculty of State University of Campinas (Unicamp) in Limeira campus (http://www.ft.unicamp.br/~espgov).

A set of slides, resulting from tailoring WORK4E's set of slides, was used in class [15]. The course unit had a total of 32 h, eight each Saturday. Thirty-four students participated. All of them have IT related graduation and full time IT related job during the week. An inquiry in the first day of class revealed that they had low level of knowledge about SPI.

5.1 Objective and Planning (Experimental Design)

The main objective of this evaluation, based on the general evaluation objective, was: (Obj.1) *analyze* WORK4E Method *for the purpose of* evaluate its effectiveness as the degree to which it is successful in potentiate "learning SPI by doing SPI" *with respect to* motivation, experience using the method and learning *from the viewpoint of* student's perception *in the context of* that SPI Introduction class. An additional objective was defined for an initial validation of EM4E: (Obj.2) *analyze* EM4E *for the purpose of* evaluate *with respect to* its applicability and usefulness *from the viewpoint of* its author *in the context of* the evaluation of a SPI Introduction class. An extra section was included in the results for this objective.

Data collection for evaluation was planned for the fourth and last day of class. A questionnaire and a spreadsheet were prepared, both adapted from the respective EM4E's questionnaire and spreadsheet. The adaptation was the inclusion of course identification and the evaluation date. The spreadsheet is implemented in Microsoft® Excel® for Mac 2001 Version 14.7.3 (170325) software. The calculations and graphs were performed with this software. 40 hard copies of the questionnaire, with two pages each, were produced for this evaluation.

5.2 Execution

In the fourth day of class, I presented, to all thirty-three students in the class at that time, the evaluation objective and guidelines for its accomplishment, based on the initial text of the questionnaire: "We would like you to answer the questions below, about using the WORK4E method, to help us understand its learning impact and improve the method. All data will be collected anonymously and will only be used for this purpose. Consolidated data, data analysis results and results interpretation may be published in technical articles."

I also communicated that participation in the evaluation was voluntary. All students agreed to participate. A questionnaire's printed copy was handed out to each student.

Each student recorded his/her evaluation in the printed copy. They took from 20 to 30 min to deliver the evaluations. I received thirty-three evaluations and stored with no assignment. The evaluations were then randomly numbered from A01 to A33. The data of each evaluation were then entered in the spreadsheet, identified by the assigned number. I registered total of 2,508 numerical data (76 for each evaluation) and 145 comments.

5.3 Analysis and Interpretation for Obj.1 – Evaluate WORK4E

Analysis and interpretation of collected data are organized regards each element of the questionnaire: affirmations, educational objects and comments.

Table 2. Percentages of affirmations responses by type

Aff./Res.	Relevance			Satisfaction		Challenge		Short and Long term learning		
	1	2	3	4	5	6	7	8	9	10
−2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
−1	6.1%	0.0%	6.1%	9.1%	9.1%	18.2%	0.0%	0.0%	3.0%	3.0%
0	3.0%	0.0%	6.1%	27.3%	27.3%	9.1%	9.1%	6.1%	0.0%	9.1%
1	54.5%	87.9%	60.6%	39.4%	57.6%	48.5%	54.5%	36.4%	0.0%	36.4%
2	36.4%	12.1%	27.3%	24.2%	6.1%	24.2%	36.4%	57.6%	97.0%	51.5%

Affirmations: Table 2 shows the percentage of each type of response (−2, −1, 0, 1, or 2) for each affirmation (from 1 to 10). Statements are organized by their dimensions (Relevance, Satisfaction, Challenge, and Short and Long Term Learning). Table 2 shows 0.0% of −2 (completely disagree) responses, only 5.5% of −1 (disagree) responses, and 9.7% of 0 (neutral) responses. Responses 1 (agree) and 2 (strongly agree) were respectively 47.6% and 37.3%, which represents 84.9% of all responses.

In relation to affirmations of Relevance dimension (affirmations 1, 2 and 3), they had a high percentage of concordances (90.9%, 100.0% and 87.9%). In relation to affirmations of the Satisfaction dimension (affirmations 4 and 5), the percentages of concordances were lower than those of Motivation, although they were still high (63.6% and 63.7%). The neutral percentages, both 27.3%, were considerable, indicating an opportunity for improvement.

In relation to affirmations of Challenge dimension (affirmations 6 and 7), the percentage of agreement of the first affirmation was good (72.7%) and the one of the second one was high (90.9%). The percentage of non-agreement of the first statement, however, was considerable (18.2%), which indicates another opportunity for improvement. In relation to the affirmations of Short and Long term learning dimensions (affirmations 8, 9 and 10), there were a high percentage of concordances (94.0%, 97.0% and 87.9%). Among the results, 89.9% of answers either agree or strongly agree that the method is relevant, provides satisfaction, is challenging and provides short and long term learning.

Educational objects: Table 3 presents an excerpt of the results of the evaluations on educational objects with one educational object (Edu.Obj.), aspects (Asp.) "Remember what is" (Rem.), "Understand how it works" (Und.) and "Apply in practice" (Appl.), before and after (B/A) the use of the method, in the scale of 1, 2, 3, 4 and 5, being 1 – very low and 5 – very high. It also presents the standard deviations (StdDev), the p-value between before and after values, and the differences between after and before for each object. The complete table is in [16].

Table 3. Excerpt of educational objects' evaluation

Edu.Obj.	Asp.	B/A	Avg.	StdDev	p-value	Dif.
(1) Software process improvement	Rem.	Before	1.85	0.107	0.000000000000	2.15
		After	4.00	0.000		
	Und.	Before	1.45	0.321	0.000000000000	2.48
		After	3.94	0.043		
	Appl.	Before	1.24	0.171	0.000000000001	1.94
		After	3.18	0.129		
(…)						

In relation to SPI educational object, from Table 3, there is an average increasing of 2.15 (from 1.85 before and 4.00 after the course) in terms of remember what SPI is. There is an average increasing of 2.48 (from 1.45 before and 3.94 after the course) in terms of understand how SPI works. There is an average increasing of 1.94 (from 1.24 before and 3.18 after the course) in terms of apply in practice.

In relation to aggregate eleven educational objects, from the complete table, there is an average increasing of 1.99 (from 1.89 before and 3.88 after the course) in terms of remember what each educational object is. There is an average increasing of 2.01 (from 1.86 before and 3.87 after the course) in terms of understand how each object learning works. There is an average increasing of 1.76 (from 1.71 before and 3.47 after the course) in terms of apply each object learning in practice.

Comments: There are 145 comments, numbered from 1 to 145. In all, six comments were written in the "Others" section: identified as 13, 52, 100, 138, 139 and 145. These comments were analyzed and reclassified in Strength (100, 138 and 139), Weakness (145) and Improvement (13 and 52). Figure 6 shows analyses of comments.

Figure 6-a shows the distribution of comments into Strength (84 comments), Weakness (34) and Improvement Opportunity (27). Although the questionnaire asks for comments on the method, some of them was not actually on it. After analysis, I identified comments on the style of teaching, on SPI itself, and others. Figure 6-b shows the distribution of comments in terms of their object: method (104 comments: 63 strengths and 41 weaknesses/improvements), teaching (18), SPI (11) and others (12). Then I analyzed each comment on method object, in iterative cycles to identify similar comments and assign generic labels representing them, using grounded theory style of coding. After some cycles, 33 generic labels were identified: 13 for strength and 20 for weakness and improvement. Then the number of original comments for each generic

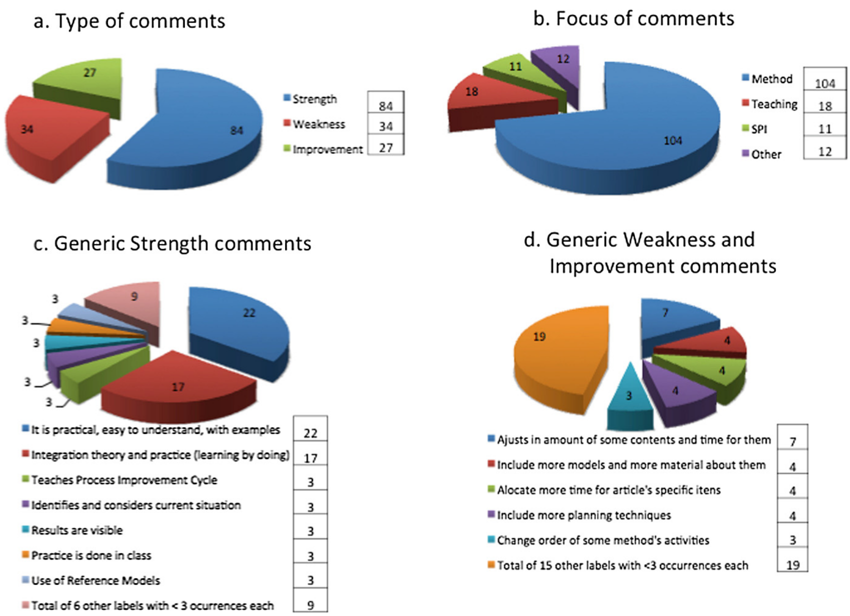

Fig. 6. Analyses of comments

label was counted. Figure 6-c shows the distribution of 7 strength generic labels with three or more occurrences and Fig. 6-d shows the distribution of 5 weakness and improvement opportunity generic labels with three or more occurrences.

Two generic labels stand out as strengths: the method is practical, easy to use (23 occurrences) and potentiate learning by doing (19 occurrences). Among weaknesses and improvement opportunities one generic label stand out: the method needs adjustments in amount of some contents and time allocated for them (7 occurrences).

5.4 Analysis and Interpretation for Obj.2 – Evaluate EM4E

The data collection for evaluation took about 50 min including explanations and answers in class. This can be considered "minimal time and minimizing the interruption of the instructional unit" in a 32 h course unit. It was "easy to use and did not require advanced knowledge of its users in the area of education, measurement or statistics" given that the students answered without difficult the questionnaire. Therefore the model attends applicability aspect of Obj.2.

To evaluate usefulness aspect of Obj.2, analyses and interpretations of Obj.1 were revised. Analyses and interpretations of affirmations, educational objects and comments identified relevant strengths of WORK4E method. Analyses and interpretations of affirmations and comments identified relevant improvement opportunities for WORK4E. Therefore the model attends usefulness aspect of Obj.2 from the viewpoint of its author.

6 Conclusion

Analyses and interpretations of the collected data provided initial validation of EM4E model. The results from this evaluation of WORK4E method using EM4E model, indicate that the method is effective in potentiate “learning SPI by doing SPI”. There are, however, four main threats to validity. First, just one person developed the method and the model, taught the class, and executed the evaluation. Therefore, there are possible biases. To handle this threat, the evaluation model, process, data, analyses and interpretations are described in details, in a systematic and traceable manner, elsewhere [14, 16, 23] and overviewed in this article. Second threat is that the evaluation is just of one edition of the course. Therefore these results should not be generalized. Third, it is the first usage of a new evaluation model. To handle this threat, the model was developed in a structured and traceable manner from a stable more used model (MEEGA). Actually this evaluation also could be considered as a customized usage of MEEGA. Finally, the fourth threat to validity is that both the model and consequentially this evaluation are just on Kirkpatrick’s first level (Reaction). The next level (Learning) should be included in EM4E model.

In addition, at the end of this course unit, two other formal evaluations of student learning level were performed: a final exam and an article. The final exam had two parts. One part had eighteen direct multiple-choice questions, covering all eleven educational objects, following revised Bloom’s “remember what is” and “understand how it works”. Another part had an open question asking for an estimation and justification of the capability level of a process area given a description of a scenario. This part follows revised Bloom’s “apply in practice” on three educational objects: process area, capability levels and assessment. Regards first part, there were an average of 89% right answers, medium 91%, standard deviation 0.09, minimum 65% and maximum 100%. Regards second part, there were an average of 73% right answers, medium 86%, standard deviation 0.28, minimum 0% and maximum 100%. Other student evaluation was analyzing articles from eight groups. Following WORK4E, each article described the first phases of a SPI cycle. After analyses, there were an average of 83% satisfaction, medium 83%, standard deviation 0.07, minimum 72% and maximum 92%. These results are consistent with EM4E evaluation results and reinforce those results, although with similar threats to validity.

Further work is organized into WORK4E method, EM4E model, its usage and a model for WORK. WORK4E will be improved considering improvement opportunities from this evaluation, automation of its usage and inclusion of more educational games aspects on it. EM4E should also be improved considering a new recent version of MEEGA model [18], recent proposals for more robust statistical methods for empirical software engineering [19], and the next level (Learning) of Kirkpatrick’s model. In addition, further evaluation of EM2E should be performed in terms of validity and reliability of the scale used for data collection. EM2E should be used in more evaluations of courses. Finally, an evaluation model for WORK should be developed using EM2E experience as reference.

Acknowledgments. The author thanks the students of the mentioned course for participating in the SPI learning process and for answering the evaluation questionnaire, the authors of MEEGA model for providing reference to EM4E model, and the reviewers of SPICE 2017 for providing useful comments.

References

1. Salviano, C.F.: Teaching process improvement by establishing process modeling profile to drive process improvement – the PRO2PI-WORK4E method. In: [20], pp. 63–69 (2015)
2. Salviano, C.F.: Model-driven process capability engineering for knowledge working intensive organization. In: SPICE 2008, Nuremberg, Germany, pp. 1–9 (2008)
3. Savi, R., von Wangenheim, C.G., Borgatto, A.F., Buglione, L., Ulbricht, V.R.: MEEGA – a model for the evaluation of games for teaching software engineering. INCoD Technical report, 001/2012 (2012)
4. Savi, R.: Evaluation of games for knowledge dissemination (in Portuguese "Avaliação de Jogos Voltados para a Disseminação do Conhecimento"). Ph.D. Thesis in Engenharia e Gestão do Conhecimento at Univ. Fed. Santa Catarina, UFSC, Florianópolis, 236 p. (2011)
5. Savi, R., von Wangenheim, C.G., Borgatto, A.F.: A model for the evaluation of educational games for teaching software engineering (in Portuguese "Um Modelo de Avaliação de Jogos Educacionais na Engenharia de Software"). In: Proceedings of the 25th Brazilian Symposium on Software Engineering, pp. 194–203 (2011)
6. Petri, G., von Wangenheim, C.G.: How to evaluate educational games: a systematic literature review. J. Univ. Comput. Sci. **22**(7), 992–1021 (2016)
7. Kirkpatrick, D.L., Kirkpatrick, J.D.: Evaluating Training Programs. Berrett-Koehler Publishers, Oakland (1994)
8. Keller, J.M.: Development and use of the ARCS model of instructional design. J. Instr. Develop. **10**(2) (1987). doi:10.1007/BF02905780
9. Krathwohl, D.R.: A revision of Bloom's taxonomy: an overview. Theory Pract. **41**(4) (2002)
10. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012). 236 p., WORK4E
11. Jedlitschka, A., Pfahl, D.: Reporting guidelines for controlled experiments in software engineering. In: Proceedings of the 4th International Symposium on Empirical Software Engineering, Noosa Heads, pp. 95–104 (2005)
12. Basili, V.R.: Software Modeling and Measurement: The Goal/Question/Metric Paradigm. Univ. of Maryland, College Park, MD, USA, Technical report CS-TR-2956, 24 p. (1992)
13. Charmaz, K.: Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis, vol. 10. SAGE Publications Inc., London (2006)
14. Salviano, C.F.: EM4E - Evaluation Method of PRO2PI-WORK4E Method (in Portuguese "EM4E - Modelo de Avaliação do Método PRO2PI-WORK4E"). Technical report TRT0040117, CTI Renato Archer/NMPS Research Group, version 1.1 (2017)
15. Salviano, C.F.: Slides on Introduction to Software Process Improvement with PRO2PI-WORK4E 2017 edition (original in Portuguese, "Slides sobre Introdução à Melhoria de Processo de Software com PRO2PI-WORK4E edição 2017"). Technical report TRT0043117, CTI Renato Archer/ NMPS Research Group, version 1.1, 268 p. (2017)
16. Salviano, C.F.: Results of PRO2PI-WORK4E Method Evaluation in CET-811 course 2017 edition (original in Portuguese "Resultado da Avaliação do Método PRO2PI-WORK4E na Disciplina CET-811 edição 2017"). Technical report TRT0041117, CTI Renato Archer/ NMPS Research Group, version 1.1 (2017)

17. Salviano, C.F.: Starting Software Process Improvement with PRO2PI-WORK4E in an Innovative Product VSE (original in Portuguese “Início da Melhoria de Processo de Software com PRO2PI-WORK4E em MPE de Produto Inovador”). Technical report TRT0042117, CTI Renato Archer/ NMPS Research Group, version 1.1, 19 p. (2017)
18. Petri, G., von Wangenheim, C.G, Borgatto, A.F.: MEEGA+: An Evolution of a Model for the Evaluation of Educational Games. Technical report INCoD/GQS.03.2016.E, Version 1.0, ISSN 2236-5281 (2016)
19. Kitchenham, B., Madeyski, L., Budgen, D., Keung, J., Brereton, J., Charters, S., Gibbs, S., Pohthong, A.: Robust statistical methods for empirical software engineering. Empir. Softw. Eng. **22**, 579–630 (2017). doi:10.1007/s10664-016-9437-5
20. O’Connor, R.V., Mitasiunas, A., Ross, M. (eds.) In: Proceedings of 1st International Workshop on Software Process Education, Training and Professionalism. Gothenburg, Sweden (2015). http://ceur-ws.org
21. Ibrahim, R.L., Hirmanpour, I.: The Subject Matter of Process Improvement: A Topic and Reference Source for Software Engineering Educators and Trainers. Technical report CMU/SEI-95-TR-003 ESC-TR-95-003 (1995). http://www.sei.cmu.edu/reports/95tr003.pdf
22. Salviano, C.F.: Establishing ISO/IEC 15504-Based Process Capability Profile to Process Improvement. In: Presented at 16th EuroSPI Conference on Tutorial Slides, Spain (2009)
23. Salviano, C.F.: PRO2PI-WORK4E project (2017). https://www.researchgate.net/project/PRO2PI-WORK4E-Method-for-teaching-software-process-improvement

Towards a Strategy for Process Improvement Education and Training

Linda Ibrahim[1] and Antanas Mitasiunas[2(✉)]

[1] Enterprise SPICE, Washington, DC, USA
rlibrahim@aol.com
[2] Vilnius University, Universiteto Street 24, 01513 Vilnius, Lithuania
antanas.mitasiunas@mif.vu.lt

Abstract. At the 2015 Software Process Education, Training and Professionalism workshop, the focus was on bringing together a Manifesto to include values and principles. However, without a strategy driven by a commonly shared vision, it may be difficult to seek more tangible outcomes. This paper drafts a proposed strategy that might help us move forward.

Keywords: Software process improvement · Education and training · Professionalism · Strategy for process improvement education and training

1 Introduction

This paper examines issues, challenges and opportunities regarding process improvement education and training. It recommends bringing together all stakeholders to develop a strategy to build on strengths and opportunities, and diminish the weaknesses and threats that exist in our internal and external environments. It is hoped these thoughts can help us collaboratively develop a strategy in pursuit of a common vision regarding process education, training and professionalism.

2 Background

Several papers were presented at the 2015 IWSPETP workshop covering experiences and recommendations regarding academic education and professional training [SPE15]. Analysis of these papers and the subsequent workshop discussions started to set a course for improving education, training and professionalism. As a first step, participants worked towards developing a Manifesto to include values and principles.

Values were defined as representing the core priorities in an education culture, including what drives priorities and how you truly act when doing education.

Principles were defined as basic generalizations that are accepted as true and that can be used as a basis for education reasoning or education behavior.

Workshop participants were asked to elaborate on the Values and Principles that were derived during the workshop, for eventual publication of the Manifesto. Initial Manifesto, consisting of 10 values and 4 principles, was presented in 16th International Conference SPICE 2016 [Joh16]. But, although values and principles are critically

A. Mas et al. (Eds.): SPICE 2017, CCIS 770, pp. 522–528, 2017.
DOI: 10.1007/978-3-319-67383-7_38

important topics, it remains paramount to describe a path forward. This could be contained in a strategy.

3 What Might a Strategy Contain?

There are many variations and formats for a strategy, and many proposed approaches for derivation. See for example [Qui16a] or [Dix02]. We suggest here the following simple way. Develop a vision, and a mission, examine the internal and external environment to derive goals, and attach initiatives to each goal with timelines and responsibilities for accomplishment.

4 Vision

Where do we want to be? What do we want things to be like? Here is a draft vision as proposed at SWEPT 2015: “The process improvement profession is recognized, valued, and effective in helping organizations improve their performance”.

5 Mission

What do we need to do to achieve the vision? To achieve the vision we need to transform process improvement into a core asset of industry and organizations.

6 The Current Environment

A common technique used in developing a strategy is to examine internal and external factors in the current environment, so as to try to find ways to leverage this information. This might be done via a SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis. For further information on SWOT and its use in strategy development see e.g. [Ber16, Qui16b, Tay16]. Below are some SWOT observations on our current situation, as extracted from [Ibr15] and embellished based on SPETP 2015 paper reviews and workshop discussion. For further validation, the authors reviewed the SPETP 2015 papers and those addressing one or more environmental issue include [Ros15, Here15, Lap15, Sch15, Herp15, Mir15, Sal15, Por15, Mcq15]. Those factors are included/consolidated into this SWOT analysis.

6.1 Strengths

What are our strengths? (*We need to maintain, build on and leverage these.)*

- Extensive community of people working in process improvement education and training and professional certification – including Universities, Colleges, Professional Societies, Institutes

- Courses offered using various delivery mechanisms e.g. on-line, instructor-led, in-house, off-site, options for self-study
- Process improvement knowledge captured via various initiatives
- Enormous experience and practical knowledge have been accumulated regarding what needs to be done for success in PI
- Information on various teaching methods is available
- Work is available describing learning models.

6.2 Weaknesses

What are our weaknesses? *(We need to remedy, change, stop and overcome these.)*

- Dwindling, sporadic interest in process improvement in industry and government
- Confusion in terminology regarding training and certification offerings e.g. Business Process Management (BPM), Quality Training, BPI (I = Innovation or Improvement), black belt, 6-sigma, Lean 6-sigma, ITIL, Business Process Re-engineering, TQM, etc.
- Confusion regarding which process improvement approach might help the most e.g., Model-based, SPICE, CMMI, Six-sigma, Lean, IDEAL, black belt, Lean Six-Sigma, ITIL – and who is best qualified to offer this training
- Insufficient attention to process in university courses to ground the fundamentals
- Inadequate training and qualification of PI professionals
- Lack of standardization regarding process education and training content – similar topics, overlap, inconsistency, various bodies of knowledge – lack of accepted PI standards
- Poor understanding of competencies, roles and responsibilities of PI activities
- Stove-piped professional courses not recognizing the needs to integrate PI approaches, or the value obtained from various approaches
- Training and education too expensive, too time-consuming
- Cost of standards too high
- University/college lecturers lack PI experience and training
- Limited number of adequately qualified PI professionals – shortage of suitably qualified PI professionals
- Graduates have low level of key qualifications especially independent thinking, teaming, communication skills, problem solving, awareness of real PI challenges
- Industry dissatisfied regarding level of preparedness of students entering job market – students do not have skills needed
- Students don't know target state of knowledge
- Universities do not address learning to learn.

6.3 Opportunities

What are our opportunities? *(We need to prioritize, capture, build on and optimize these.)*

- Clarification and standardization of subject matter, body of knowledge for process improvement

- Internationally recognized common content and authorization for professional certifications
- Curriculum guidelines for process improvement education and training
- Undergraduate capstone projects as well as graduate projects in industry
- Collection and publication of data on availability and effectiveness of education, training and professional programs
- Research on delivery modes and their effectiveness
- Reduce training costs, offer distance learning, Massive Open Online Course (MOOC), use new technologies, webinars, flexible learning paths
- Work to ensure executives and decision-makers understand the value of process improvement to address dwindling interest in process improvement
- Provide managers broader views on PI issues such as culture and change
- Provide certification to practitioners
- Bridge the gap between education and training
- Dialog and collaboration between universities and industry
- Provide education, training and guidance to our customers based on the accumulating codified wealth of process knowledge and information available
- Government funding for lecturers' PI work
- Influence government, professional bodies and industry to endorse PI education and training
- Competitions, awards and contests related to PI
- Gamification strategies and tools for PI education and training
- Using problem solving methods, hands-on training, teamwork practices, learning from each other, problem based learning to motivate students
- Teach ability to learn and learning to learn so as to enable lifelong learning
- Teachers should use a PI process to teach PI
- Teachers need to create the possibilities for the production or construction of PI knowledge
- Industry needs college graduates who understand processes and how improve them
- Need more partnerships between universities and industry
- Need PI courses beneficial to both faculty and industry.

6.4 Threats

What are our threats? *(We need to counter, minimize or manage these.)*

- Lack of buy-in from customers regarding the need for process improvement and hence education and training
- PI perceived as low priority by most small entities/enterprises
- Lack of cooperation and buy-in from education and training institutions to work together to improve the quality and available of process improvement education and training
- Dwindling, sporadic interest in process improvement in industry and government
- Competing training organizations
- Know-how becomes outdated soon
- Teacher's role may need to change

- Learning to learn is not widely recognized in University environments and it is a critical ability for graduates
- Teaching PI is not a trivial activity
- Much of what is taught currently is self-taught.

7 Goals and Initiatives

We propose 4 goals that can help realize the vision and that address some of the factors in our environment as described above. Each goal includes briefly described initiatives that might be pursued to meet that goal. Time frames and responsibilities for each initiative could be discussed at the 2017 event.

Goal 1: Establishment: of the stakeholders and participants for this work:

- Governance body – establish a governance/oversight body to track progress and provide direction
- Participants – engage/enlist people who agree to participate in various roles, to include experts on process technical issues and managers and practitioners in the field
- Support and involvement – engage stakeholders to form a collaborative network of those who agree to work together, endorse outcomes and support achievement of the vision.

Goal 2: Fundamentals: development of the bases for providing process improvement education and training:

- Foundations – identify the theoretical and practical knowledge needed as a foundation for process improvement
- Body of knowledge – develop a standardized body of knowledge that consolidates, integrates and structures the great wealth of process improvement knowledge available in various documents and education and training venues
- Analysis – collect and publish data on the availability and effectiveness of process improvement education, training and professional programs
- Curricula – develop curricula for the various audiences targeted for process improvement training and education
- Learning models – develop and endorse models that can be used to understand and effect learning
- Culture – understand why interest in process improvement is dwindling and how to transform process improvement into a core asset of industry and government. Study what motivates organizations and students in the process improvement arena.

Goal 3: Innovation:

- Gamification – explore methods to gamify process improvement education and training and measure effectiveness
- Tools and techniques – pilot new methods, tools, and techniques for delivery and measure effectiveness

- Collaboration – actively engage stakeholders in new ways to provide education and training.

Goal 4: Delivery/Recognition: Examination of various delivery mechanisms and deployment paths:

- Delivery mechanisms – explore, implement and measure the value and effectiveness of various teaching approaches and delivery mechanisms
- Accreditation and certification – work on mechanisms for the accreditation of organizations, trainers, and recipients of standardized process improvement education and training.
- Recognition – provide suitable recognition for those engaged in the process improvement profession.

8 Conclusions and Recommendations

This paper recommends working together to develop a strategy for improving process improvement education and training. It provides some views on our current environment, and proposes a strategy aimed at achieving our vision.

References

[SPE15] 1st International Workshop on Software Process Education, Training and Professionalism (SPETP), Gothenburg, Sweden, 15 June (2015)

[Joh16] Johansen, J., Colomo-Palacios, R., O'Connor, Rory V.: Towards a manifesto for software process education, training and professionalism. In: Clarke, Paul M., O'Connor, Rory V., Rout, T., Dorling, A. (eds.) SPICE 2016. CCIS, vol. 609, pp. 98–105. Springer, Cham (2016). doi:10.1007/978-3-319-38980-6_8

[Qui16a] Quick MBA, The Strategic Planning Process, www.quickmba.com

[Dix02] Dix, J., Matthews, H.: The Process of Strategic Planning, Fisher College of Business. Ohio State University (2002), http://fisher.osu.edu/supplements/10/1470/All_Articles.pdf

[Ber16] Berry, T.: What is a SWOT Analysis? Bplans (2016), http://articles.bplans.com/how-to-perform-swot-analysis

[Qui16b] Quick MBA, SWOT Analysis, www.quickmba.com

[Tay16] Taylor, N.: SWOT Analysis: What It Is and When to Use It, Business News Daily (2016)

[Ibr15] Ibrahim, L.: Process Education Training and Professionalism – Let's Bring Together Process Improvement Knowledge. In: SPETP 2015, June 2015

[Ros15] Ross, M.: Process Improvement – Barriers and Opportunities for Teaching and Training. In: SPETP 2015, June 2015

[Here15] Heredia, A., Colomo-Palacios, R., Amescua-Seco, A.: A systematic Mapping Study on Software Process Education. In: SPETP 2015, June 2015

[Lap15] Laporte, C., O'Connor, R.: Software process improvement in graduate software engineering programs. In: SPETP 2015, June 2015

[Sch15] Schweigert, T., Biro, M.: The SPI Manifesto and the corresponding ECQA certified SPI manager Training. In: SPETP 2015, June 2015

[Herp15] Herpers, M.: The teacher's role in gamification in software engineering at universities (field report) – or how geeks can be inspired to sing. In: SPETP 2015, June 2015

[Mir15] Mirzianov, O., Mitasiunas, A.: Continuous learning process assessment model. In: SPETP 2015, June 2015

[Sal15] Salviano, C.: Teaching process improvement by establishing process modeling profile to drive process improvement – The PRO2PI-WORK4E method. In: SPETP 2015, June 2015

[Por15] Portela, C., Alexandre, M., Vasconcelos, L., Oliveira, S.: Software process education oriented to software industry needs. In: SPETP 2015, June 2015

[Mcq15] McQuaid, P., Banerjee, R.: Establishing long-lasting relationships between industry and academia. In: SPETP 2015, June 2015

Author Index