Fergal McCaffery
Rory V. O'Connor
Richard Messnarz (Eds.)

Communications in Computer and Information Science 364

# Systems, Software and Services Process Improvement

20th European Conference, EuroSPI 2013
Dundalk, Ireland, June 2013
Proceedings

Springer

# Communications in Computer and Information Science 364

Fergal McCaffery Rory V. O'Connor
Richard Messnarz (Eds.)

# Systems, Software and Services Process Improvement

20th European Conference, EuroSPI 2013
Dundalk, Ireland, June 25-27, 2013
Proceedings

Springer

Volume Editors

Fergal McCaffery
Dundalk Institute of Technology, Ireland
E-mail: fergal.mccaffery@dkit.ie

Rory V. O'Connor
School of Computing
Dublin City University, Ireland
Email: roconnor@computing.dcu.ie

Richard Messnarz
ISCN, GesmbH
Graz, Austria
E-mail: rmess@iscn.com

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-39178-1 e-ISBN 978-3-642-39179-8
DOI 10.1007/978-3-642-39179-8
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013941144

CR Subject Classification (1998): D.2.9, K.6.3, D.2, K.6, J.1, H.3.5, H.4



*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

This textbook comprises the proceedings of the 20th EuroSPI Conference, held during June 25–27, 2013, in Dundalk, Ireland.

Since EuroSPI 2010, we have extended the scope of the conference from software process improvement to systems, software and service-based process improvement. EMIRAcle is the institution for research in manufacturing and innovation, which came out as a result of the largest network of excellence for innovation in manufacturing in Europe. EMIRAcle key representatives joined the EuroSPI community, and papers as well as case studies for process improvement on systems and product level will be included in future.

Since 2008, EuroSPI partners have packaged SPI knowledge in job role training and established a European certification association (www.ecqa.org) to transport this knowledge across Europe using standardized certification and examination processes.

Conferences were held in Dublin (Ireland) in 1994, in Vienna (Austria) in 1995, in Budapest (Hungary) in 1997, in Gothenburg (Sweden) in 1998, in Pori (Finland) in 1999, in Copenhagen (Denmark) in 2000, in Limerick (Ireland) in 2001, in Nuremberg (Germany) in 2002, in Graz (Austria) in 2003, in Trondheim (Norway) in 2004, in Budapest (Hungary) in 2005, in Joensuu (Finland) in 2006, in Potsdam (Germany) in 2007, in Dublin (Ireland) in 2008, in Alcala (Spain) in 2009, in Grenoble (France) in 2010, in Roskilde (Denmark) in 2011, and in Vienna (Austria) in 2012.

EuroSPI is an initiative with the following major action lines http://www.eurospi.net:

- Establishing an annual EuroSPI conference supported by software process improvement networks from different EU countries
- Establishing an Internet-based knowledge library, newsletters, and a set of proceedings and recommended books
- Establishing an effective team of national representatives (from each EU-country) growing step by step into more countries of Europe
- Establishing a European Qualification Framework for a pool of professions related with SPI and management. This is supported by European certificates and examination systems

EuroSPI has established a newsletter series (newsletter.eurospi.net), the SPI Manifesto (SPI = Systems, Software and Services Process Improvement), an experience library (library.eurospi.net) that is continuously extended over the years and is made available to all attendees, and a Europe-wide certification for qualifications in the SPI area (www.ecqa.org, European Certification and Qualification Association).

A typical characterization of EuroSPI is reflected in a statement made by a company: "... the biggest value of EuroSPI lies in its function as a European knowledge and experience exchange mechanism for SPI and innovation."

Since its beginning in 1994 in Dublin, the EuroSPI initiative has outlined that there is not a single silver bullet with which to solve SPI issues, but that you need to understand a combination of different SPI methods and approaches to achieve concrete benefits. Therefore, each proceedings volume covers a variety of different topics, and at the conference we discuss potential synergies and the combined use of such methods and approaches. These proceedings contain selected research papers under seven headings:

- Section I: SPI Safety and Regulation Issues
- Section II: SPI Lifecycle and Models
- Section III: SPI Quality and Testing Issues
- Section IV: SPI Networks and Teams
- Section V: SPI and Reference Models
- Section VI: SPI and Implementation
- Section VII: Selected Key Notes and Workshop Papers

Section I presents three papers related to "SPI Safety and Regulation Issues." Nevalainen et al. discuss formal assessment techniques applied to safety-critical systems. Mayer et al. highlight issues of risk management processes in the context of telecommunications regulation, and Flood et al. approach these issues from a medical device perspective dealing with the ISO 62366 standard.

Section II presents three papers under the umbrella topic of "SPI Lifecycle and Models." Firstly, Monasor et al. describe a feasibility study simulating global software development processes for use in education. Winkler et al. identify risks, challenges, and candidate solutions to identify how to bridge the gap from research to industry. The final paper of this section by Krishnamurthy and O'Connor presents an analysis of the software development processes of open source e-learning systems.

Section III presents papers related to "SPI Quality and Testing Issues." In the first paper, Toroi et al. present the first official version of SAWO, a functional defect classification scheme developed to enable the usage of defect data for SPI purposes. Petrova-Antonova et al. propose an approach based on a fault injection technique for generation and execution of fault tolerance test cases, which is automated through the implementation of two software tools for fault injection and test case generation and execution. In the final paper, Gabriel Alberto García-Mireles et al. describe a framework to support software quality trade-offs from a process-based perspective.

Section IV explores the theme of "SPI Networks and Teams." In the first paper Jermakovics explains that collaboration is important to software development processes and collaboration networks help us understand its structure and patterns. Ponisio et al. present an approach that uses techniques from network analysis to support organizations in processing and understanding this information. In the third paper, Petri Kettunen discusses the many facets of high-performing software teams and takes a capability-based analysis approach to investigating teams.

Section V presents three contributions dealing with associated issues surrounding the topic of “SPI and Reference Models.” In the first paper, Jeners and Lichter take an automated comparison approach to the smart integration of process improvement reference models. In the second paper, Fricker et al. examine how an existing reference model can be tailored to a domain it has not been designed for initially, in this instance the healthcare sector. In a second paper from Jeners et al., the authors describe mapping in the complex world of software processes with the context of software development projects and discuss their initial mapping efforts.

Section VI discusses issues in “SPI and Implementation.” In the first paper, Sussy Bayona et al. review the critical success factors related to people in software process improvement. In the second paper, de Souza Cavalcanti Rocha et al. present a proposal for the improvement predictability of cost using earned value management and quality data. In the final paper of this section, Munoz et al. discuss the involvement of stakeholders in software processes improvement to reduce change resistance.

Section VII presents selected keynotes from EuroSPI workshops concerning the future of SPI. From 2010 onwards, EuroSPI invites recognized key researchers to present work on new future directions of SPI. These key messages are discussed in interactive workshops and help create SPI communities based on new topics.

Five invited papers concerning “Agile Development Paradigms” discuss experiences with the adoption of agile development paradigms in software engineering and in product developments for the market, as well as how SPICE-based assessment methods need to be tailored to accept agile approaches in capability assessments. Schweigert et al. discuss the needs for an agile maturity model and analyze how maturity models would really measure agility. The paper by Schweigert, Ekssir-Monfared, and Ofner describes an approach to forming an agile management process and uses the example of a Test SPICE implementation to outline how this would work. Papatheocharous and Andreou describe how agile approaches have been adopted in organizations based on an empirical analysis. Laanti et al. discuss the different interpretations of agile approaches, since teams who implemented agile approaches in projects have placed emphasis on different key issues in the past. In the next paper, de Amescua et al. outline how agile software developments are adopted in application areas where the products are used in a large market.

Three invited papers concerning “Creating Environments Supporting Innovation and Improvement” illustrate that SPI is inherently linked to innovation and that innovation requires a transfer of ideas to an exploitation, a strategy for valorization of new ideas and products or services, and an understanding of a networking on a multicultural scale. Sheriff et al. analyze the relationship between innovation, value creation, and the sustainability of values created, and they view the understanding of this relationship as a driver for innovation. Marek Gavenda et al. outline that the sustainability of an innovation is inherently linked with entrepreneurship and describe a set of competencies needed to achieve this. Finally, Georgiadou and Siakas propose defining valorization as a process and implementing an innovation- and valorization-specific maturity model for continuous improvement of the valorization process.

One invited paper concerning "SPI and Measurement" by Thomas Fehlmann and Eberhard Kranich illustrates that Six Sigma is not just a tool for production capability but that it can also be applied for software development using the mobile phone application development as an example.

Four invited papers on "Risk Management and Functional Safety" illustrate experiences from the medical device and automotive industry in the implementation of recent risk management and functional safety standards. Finnegan, McCaffery, and Coleman describe an assurance and assessment framework for networked medical devices integrating the concepts from ISO/IEC 15504 with medical device standards. Messnarz et al. describe experiences with the implementation of functional safety standards in the automotive industry and what level of know-how is needed by functional safety managers and functional safety engineers to effectively implement risk management and functional safety. Kreiner et al. describe a new initiative of automotive clusters in Europe with an integrated view of product development and process quality based on ISO/IEC 15504 (Automotive SPICE), Lean Six Sigma, and functional safety standards. Finally, Botond Tényi et al. describe experiences in the implementation of risk management in a leading medical device engineering company.

June 2013

Fergal McCaffery
Rory V. O'Connor
Richard Messnarz

# Recommended Further Reading

In [1] the proceedings of three EuroSPI conferences were integrated into one book, which was edited by 30 experts in Europe. The proceedings of EuroSPI 2005 to 2011 have been published by Springer in [2], [3], [4], [5], [6] [7] [8] and [9], respectively.

## References


1. Messnarz, R., Tully, C. (eds.): Better Software Practice for Business Benefit – Principles and Experience, 409 pages. IEEE Computer Society Press, Los Alamitos (1999)
2. Richardson, I., Abrahamsson, P., Messnarz, R. (eds.): EuroSPI 2005. LNCS, vol. 3792. Springer, Heidelberg (2005)
3. Richardson, I., Runeson, P., Messnarz, R. (eds.): EuroSPI 2006. LNCS, vol. 4257. Springer, Heidelberg (2006)
4. Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.): EuroSPI 2007. LNCS, vol. 4764. Springer, Heidelberg (2007)
5. O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.): EuroSPI 2008. CCIS, vol. 16. Springer, Heidelberg (2008)
6. O'Connor, R.V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.): EuroSPI 2009. CCIS, vol. 42. Springer, Heidelberg (2009)
7. Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.): EuroSPI 2010. CCIS, vol. 99. Springer, Heidelberg (2010)
8. O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.): EuroSPI 2011. CCIS, vol. 172. Springer, Heidelberg (2011)
9. Winkler, D., O'Connor, R.V., Messnarz, R. (eds.): EuroSPI 2012. CCIS, vol. 301. Springer, Heidelberg (2012)

# Organization

## Board Members

EuroSPI Board Members represent centers or networks of SPI excellence having extensive experience with SPI. The board members collaborate with different European SPINS (Software Process Improvement Networks). The following six organizations have been members of the conference board for the last 12 years:

- ASQ,http://www.asq.org
- ASQF, http://www.asqf.de
- DELTA, http://www.delta.dk
- ISCN, http://www.iscn.com
- SINTEF, http://www.sintef.no
- STTF,http://www.sttf.fi

## EuroSPI Scientific Program Committee

EuroSPI established an international committee of selected well-known experts in SPI who are willing to be mentioned in the program and to review a set of papers each year. The list below represents the Research Program Committee members. EuroSPI2 also has a separate Industrial Program Committee responsible for the industry/experience contributions.

| | |
|---|---|
| Alain Abran | ETS-University of Quebec, Canada |
| Alberto Sillitti | Free University of Bolzano, Italy |
| Anca Draghici | Universitatea Politehnica din Timisoara, Romania |
| Andreas Riel | Grenoble Institute of Technology, France |
| Antonia Mas Pichaco | Universitat de les Illes Balears, Spain |
| Antonio De Amescua | Carlos III University of Madrid, Spain |
| Bee Bee Chua | University of Technology Sydney, Australia |
| Christian Kreiner | Graz University of Technology, Austria |
| Christiane Gresse von Wangenheim | Federal University of Santa Catarina, Brazil |
| Darren Dalcher | Middlesex University, UK |
| Dieter Landes | Fachhochschule Coburg, Germany |
| Dietmar Winkler | Vienna University of Technology, Austria |
| Fergal McCaffery | Dundalk Institute of Technology, Ireland |
| Javier Garíca-Guzman | Carlos III University of Madrid, Spain |
| Jose Antonio Calvo-Manzano | Universidad Politecnica de Madrid, Spain |
| Keith Phalp | Bournemouth University, UK |
| Kerstin Siakas | Alexander Technological Educational Institute of Thessaloniki, Greece |

| | |
|---|---|
| Luigi Buglione | Ingegneria Informatica, Italy |
| Marion Lepmets | Dundalk Institute of Technology, Ireland |
| Markku Oivo | University of Oulu, Finland |
| Michael Reiner | IMC Fachhochschule Krems, Austria |
| Patricia McQuaid | California Polytechnic State University, USA |
| Paul Clarke | Dundalk Institute of Technology, Ireland |
| Paula Ventura Martins | FCT University of Algarve, Portugal |
| Ricardo Colomo Palacios | Universidad Carlos III de Madrid, Spain |
| Rory V. O'Connor | Dublin City University, Ireland |
| Serge Tichkiewitch | Grenoble Institute of Technology, France |
| Timo Mäkinen | Tampere University of Technology, Finland |
| Timo Varkoi | Tampere University of Technology, Finland |
| Torgeir Dingsøyr | SINTEF ICT, Norway |
| Valentine Casey | Dundalk Institute of Technology, Ireland |

## General Chair

Richard Messnarz

## Scientific Chairs

Fergal McCaffery
Rory V. O'Connor

The experience portfolio of the Chairs covers different market segments, different sizes of organizations, and different SPI approaches. This strengthens the fundamental principle of EuroSPI to cover a variety of different markets, experiences, and approaches.

## Acknowledgments

Some contributions published in this book have been funded with support from the European Commission. European projects (supporting ECQA and EuroSPI) contributed to this Springer book including SafEUr (ECQA Certified Safety Manager), SIMS (ECQA Certified Social Media Expert), VALO (ECQA Certified Valorisation Expert), I2E (Idea to Enterprise), AQUA (Knowledge Alliance for Training Quality and Excellence in Automotive), LSSH (Lean Six Sigma for Health Care).

*In this case the publications reflect the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

Education and Culture DG

Lifelong Learning Programme

# Table of Contents

## SPI Safety and Regulation Issues



## SPI Lifecycle and Models



## SPI Quality and Testing Issues

## SPI Networks and Teams



## SPI and Reference Models



## SPI Implementation

## Selected Key Notes and Workshop Papers

## Agile organisations and an agile management process group



## Managing Diversity and Innovation

## SPI and Measurement



## Risk Management and Functional Safety Standards

# Making Software Safety Assessable and Transparent

Risto Nevalainen[1], Alejandra Ruiz[2], and Timo Varkoi[3]

[1] Spinet Oy, Finland
[2] Tecnalia, Spain
[3] Finnish Software Measurement Association – FiSMA ry, Finland
risto.nevalainen@spinet.fi, alejandra.ruiz@tecnalia.com,
timo.varkoi@fisma.fi

**Abstract.** Most formal assessment and evaluation techniques and standards assume that software can be analysed like any physical item. In safety-critical systems, software is an important component providing functionality. Often it is also the most difficult component to assess. Balanced use of process assessment and product evaluation methods is needed, because lack of transparency in software must be compensated with a more formal development process. Safety case is an effective approach to demonstrate safety, and then both process and product are necessary evidence types. Safety is also a likely candidate to be approached as a process quality characteristic. Here we present a tentative set of process quality attributes that support achievement of safety requirements of a software product.

**Keywords:** software process, process assessment, software safety.

## 1 Introduction

Critical systems are defined as those that in case of an incident or misbehaviour can lead to an accident that will put people or the environment in danger, resulting in injuries and or casualties. Safety is considered as a general property of the whole system and so its plans, developments and implementations must follow strict rules in order to prevent failures of the system and their consequences and risks.

Software-based systems are increasingly important in safety. They replace old wired and analog systems, and they also bring new technologies in safety. They are more standardized and functionality-rich than earlier generations. We can even say that they are more reliable. At least we can use diversity and redundancy more effectively, because digital systems are typically cheaper than old analog systems.

But software brings also problems. Behaviour of software is rather deterministic (i.e. exactly predictable) than probabilistic (i.e. likely to happen). We have to compensate these deficiencies somehow, for example by formal and visible process and by extensive documentation. Still some uncertainty remains and the ultimate “zero defect” or “high reliability” goal is very difficult to achieve.

To some extent we can even challenge the current definitions of safety. For instance, Leveson [1] states that: *“Highly reliable software is not necessarily safe. Increasing software reliability will have only minimal impact on safety.”* With control

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 1–12, 2013.

systems reliability and safety are often confused and the same principles that have worked with hardware are applied to software. In addition, there is no explicit relationship between process quality and product quality. However, the development processes affect the quality of software, including its safety. The main concern should be in management and development of requirements [2].

This paper presents three approaches for assessments of software safety: alignment of process assessment and product evaluation methods with a new concept of property; presenting a new process quality characteristic for safety; and application of safety cases to support safety assessment. The approaches are developed based on our experiences in various safety-critical domains. The approaches are continually evaluated partly in on-going research projects but also in real assessments.

# 2 Process and Product Perspectives in Assessing Software Safety

## 2.1 Different Approaches in Safety Assessment and Evaluation

We can evaluate safety-critical software from several viewpoints. The key output, the software product, can be evaluated against a predefined set of e.g. quality requirements. Safety assessment can study both the product and the processes used in development and use of the software, based often on domain specific standards. Process assessment focuses typically on the product development phase. All these approaches (Table 1) produce valuable information in building trust on the safety of the product. So far, harmonization of these approaches is missing for safety-critical software.

**Table 1.** Comparison of main approaches in software safety evaluation

| Topic | Product evaluation approach | Safety assessment approach | Process assessment approach |
|---|---|---|---|
| Main purpose of the approach | To analyse and show compliance of product (artefact) by using selected criteria | To demonstrate compliance with a selected reference (standard) | To demonstrate capability to develop, deliver and improve |
| Main focus in safety-critical domain | Product quality, especially reliability metric and data, for example MTBF | Compliance with generic or domain specific safety standard, certification | Process evidences to demonstrate achievement of safety management and engineering |
| Specifics of each approach | Internal, external, in use metric | Inspections, reviews, V&V evidences, technical practices and methods | Professional practices, work products, capability levels |
| Commonalities with other approaches | V&V metric, measurement and analysis practices | Engineering methods and competences | Process results, like mandatory work products |
| Typical standard(s) and models | ISO/IEC 25000 family (SQUARE) | IEC 61508, ISO 26262 IEC 60880, IEC 62304 | ISO/IEC 15504 (SPICE), Automotive SPICE, Nuclear SPICE |

Product evidences can also serve as safety assessment and process assessment evidences and vice versa. So, it is meaningful to harmonise those approaches to support each other. An example could be traceability, which is a direct requirement in safety assessment standards and in process assessment models. Another good example is testing coverage, which can be classified as both product and process evidence for verification and validation (V&V) activities.

Product quality model ISO/IEC 25010 [3] (known also as SQUARE framework) includes eight characteristics in internal and external metric and five characteristics in in-use metric. Reliability is one characteristic, including Maturity, Availability, Fault tolerance and Recoverability as sub-characteristics. Safety is less obvious sub-characteristic in in-use model, belonging to Freedom from risk characteristic. It is called there “Health and safety risk mitigation”. Safety is then most relevant in existing operational systems. This view is limited, because safety can and should be built in the system and software by a rigorous development process. Safety can be seen as a combination of process quality and product quality. Main metric for safety in ISO/IEC 25024 draft standard is operational experience, expressed as number of failures / cumulative operation time per given period (typically hour).

Safety assessment is a widely used method in certification. It contains typically different analyses to calculate system reliability and risks for failure, as required in the selected reference standard. The most common standard is IEC 61508, Functional safety [4]. Quite common is to make safety assessment against some Safety Integrity Level (SIL). This approach is based on probabilistic behaviour of the system. It works well for the whole system and for hardware components.

Software is more problematic, because its behaviour is rather deterministic than probabilistic. Detailed checklists are typically used to cover requirements in the selected reference standard. The result of safety assessment can be a statement of conformity or certificate. This result can be very valuable if the reference standard is reasonably up to date and the system under evaluation does not include too experimental technologies. The open issue is trust on the conformity in reference standard, and is that any guarantee of software safety.

Process quality can be covered by the SPICE process assessment. It is based on an international standard, currently ISO/IEC 15504 [5] (in the future ISO/IEC 330xx series [6]). ISO/IEC 15504 origins are in generic software development. Some domains have developed their own variants of the framework, such as Automotive SPICE and SPICE for SPACE. The latest domain specific Process Assessment Model (PAM), developed based on ISO/IEC 15504 principles, is Nuclear SPICE that is intended to address the highest safety requirements. This work is a part of a large Finnish nuclear safety SAFIR 2014 research program [7].

One important topic in process quality and assessment is the extent of validation and verification (V&V) in the software lifecycle. The safety lifecycle contains normally a quality assurance process, for example independent process review or audit. Additionally, independent technical reviews, independent tests and acceptance phases can exist. Process assessment in safety context is normally a mix of basic SPICE type approach and use of selected safety standard(s).

These basic approaches have also significant overlaps. Safety standards have direct requirements for product or system, even they are mostly process centric. Good conformity with standards has always evidences from both product and process.

Also process assessment has both process and product view. Work products are direct evidences of process in SPICE model, at the same logical level as process specific practices. Generic work products are evidences for higher capability and also for safety in such context.

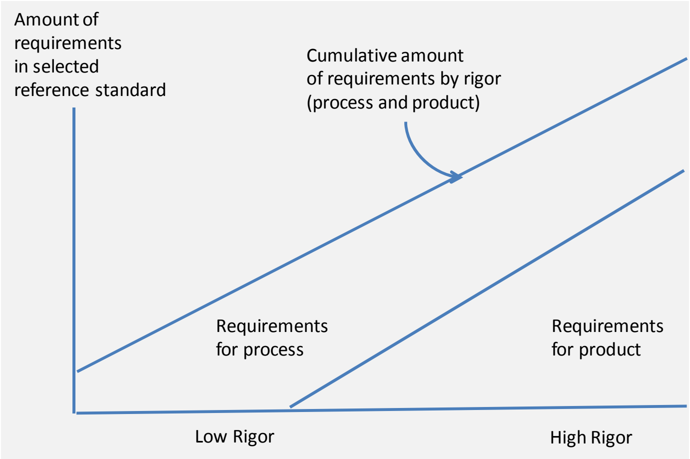



**Fig. 1.** Requirements for process and product at different levels of rigor in safety standards

### 2.2 Integration of Product and Process Quality

SQUARE framework defines product quality at three levels: the overall quality, selected product quality characteristics and further selected product quality attributes. In current version of SQUARE there are 13 product quality characteristics. The work is still on-going in defining product quality attributes and their candidate metrics. The origin of SQUARE is in software, so we can interpret the concept of product quality as "software quality".

We have proposed for Process Assessment standardization community to use similar hierarchy for process quality. Currently ISO/IEC JTC1/SC 7 Working Group 10 is progressing to upgrade 15504 set of standards and to develop a new set of assessment requirements as ISO/IEC 33000 set of standards. This development introduces a new concept of process quality. Process quality concept harmonizes the terminology with product quality (Figure 2).

We can see immediately useful combinations, for example to study which processes contribute in some selected product quality attribute.

Going further, we can add concept of property both in product and process quality. Property can be seen as a real life instance of some attribute. Obviously, real life does not classify so beautifully as our quality concepts. So, any property can be either product property, process property or both. The usefulness of property concept is in its details. It can be observed and even measured objectively.

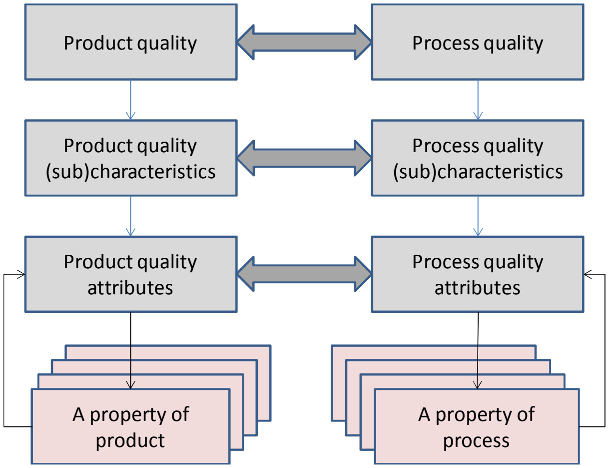



**Fig. 2.** Alignment of product and process quality concepts

## 2.3 Use of Methods and Properties in Nuclear SPICE

In this chapter we use Nuclear SPICE [7] as the reference model to explain how methods and properties can be used as evidence for product and process. Nuclear standards IEC 61513 and further IEC 60880 are based in IEC 61508 series and may include all SIL levels in their requirements. Nuclear standards have a different concept of classes (1, 2, 3) and categories (A, B, C). Nuclear SPICE is designed to satisfy all classes and all categories. So, it shall cover also all techniques and methods included in IEC 61508. Of course, engineering judgment is needed in using methods because otherwise model would be too heavy for practical use.

IEC 61508-7:2010 Annex C [8] lists the topics in which the concept of safety property is proposed. In most cases, it is fairly easy to find corresponding processes in Nuclear SPICE as the list above, as seen in Table 2.

**Table 2.** Mapping topics of safety properties with Nuclear SPICE processes

| Topic in IEC 61508-7 Annex C | Corresponding process in Nuclear SPICE | Comments |
|---|---|---|
| Software Safety Requirements Specification | DEV.1 Software requirements analysis | |
| Software architecture design | DEV.2 Software architectural design | |
| Support tools and programming language | - | This process is mainly missing in SPICE |
| Software detailed design | DEV.3 Software detailed design | |
| Software module testing and integration | DEV.4 Software construction<br>DEV.5 Software integration | |
| Programmable electronics integration (hardware and software) | ENG.5 System integration | ENG.5 needs interpretation but has good match |
| Software aspects of system safety validation | ENG.5 System integration<br>ENG.6 Systems qualification testing<br>SAF.2 Safety Engineering | Scattered coverage in SPICE processes. |

**Table 2.** (*continued*)

| | | |
|---|---|---|
| Software modification | SUP.8 Software problem resolution<br>SUP.9 Software change request management | |
| Software verification | SUP.4 Verification | |
| Functional safety assessment | SAF.1 Safety Management<br>SAF.3 Safety Qualification | Scattered coverage in SPICE processes. |

We can also see that in some cases mapping is not straightforward. For example in topic "Support tools and programming language" no one Nuclear SPICE process covers it, at least not in this level of details. It needs interpretation and also further development of Nuclear SPICE.

One important finding is also that the set of properties in IEC 61508 does not cover all relevant topics in nuclear safety. Nuclear SPICE has an extensive set of processes for system level specifications and design, and properties are missing there. Also quality assurance (SUP.3 in Nuclear SPICE) has no properties. The conclusion is that also other kinds of evidences are needed than what we have specified so far. The software SPICE (ISO/IEC 15504-5:2012) can be used as such, but it needs more engineering judgment and interpretation in some topics, than what is maybe acceptable.

# 3 Safety from Process Quality Viewpoint

## 3.1 Process Quality Characteristic

In the current development of process assessment standards, new concepts are adapted that enable new approaches to address process quality. Safety can be considered as one example of process quality characteristics, which could be used in assessing process quality when developing software for safety-critical domains.

The adopted principles include that process quality is composed of quality characteristics, where the required set of characteristics depends on the applicable stakeholder needs and organization's business goals. In addition, process quality shall be measurable. The key terms are defined as follows [6]:

Process quality

- ability of a process to satisfy stated and implied stakeholder needs when used in a specified context

Process quality characteristic

- a measurable aspect of process quality; category of process attributes that are significant to process quality

By nature, process assessments are based on sampling and thus do not provide proper data for any probabilistic assessments. Therefore, it is important to understand that process assessment cannot be used to qualify a software or system product. However, process assessments can point out the risks related to achieve the required level of product or system quality, including safety.

### 3.2 Safety Process Quality Attributes

We have specified a preliminary model to address safety by process assessment [2]. At this stage, the model is tentative and intended to serve as a starting point for discussion on how processes influence in implementing software safety requirements. The model consists of two sets of process quality attributes (PA) for process assessment in safety domain. The basic set is intended to include attributes that meet the elementary requirements for trustworthy software development. The extended set adds process attributes that support management of processes that support safety activities. The attributes should be applied typically on development and quality assurance processes. Management processes are often too generic for this purpose. Applicable processes are described in the Nuclear SPICE PAM.

**Table 3.** Basic set of process quality attributes for safety

| Process Attribute | Description |
|---|---|
| PA 1 | Process performance<br>- process achieves its defined process outcomes |
| PA 2 | Dependability<br>- reliability; process performs as required in normal conditions<br>- availability; process can be performed when needed<br>- maintainability; process can be modified easily to add capabilities; performance can be improved; faults and errors can be corrected |
| PA 3 | Requirements control<br>- traceability<br>- specifications coverage<br>- constraints<br>- safety analysis<br>- reuse |
| PA 4 | Safety engineering<br>- safety demonstration<br>- reviews<br>- verification and validation<br>- quality assurance |

**Table 4.** Extended set of process quality attributes for safety

| Process Attribute | Description |
|---|---|
| PA 5 | Safety management<br>- safety strategy alignment<br>- safety life cycle; defined activities involved in the implementation of safety-related systems<br>- responsibilities and resourcing<br>- monitoring<br>- test and simulation environments |
| PA 6 | Compliance<br>- standards<br>- defined process<br>- process tailoring |

**Table 4.** (*continued*)

| | |
|---|---|
| PA 7 | Quantitative management<br>- quantitative analysis; measurement objectives; measures<br>- quantitative control; techniques; causes of variation |
| PA 8 | Risk management<br>- management of events that effect achievement of business goals<br>- qualitative and quantitative risk analysis for a process; probabilistic risk analysis |
| PA 9 | Information security<br>- preservation of confidentiality, integrity and accessibility of information during the execution of a process |

The presented process safety attributes and sets shall be further elaborated and extended in descriptions. To enable efficient assessments, the process attributes will be completed with appropriate assessment evidence classes, including Generic Practices, Generic Resources and Generic Work Products.

## 4 Safety Cases as a Support Tool for Safety Assessment

Safety case is a requirement in many safety standards. Safety cases were originally inspired in the nuclear industry and have been used for more than 50 years. They have successfully been used also in other industries like chemical, military systems, the off-shore oil industry, rail transport, and recently in the aviation and automotive industries. According to the definition from ISO 26262, it is defined as an "argument that the safety requirements for an item are complete and satisfied by evidence complied from work products of the safety activities during development" [9]. Safety argumentation provides a valuable tool for the development of critical systems. A safety case is based on a goal representing an assertion that can be assessed as true or false. To reach to this target goal, we should construct an argumentation based on several items as described by J.R. Inge [10]:

- The scope of the system or activity being addressed, together with details of its context or environment.
- The management system used to ensure safety goal.
- The requirements, legislation, standards and policies applicable, with evidence that they have been met or complied with.
- Evidence that risks have been identified and appropriately controlled, and that the residual level of risk is acceptable.
- Independent assurance that the argument and evidence presented is sufficient for the application in question.

But sometimes, safety cases are seen with criticism due to several reasons according to Johnson and Robins [11]. One of the main complaints against safety cases is that it is always possible to find or produce evidence that something is safe. It is the confidence level that is put into that evidence what gives strength to the argumentation. Unfortunately, for safety analysis there is no complete mathematical theory to base arguments and guarantee completeness. It is also important to highlight that argumentation reasoning is sometimes made under non-explicit assumptions. Rasche enumerates the following general problems related to safety cases [12]:

- The amount of work required to construct a safety case including the specialized and costly (outside) resources required
- Problems associated with obtaining and validating data to justify a probabilistic risk analysis.
- Too much focus on technical risk and not enough on meeting the needs of workers

Unfortunately there is not a method from preventing in given inappropriate argumentation. The ideal scenario for creating strong, complete safety cases is to provide an independent, non-subjective argumentation. This could be reached by demonstrating that major hazards of installation and the risks to personnel therein have been identified and appropriate controls provided.

## 4.1 Safety Cases Argumentation

Standards tend to be prescriptive regarding specific solutions and process-oriented rather than product oriented. Safety Cases could be the bridge between these two approaches and balance the process-oriented with the product-oriented approaches. A strategy can be used to describe generic approaches to the arguments that are used in support of a goal or claim, such as reference to appropriate standard sections. There has been some research on this line that indicate that a solution for this could be the use of goal-based safety case such as described by Stensrud et al. at WOSORCER workshop 2011 [13] where they propose a hybrid approach to transform prescriptive elements in the standard IEC 61508 from a table format into a safety case format, creating then safety case patterns that map with the prescriptive elements from the standard.

Weaver et al. [14] presented a safety case framework that includes the top level software safety argument where the top level goal is that the system is acceptably safe. The top level goal is further broken down into sub-goals including that the safety requirements are valid. On the decomposition of these sub-goals we are able to link with the ideas previously commented from Stensrud et al. [13] to map with the IEC 61508 requirements and go deeper into the product characteristics.

In general, within the safety assurance research community as described by Flood and Habli, safety cases are increasingly viewed as consisting of three types of arguments [15]:

- risk (or “primary”) arguments – that aim to establish that the system is acceptably safe to be deployed.
- confidence (or “backing”) arguments – that are used to justify that sufficient confidence can be placed in evidence and inferences of the risk arguments
- compliance arguments – that show that requirements of the applicable standards have been satisfied

Special attention should be put into the confidence arguments, which are the key to make strong and credible argumentation. In ISO 26262 standards one of the requirements is to demonstrate a safety culture within the company. Well defined safety engineering processes are important as they offer confidence on the argument, and also the evidence that those processes are being followed demonstrate that the best practices identified by the company are put into practice.

### 4.2 Evolutionary Safety Cases

For a long time, it has been usual to leave the development of the safety case to the end of the project; however this approach can lead into a costly strategy as changes to the design at that time are very expensive. With the same view of early validation and verification, safety cases as internal audits can help in reducing the possible risks.

The creation of evolutionary safety cases along the project as a way to both mitigate possible risks, follow up design decisions with impact on safety, and at the same time as a powerful tool to support management from the safety point of view.

ISO 26262 encourages the idea of incremental safety cases. It recommends that the safety case should be developed along with the system. The standard proposes to have refined safety cases in which with each phase, the information is completed and the strategy for the next phase is defined.

We can define three different stages of the safety case:

- Preliminary safety case: At this stage, we will include information regarding: system scope; top safety requirements; main hazards; possible strategies; development approach; type of evidences needed. In this stage it should be assured that all hazards are covered and the mitigation strategies are possible to be put into practice (within budget, time etc.).
- Interim safety case: At this stage, we are able to increase the confidence on the design in comparison with argumentation from the preliminary safety case. In order to strength the argumentation, evidence for the preliminary validation is important to address the independence of the validation results, giving more confidence.
- Final safety case: includes complete arguments, from all types of argumentation described in previous section. The evidences such as: observation, measurements, testing and analysis of the implemented system support all possible arguments.

It is important to highlight that safety cases can be modified or changed throughout the operational life of the system, as additional safety evidence becomes available or new risk appears.

### 4.3 Safety Cases as a Support Tool for Safety Assessment

It is not rare, while doing safety assessment, to be presented long reports referencing to evidences, but those reports have lack in clarity on how those evidences relate to the safety requirements and how it is understood to comply with the standard.

Avionics standards do allow an applicant to propose “alternative methods of compliance" for some objectives, provided it can be shown how their new methods satisfy the “intention" of the objectives. The difficulty is that the intent of most objectives is not formulated explicitly. Thus, a reasonable enhancement to guidelines such as

DO-178B would be to include documentation of the intent of each objective. We could go further than this, and to supply a full argument that the evidence required by the standard does ensure satisfaction of explicitly stated safety goals. Such argumentation would be generic at standard level, but it could be also applied at the level of safety demonstration for a particular certification project.

ISO 26262 proposes to tailor some activities in order to propose forms to comply with the standard and at the same time that are adequate for the project. This involves interpretation of the standard and needs to be understood and agreed by both the company and the person in charge of the assessment.

The idea behind a safety case described by Tim Kelly [16] is that the application of an argumentation approach to the concept of target compatibility would require definitions, assumptions, and limitations to be made visible. This allows a much clearer evaluation for the contribution and limit to the overall correctness of the software and therefore its contribution to safety of the system.

On the SPICE assessment different indicators are defined. The indicators can be seen as the different goals to achieve. The base practices are the strategies which can be followed in order to comply with the objectives and the output work products can be seen as evidences for those strategies to been followed. The association with the SPICE assessment is easily mapped into the compliance argument type that was described before. The capability dimension of the SPICE model and how this capability is improved offers the confidence argumentation. However in the SPICE model the explicit risk arguments are missing or re implicit. Those arguments are linked mainly with the product properties (Fig. 2). The implementation of base practices can differ from a company to another. The negotiation between the company and the assessor can be more efficient and fruitful when the rationale (argumentation) is well understood and shared by all parties, and safety cases can be very helpful in this. Safety cases are a powerful tool to express the argumentation behind the compliance of the different requirements from the standards and at the same time, are able to express in a comprehensive and clear way many design decisions in relation with safety requirements.

## 5 Conclusions

This paper proposes three different approaches to improve assessability of software safety by presenting. First, integrated approach on product and process quality balances the use of process assessment and product evaluation methods. A new concept of property is added both in product and process quality. Second, safety is considered as a process quality characteristic. This enables assessment of software development processes using a specific set of process safety attributes. Third, safety cases can be used to support safety assessment and demonstration. Safety cases provide the argumentation for meeting the safety requirements of systems. Use of these approaches needs to be considered case by case. The overall critically of the application is the main driver in selecting an appropriate scope and combination of methods for safety assessment. The aim is to improve trust on software safety and to minimize the risks.

**Acknowledgements.** This work has been partially funded by the National Authorities involved in RECOMP and the Advanced Research & Technology for Embedded Intelligence and Systems (ARTEMIS) within the project RECOMP under Grant agreement no. 100202. Any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of funding agencies. This work has been partially funded by Finnish national nuclear safety program SAFIR2014. In project CORSICA, new approaches and V&V methods have been developed for software safety. A method called Nuclear SPICE implements many of the ideas in this article. The research leading to these results has received funding from the FP7 programme under grant agreement n° 289011 (OPENCOSS).

## References

1. Leveson, N.G.: Engineering A Safer World: Systems Thinking Applied to Safety. MIT (2011)
2. Varkoi, T.: Safety as a Process Quality Characteristic. In: Proceedings of SPICE 2013 Conference (accepted for publication, 2013)
3. ISO/IEC 25010:2011, Systems and software engineering–Systems and software Quality Requirements and Evaluation (SQuaRE)–System and software quality models (2011)
4. IEC 61508-3 Ed. 2.0, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements (2009)
5. ISO/IEC 15504-5:2006, Information technology – Process assessment – Part 5: An exemplar Process Assessment Model (2006)
6. ISO/IEC 33001 DIS, Information technology – Process assessment – Concepts and terminology. ISO/IEC (2013)
7. FiSMA 2011-1: S4N Method Description - Nuclear SPICE PRM and PAM. FiSMA (2012)
8. IEC 61508-7 Ed. 2.0, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 7: Overview of techniques and measures (2009)
9. ISO 26262, Road vehicles – Functional safety, ISO (2011)
10. Inge, J.R.: The Safety Case: Its development and use in the United Kingdom. In: Equipment Safety Assurance Symposium, Bristol, UK (2007)
11. Johnson, C.W., Robins, D.A.: Myths and barriers to the introduction of safety cases in space-based systems. In: 29th International Systems Safety Society, Las Vegas, USA (2011)
12. Rasche, T.: Development of a safety case methodology for the Minerals Industry – a discussion paper. Minerals Industry Safety and Health Center (2001)
13. Stensrud, E., Skramstad, T., Li, J., Xie, J.: Towards Goal-Based Software Safety Certification Based on Prescriptive Standards. In: First International Workshop on Software Certification, WoSoCER (2011)
14. Weaver, R.A., McDermid, J.A., Kelly, T.P.: Software Safety Arguments: Towards a Systematic Categorisation of Evidence. In: Proceedings of the 20th International System Safety Conference (ISSC), System Safety Society, Denver (2002)
15. Flood, M., Habli, I.: Multi-Viewpoint Safety Cases. In: Proceedings of the 6th IET International System Safety Conference, Birmingham, United Kingdom (2011)
16. Kelly, T.: Arguing Safety - A Systematic Approach to Managing Safety Cases. PhD thesis, Department of Computer Science, The University of York (1998)

# Sector-Based Improvement of the Information Security Risk Management Process in the Context of Telecommunications Regulation

Nicolas Mayer, Jocelyn Aubert, Hervé Cholez, and Eric Grandry

CRP Henri Tudor, 29 avenue John F. Kennedy, L-1855 Luxembourg, Luxembourg
{nicolas.mayer,jocelyn.aubert,
herve.cholez,eric.grandry}@tudor.lu

**Abstract.** The current European regulation on public communications networks requires today that Telecommunications Service Providers (TSPs) take appropriate technical and organizational measures to manage the risks posed to security of networks and services. However, a key issue in this process is the risk identification activity, which roughly consists in defining what are the relevant risks regarding the business operated and the architecture in place. The same problem appears when selecting relevant security controls. The research question discussed in this paper is: how to adapt generic Information Security Risk Management (ISRM) process and practices to the telecommunications sector? To answer this research question, a four-step research method has been established and is presented in this paper. The outcome is an improved ISRM process in the context of the telecommunications regulation.

## 1 Introduction

Information systems are everywhere and their roles are central for all organisations because of the increasing amount of information managed during the last decades. Due to the criticality of the information exchanged, more and more supervision is needed and operated by national, European or even international authorities. This supervision and the associated regulations are often defined at a sector-based level. One of the leading sector having adopted such a model is the financial sector, with a National Regulatory Authority (NRA) established in every country and dealing with sector-based regulations, defined at the international and/or national level (e.g., Basel II agreements, the Sarbanes-Oxley Act, etc.). The same approach is currently applied to the telecommunications sector, with a supervision of the Telecommunications Service Providers (TSPs) operated by the NRAs of the different countries, such as ILR (*Institut Luxembourgeois de Régulation*) that is the NRA in Luxembourg.

The recent EU Directive 2009/140/EC [1] amends existing directives on framework (2002/21/EC), authorization (2002/20/EC), and access (2002/19/EC) of electronic communications networks and facilities. This directive should be transposed into a national legislation by all the EU member states and it has been done by the Luxembourg country with the publication of the law of the 27th February

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 13–24, 2013.

2011 on electronic communications networks and services [2]. The EU Directive introduces Article 13a on security and integrity of networks and services. This article says that Member States shall ensure that providers of public communications networks "take appropriate technical and organizational measures to appropriately manage the risks posed to security of networks and services" [1]. In addition, the article point out that "these measures shall ensure a level of security appropriate to the risk presented".

In 2010, the European Network and Information Security Agency (ENISA), as the centre of network and information security expertise for the European Union, initiated a series of meetings with the European Commission, Ministries, and Telecommunications NRAs to achieve a harmonized implementation of Article 13a. The result of this work was published in December 2011 in a document entitled "Technical Guideline for Minimum Security Measures" [3]. This document gives guidance to NRAs about the implementation of Article 13a. The starting point of the Minimum Security Measures is to identify, evaluate, and prioritise information security risks by establishing and maintaining an appropriate governance and risk management framework. This document explains also that the telecommunications organisations "should perform risk assessments, specific for their particular setting" [3]. For example, a particular characteristic of the telecommunications sector with regard to risk assessment is that the focus in put on the integrity of the networks and on the continuity of supply of services.

Based on this context, the research question discussed in this paper is: how to adapt generic Information Security Risk Management (ISRM) process and practices to the telecommunications sector? The outcome is a fine-tuned method, supported by a tool, aiming at helping the TSPs to perform efficiently ISRM, in order to be compliant with the European [1] and national [2] regulation. The main contribution of this paper is not on the resulting method (and tool) in itself, but is focussed on the approach followed to improve and fine-tune the ISRM process for our context.

Section 2 is an introduction to ISRM and explains the context of the telecommunications sector. Section 3 presents the research method applied to improve the ISRM process for the telecommunications sector. The two first steps of this research method are about the modelling of the telecommunications services through business processes and a reference architecture and are explained in Section 4. Then, the third step of the research method about the definition of a service-related knowledge base of risks is depicted in Section 5. Finally, Section 6 is about current state of the research work, conclusion and future work.

## 2 Information Security Risk and the Telecommunications Sector

The complete approach described in this paper is focussed on the concept of service, and more specifically on the concept of telecommunications service. A telecommunications service is a service provided by a TSP and "normally supplied for remuneration, which wholly or mainly provides the conveyance of signals on electronic communications” [4]. The conveyance of signals consists in the transmission, between or among points specified by the user, of information of the user's choosing. The TSP has the responsibility for the acceptance, transmission, and delivery of the message [5]. Examples of telecommunications services registered and monitored by ILR are: fixed-line telephony service, mobile telephony service, dial-up internet access service, mobile internet access service, etc.

Each service can be decomposed in a set of business processes that are needed to establish and provide the service. Each telecommunications service, and thus each related business process, is realized by the information system of the telecommunications organization. We consider that the information system of a TSP is “a system in which human participants and/or machines perform work (processes and activities) using information, technology, and other resources to produce informational products and/or services for internal or external customers” [6]. The components of the information system are organized so that the system reaches its objectives: this organization is described in the architecture of the system: “an architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution” [7]. As the system has the objective of fulfilling the telecommunications services, we will name the architecture that describes the system, the *telecommunications service architecture*. In the TSP sector, there are many types of architecture components, including information, hardware, network components, software, intangibles, but also people and facilities playing a role in the information system and so in its security [7], [8].

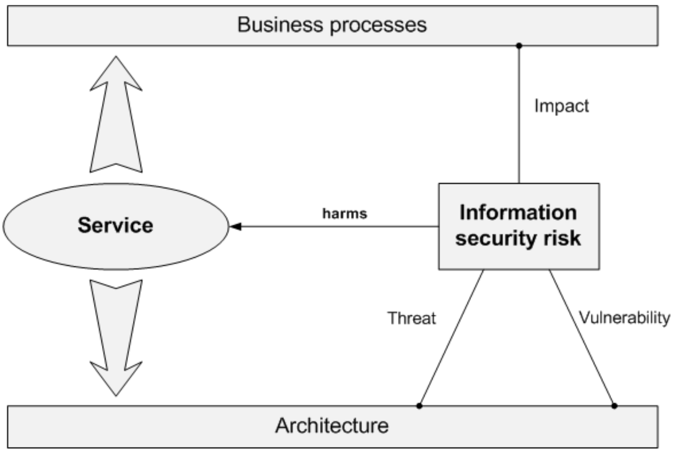



**Fig. 1.** ISRM in the context of the telecommunications sector

From a security point of view, risks are harming the telecommunications services. An information security risk is defined by three components: Risk = Threat * Vulnerability * Impact. In other words, risk is characterized by the opportunity of a threat targeting components of the architecture, to exploit one or more vulnerabilities originating from the design decisions of the architecture, and leading to an impact on business processes [8]. An example of information security risk is: a thief penetrating a telephone exchange (threat) because of lack of physical access control (vulnerability), stealing cables and thereby provoking loss of availability of the telephony service (impact). Fig. 1 depicts the main components described in this section and their relationships.

## 3 Research Method

In order to reach our objective of adapting the ISRM process and practices to the telecommunications sector in a structured way, we followed a research method composed of four steps described below. Although the target sector is highly competitive, we made the actors (service providers and the regulator) collaborate in order to co-construct the results of each step of the research method. It is indeed a way to ensure that the results are designed according to the needs and constraints of the actors, and that they are adopted by the end-users when they are rolled out. Workshops have been organised to present the objectives, and then to design and validate the components of the method. A representative panel of TSPs, covering all services, infrastructures and telecommunications media (e.g., optical fibre, satellite, etc.) were appointed by ILR to participate to the workshops. However, all other TSPs were invited to provide information by email through surveys performed before and after the different workshops.

**Step 1** – *Modelling of the telecommunications services through business processes*: The first step consists in defining the different processes composing each telecommunications service. A literature review is performed in order to identify relevant documentation about telecommunications processes. Then, based on the literature review, a set of business processes is associated to each telecommunications service.

**Step 2** – *Modelling of the telecommunications services through an information system architecture*: The second step of our research method consists in the description of the information system supporting each telecommunications service: listing the components that implement each service permits identifying the relevant threats and vulnerabilities and tracing these back to the actual service. The main challenge in this activity is to select the right level of abstraction in the description of the information system, taking into account the following objectives: (1) we aim at describing the telecommunications information system for the purpose of security risk management; (2) we target a description applicable to all operators providing the selected services in Luxembourg.

To describe the information system, we adopt the ISO/IEC 42010:2007 standard [7], which defines that a system (our telecommunications information system) has an architecture described by an architectural description. More important, the standard acknowledges that the architecture can be described from multiple viewpoints according to the specific concerns of the stakeholders. The scope of the second step of our research method is to build the architectural view that supports the management of information security risks for the telecommunication organisations in Luxembourg: we abstract away both the details that do not pertain to the domain of security risk management, and the specificities of each organisation. Building this telecommunication service architectural view can also be seen as building an industry architecture dedicated to security risk management in telecommunications according to the TOGAF architecture continuum [9]: this architecture is indeed a reference model for each TSP having the objective of managing information security risks.

**Step 3** – *Definition of the service-related knowledge base of risks*: A key issue in ISRM is the risk identification activity, which roughly consists in defining what are the relevant risks, and thus the relevant threats, vulnerabilities and impacts, regarding the business operated and the architecture in place. Some generic knowledge bases already exist [10], [11], helping the analyst in the risk identification phase. However, it is generally difficult for non-experienced people to deal with such a knowledge base and determine what are the relevant sets of risk they need to consider. The same problem appears during the risk treatment phase and the security controls selection [10], [12]. The objective of this step is to predefine for each telecommunications service the (most) relevant threats and vulnerabilities, based on the reference architecture defined during step 2 and the (most) relevant impacts based on the business processes defined during step 1.

**Step 4** – *Integration of the results in a software tool and experimentation*: The different models established during step 1 and step 2, as well as the risk and control knowledge bases established during step 3, are then integrated into a software tool already used to do ISRM in the frame of Information Security Management System (ISMS) establishment [13]. This tool needs to be validated by the NRA of Luxembourg, and then distributed to the TSPs as a support to fulfil their regulatory requirements related to ISRM.

## 4 Modelling of the Telecommunications Services through Business Processes and an Information System Architecture

The two first steps of the research method are about the modelling of the telecommunications services through business processes (step 1) and through an information system architecture (step 2). This section presents these two first and related steps, illustrated and summarised by Fig. 2.

### 4.1 Business Process Modelling

Step 1 of our research method consists in defining the different processes composing each telecommunications service. This step is crucial to understand the TSP activities and to propose a concrete business process framework understandable and meaningful for the TSPs.

As a first stage (**step 1.1**), a literature review was performed in order to identify relevant documentation about telecommunications processes. Several models and documents were studied and analysed. Finally, we mainly focussed on two well-established and well-accepted process models: the Business Process Framework ("eTOM") of the TMForum [14] and the Telecommunications Process Classification Framework of IBM and the American Productivity & Quality Center (APQC) [15]. These two models were widely known by Luxembourgish TSP and sometimes already used. Based on these models, we wanted to establish a customized process model not to give priority to one specific model, and mainly to have a model really adapted to our needs.

The process model was established in **step 1.2**. In order to be accurate and avoid a cumbersome model, we defined the scope of the ISRM to focus only on the core business of TSPs. Thus, we selected only the relevant processes which can have a negative impact on the security or continuity of services provided over the networks. The selection of processes was done by comparing and bringing together processes of the different models. In our scope, the processes of different models were fairly similar, and then the process selection was straightforward. Only some processes were grouped together to avoid complexity and some others were divided to be more significant and relevant for ISRM. Two groups of processes were defined to separate (1) all processes related to the "Enterprise Management Processes" including support processes, strategy processes, etc., and (2) all "Operating Processes" that are directly related to the telecommunications service management. This separation in two groups is particularly useful since TSPs often manage several different services, and to perform ISRM on a high number of processes is a complex and substantial work. Our model allows performing the ISRM process on the "Enterprise Management Processes" only once, independently of the number of services managed by the TSP, because the activities performed in this group of processes are common to all delivered services. Then, the ISRM process is performed on each "Operating Processes" of each telecommunications service, the operating processes being clearly different from one service to another (i.e. involving different architecture components and pursuing different business objectives). In a nutshell, for a TSP with n services, the ISRM process is applied on the following number of processes: (1 * "Enterprise Management Processes") + (n * "Operating Processes").

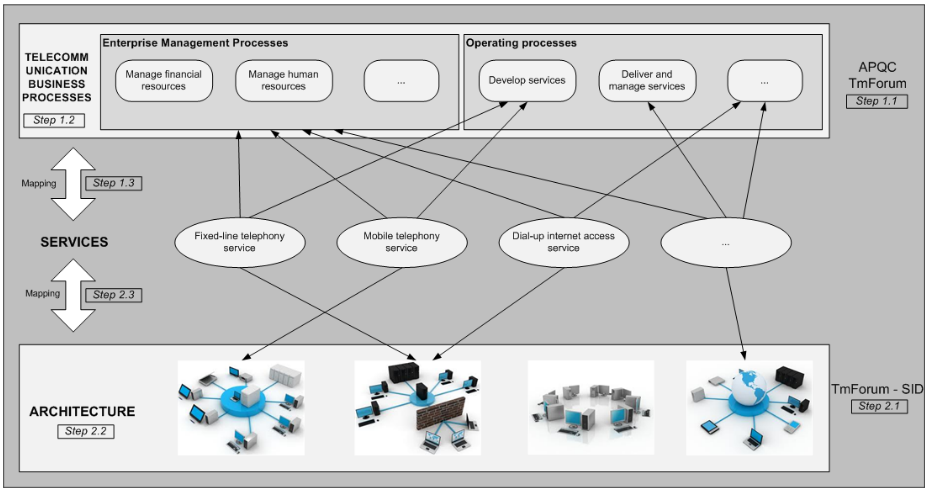



**Fig. 2.** Modelling of the telecommunications services through business processes and an information system architecture

Finally, in **step 1.3**, after having been co-designed by ILR and several TSPs, the process model was submitted to and validated by them. In this final version, the group "Enterprise Management Processes" contains 5 processes:

1.1 Develop vision and strategy,
1.2 Manage financial resources,
1.3 Manage human resources,
1.4 Manage knowledge, research, and change,
1.5 Manage stakeholder and external relationships.

In the second group, "Operating Processes", there are 6 processes:

2.1 Develop and acquire resources (application, computing, and network),
2.2 Manage resources (application, computing, and network),
2.3 Develop services,
2.4 Market & sell services,
2.5 Deliver and manage services,
2.6 Manage supplier/partner relationship.

## 4.2 Architecture Modelling

In this second step, we aim at producing the description of the Telecommunications Service Architecture. This modelling activity was performed iteratively, each iteration dealing with a specific registered service. This guarantees that the scope of the modelling exercise is better managed and that the experience gained in the first iterations is injected in the next ones.

The first stage (**step 2.1**) of this activity aims at proposing a model of the concepts that are relevant in this specific view of the system. We reviewed the literature and the industry standards in order to identify enterprise architecture models of reference for telecommunication. The works of The Open Group and TMForum have been specifically analysed and confronted to the state-of-practice of the national TSPs, and we finally selected the Information Framework (SID) from the TMForum [16]. This model suits well our needs as it is centred on the concept of service, and it describes the relations between service and resource (i.e. architecture components).

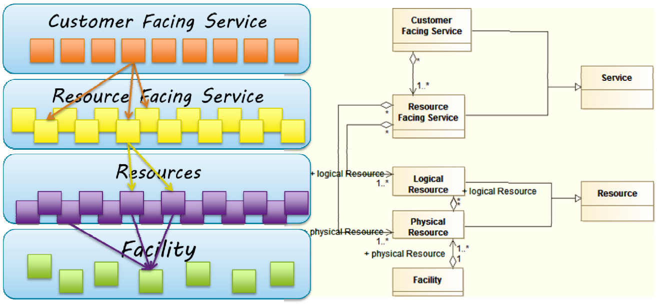



**Fig. 3.** Telecommunications Service Architecture metamodel

As illustrated in Fig. 3, we adopted a layered architecture organised according to the SID model. The Customer Facing Service layer encompasses the services registered at ILR (fixed and mobile phone and data services). The Resource Facing Service layer represents the functional blocks of the architecture: this layer has been omitted at the beginning of the modelling exercise and introduced later for sake of reuse across services. The Resources layer contains all the resources that implement the services. Several classifications of resources (logical vs. physical, asset classification from ITU-T [17]) have been considered, but without actually bringing differentiating value at this stage. Finally, we extended the model with the Facility layer in order to capture the fact that resources are physically localised in facilities.

The second stage (**step 2.2**) of this activity was to identify the resources that are meaningful in the scope of ISRM. Information about the implemented architecture was collected from the TSPs, and the resources were selected according to the abstraction principles described in the approach: is this resource meaningful for all operators, is this resource relevant to identify risks on the services managed by the regulator, etc. This modelling exercise was collaboratively conducted: the active participation of both the operators and the regulator is indeed required in order to reach the objective in the abstraction process. At the end of this stage, we had a list of resources for each telecommunications service.

The last stage (**step 2.3**) was the actual production of the telecommunications service architecture. The resources listed in step 2.2 were put in the context of the services they implement, and the reference model selected in step 2.1 was instantiated. During this modelling activity, the participants naturally grouped the resources according to some shared functions commonly accepted in the industry (Access, Core, Transmission, Business Support, Infrastructure, Security, Devices) leading to the emergence of the Resource Facing Service layer of the service architecture. An architectural description can be represented in multiple forms, from a formal model to a totally informal drawing. In this project, it was decided to represent the architecture as a catalogue of resources implementing each service. This catalogue is currently described in an Excel sheet, although it can later be transformed into a relational database accessible through a web application, or into any other technological form. The most important aspect is that the model (whatever the form of representation) respects the metamodel of the architecture: all elements are instances of the concepts and relate to other elements according to the metamodel. The outcome of step 2 of the research method is thus a catalogue of resources, implementing each registered service, and thus composing the telecommunications service architecture.

## 5 Definition of the Service-Related Knowledge Base of Risks

The third step of the method, illustrated by Fig. 4, aims at defining a service-related knowledge base of risks specially tuned for TSPs. By service-related knowledge base of risks, we mean a set of threats, vulnerabilities and impacts specifically targeting elements of the architecture modelled in the previous step. In addition, with the aim of simplifying and focussing on the telecommunications sector, we strive to simplify the base as much as possible, and to propose to this extent only elements relevant for TSPs. Wherever possible, threats and vulnerabilities are associated to generic elements of architecture, to automatically provide TSPs with possible threats and exploited vulnerabilities. In the frame of their ISRM process, for a given service, TSPs will mainly have to indicate if (pre-selected) threats apply or not, and how their system is vulnerable. It is however important for each TSP to think also about its specificities (at the business or architecture level) potentially implying specific risks, involving non-pre-selected threats/vulnerabilities.

As a first stage (**step 3.1**), an inventory of standards and references proposing knowledge bases of threats was made. Various documents were studied, including documents proposing generic approaches for risk assessment [10], [11], [18]–[22] as well as documents targeting specifically telecommunications-related risk assessment [20], [23], [24]. Threats were analysed and selected to propose a subset of threats focussing essentially (as pointed out by the law) on those harming availability of services and integrity of networks. To avoid duplicates, same threats issued from different documents were selected only once. When applicable, with the intention of simplifying threat selection, threats were grouped as far as possible: for example threats targeting the same type of components, having very close impact, having the same origin (deliberate, accidental, environmental), were checked for being grouped together. In this way, as an example, the following threats, issued from [10], *Climatic phenomenon*, *Seismic phenomenon*, *Volcanic phenomenon*, *Meteorological phenomenon* and *Flood* were grouped into a new threat called *Natural Disaster* since they generally target facilities, have environmental origin and harm integrity and/or availability. On the contrary, some threats were specified, to allow a better applicability to the telecommunications sector. This is, for example, the case of the threat *Theft of equipment* specified by *Theft of cable* meaningful for TSPs. This step led us to identify twenty-two threats. Finally, each threat is associated with generic elements of architecture (defined in step 2) that may be impacted.

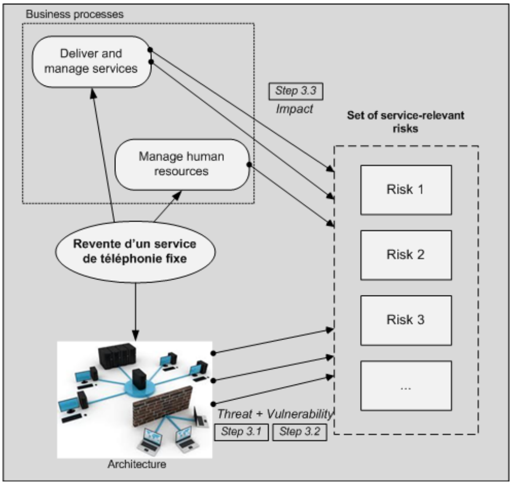



**Fig. 4.** Definition of the service-related knowledge base of risks

Similarly, in **step 3.2**, an inventory of standards and references proposing knowledge bases of vulnerabilities was made [10], [11], [19], [20], [24]. Vulnerabilities potentially exploitable by the threats selected during the previous step were selected. As in Annex D of ISO/IEC 27005 [10], examples of threats that might exploit these vulnerabilities are associated with each vulnerability. In the case of relatively close vulnerabilities issued from different sources, when relevant, an effort of reconciliation has been made. In some cases, vulnerabilities are specified, in order to be more significant for TSPs; this is, for example, the case for the vulnerability *Bad configuration* specified by *Badly configured network routers, gateways or firewalls*. This step led us to identify more than ninety vulnerabilities.

In **step 3.3**, an impact scale was defined. Such scale will be used by TSPs to qualify the effects of a threat exploiting one or more vulnerabilities on the targeted process(es). To that purpose, we rely on the scale provided by ILR for the notification of any breach of security or loss of integrity which is (1) *Between 1% and 2% of customers are affected for at least 3 hours*, (2) *Between 2% and 5% of customers are affected for at least 2 hours*, (3) *Between 5% and 10% of customers are affected for at least 1 hour* and (4) *More than 10% of customers are affected regardless of the length*. Taking into account the possible consequence of a given threat regarding the actual vulnerabilities, TSPs will have to estimate the impact using this scale. Reusing a scale already used by ILR and TSPs should facilitate its acceptance and common understanding.

Finally in **step 3.4** (currently planned), the service-related knowledge base of risks composed of threats, vulnerabilities and impacts is submitted to ILR and to several TSPs of Luxembourg during a workshop in order to collect their feedback and suggestions for improvement. This step is part of an iterative process; threats, vulnerabilities and impacts can be improved following the TSPs comments. The overall idea behind this approach is to reach a consensus, and provide a common reference basis allowing TSPs to perform risk assessment and ILR to more easily compare the results.

## 6 Current State, Conclusion and Future Work

Today, we have established and validated the telecommunications business process model (**step 1**). We have also identified the architecture components used in each telecommunications service (**step 2**). Some enhancements can still be brought like finer classification of services and resources in the architectural description: although we initially prevented from introducing details of technical implementation in the service layers (e.g., phone VoIP vs. PSTN), it might be useful to introduce this separation when specific risks only affect a specific technology; in the same way, a classification of resources according to some model might be useful when identifying generic threats and vulnerabilities. We however have to balance between complexity of the classification and level of reuse. The definition of the service-related knowledge base of risks is not currently finished (**step 3**) and the integration of the results in a software tool and experimentation (**step 4**) is still work in progress. The different threats, vulnerabilities, and impacts on services are defined, but further validation needs to be done. In the same manner, the software tool integrates all current results and the development of the first usable version is complete, but some tests by the future users should be done to adapt the tool to theirs needs and requirements. The final version of the software tool is expected in a few months. After these last validations, the main next step is to experiment the method and the tool on TSPs providing different services and being of different size and maturity level. A presentation of the results and training sessions are also planned to support them during the ISRM process.

In this paper, we have presented our approach to establish an improved process to deal with information security risks in the telecommunications sector. After having detailed our context, i.e. the telecommunications sector in Luxembourg, and our scope, i.e. what we mean by information security risk, the four-step research method we defined and applied is presented. Then, the way we performed these four steps and the results obtained are presented. As explained in the section dedicated to the current state of the work, we know that our results are still subject to validation, because most of the experimentation work is still to be done. We expect to analyse the quality and relevance of the risk management results of the TSPs, and thus to be able to validate the quality of our improved process. The ease of understanding of the process and the efficiency of the work that will be performed by the TSPs during their ISRM activities are also key indicators of the quality of our results. Finally, the use of a co-design approach, involving the different stakeholders from the beginning of the work, seems to us a key asset to quickly reach an optimized process. Regarding future work, our results can be disseminated at the European level and other countries may use them in order to comply with the European legislation. We are currently discussing and presenting our research work to the European telecommunications regulator. Another natural way of extension would be to apply the same research method to other sectors in Luxembourg having defined sector-based regulations, such as the financial sector or the records management sector.

**Acknowledgments.** Thanks to ILR, the NRA of Luxembourg.

## References

[1] Official Journal of the European Union, Directive 2009/140/EC of the European Parliament and of the Council of 25 November 2009 (2009)
[2] Journal Officiel du Grand-Duché de Luxembourg, Loi du 27 février, sur les réseaux et les services de communications électroniques (2011)
[3] Dekker, M., Liveri, D., Catteddu, D., Dupré, L.: Technical Guideline for Minimum Security Measures - Guidance on the security measures in Article 13a. In: ENISA (The European Network and Information Security Agency) (December 2011)
[4] Official Journal of the European Communities, Directive 2002/21/EC of the European Parliament and of the Council of 7 March 2002 on a common regulatory framework for electronic communications networks and services (Framework Directive) (2002)
[5] Federal Communications Commission, Telecommunications Act of 1996

[6] Alter, S.: Defining Information Systems as Work Systems: Implications for the IS. European Journal of Information Systems 17(5), 448–469 (2008)
[7] ISO/IEC 42010, Systems and software engineering – Recommended practice for architectural description of software-intensive systems. Geneva: International Organization for Standardization (2007)
[8] Dubois, É., Heymans, P., Mayer, N., Matulevičius, R.: Intentional Perspectives on Information Systems Engineering. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 289–306. Springer, Heidelberg (2010)
[9] The Open Group, TOGAF Version 9.1. Van Haren Publishing, The Netherlands (2011)
[10] ISO/IEC 27005, Information technology – Security techniques – Information security risk management. Geneva: International Organization for Standardization (2011)
[11] ANSSI, EBIOS 2010 - Expression of Needs and Identification of Security Objectives. France (2010), http://www.ssi.gouv.fr/en/the-anssi/publications-109/methods-to-achieve-iss/ebios-2010-expression-of-needs-and-identification-of-security-objectives.html
[12] ISO/IEC 27002, Information technology – Security techniques – Code of practice for information security management. Geneva: International Organization for Standardization (2005)
[13] Mayer, N.: A Cluster Approach to Security Improvement according to ISO/IEC 27001. presented at the Software Process Improvement, 17th European Conference, EuroSPI 2010 (2010)
[14] TMForum, "TM Forum - eTOM Business Process Framework, http://www.tmforum.org/BusinessProcessFramework/1647/home.html (accessed: February 11, 2013)
[15] American Productivity & Quality Center (APQC) and IBM, "Telecommunication Process Classification Framework (November 2008)
[16] TMForum, "TMForum Frameworx - SID Service Overview," GB922-4SO (2011)
[17] ITU (International Telecommunication Union), "ITU-T X.1057 Asset Management Guidelines in Telecommunication Organizations," Recommendation ITU-T X.1057 (2011)
[18] Marinos, L., Sfakianakis, A.: ENISA Threat Landscape - Responding to the Evolving Threat Environment. In: ENISA (The European Network and Information Security Agency) (September 2012)
[19] Ministerio de Hacienda y Administraciones Públicas, "MAGERIT - versión 3.0 - Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información - Libro II: Catálogo de Elementos" (October 2012)
[20] National Institute of Standards and Technology, "NIST Special Publication 800-30 Guide for Conducing Risk Assessments," (September 2012)
[21] Alberts, C., Dorofee, A.: OCTAVE Threat Profiles. Software Engineering Institute. Carnegie Mellon University, White paper
[22] Bundesamt für Sicherheit in der Informationstechnik, "Supplement to BSI-Standard 100-3, Version 2.5 - Application of the Elementary Threats from the IT-Grundschutz Catalogues for Performing Risk Analyses," Federal Office for Information Security, Bonn, Germany (August 2011)
[23] Collier, M.D.: Enterprise Telecom Security Threats (2004)
[24] ITU (International Telecommunication Union), "ITU-T X.1055 - Risk management and risk profile guidelines for telecommunication organizations" (November 2008)

# A Methodology for Software Process Improvement Roadmaps for Regulated Domains – Example with IEC 62366

Derek Flood, Fergal McCaffery, Valentine Casey, and Gilbert Regan

Dundalk Institute of Technology, Ireland
{Derek.flood,fergal.mccaffery,val.casey,gilbert.regan}@dkit.ie

**Abstract.** Software process improvement initiatives offer many benefits in terms of productivity, cost savings and quality. As part of these initiatives organisations undergo an assessment and then embark on a software process improvement program to improve their existing processes to meet a desired target. These programs can be improved by the use of process improvement roadmaps that are tailored to the organisation and are usually non-transferrable. Within regulated domains, such as the medical device industry, adherence to international standards must be achieved before products can be placed on the market. This work proposes the use of software process improvement roadmaps to assist organisations achieve compliance with medical device standards. These proposed roadmaps will be generic in nature to meet the requirements of the standard, but will be subsequently tailored to meet the specific requirements of an individual organisation. In this paper we introduce the concept of the software process improvement roadmaps for the implementation of standards and detail a methodology for developing these roadmaps.

**Keywords:** Software Process Improvement, Software Process Improvement Roadmaps, Medical Device software, IEC 62366.

## 1 Introduction

As long ago as the early to mid-nineties the benefits of software process assessment and improvement and its impact on product quality have been identified and documented in the literature [1,2,3,4]. Research in this area has continued [5] and is best summarised by Paulish and Ebert [6] as: "*with increasing process maturity – which is an investment in process improvement – there is a tangible business impact in terms of reduced cost of quality and less delays. With such data being available from different organisations it is fair to state that – if done well – process improvement has a strong business impact with sustainable ROI (Return On Investment)*". In these circumstances it is not surprising that many organisations undertake Software Process Improvement (SPI) initiatives to improve their processes thereby increasing the quality of their product and the efficiency of its development.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 25–35, 2013.

In highly regulated domains, such as the medical device industry, organisations must demonstrate the quality and safety of their products before they can be placed on the market. Regulatory bodies, such as the Food and Drug Administration (FDA) in the United States (US), regulate these organisations by auditing their development processes. To assist medical device organisations ensure the quality of their products and achieve approval to market their devices, regulatory bodies provide regulations, guidance documents and standards which outline what is required to be compliant.

Adherence to these standards can be difficult for organisations entering the medical device domain due to the lack of specific guidance on their implementation. Regulations, standards and guidance documents outline what needs to be done in order to achieve compliance, but they do not specify how this is to be achieved. Instead the regulations, standards and guidance documents allow organisations to decide on the best method for implementation.

In this work we propose to address this issue through a series of software process improvement roadmaps. These will provide guidance to organisations for adopting specific medical device standards, such as IEC 62304:2006 [7], ISO 13485:2003 [8], ISO 14971:2007 [9], and IEC 62366:2008 [10]. The proposed roadmaps do not assume that any existing processes are in place, allowing for a complete implementation of the standard for organisations with no existing processes.

For organisations that have already some processes in the place, the roadmap can assist them in implementing the remaining requirements of the standard. Through an initial assessment it will be possible to determine what aspects of the standard are already in place and then provide a detailed roadmap on those aspects of the standard that need to be implemented and how these should be applied within the organisation.

The remainder of this paper is structured as follows: Section 2 outlines the role of software within the medical device industry and the importance of standards within this domain. Section 3 then introduces the software process improvement roadmap structure while Section 4 details the methodology used for developing such a roadmap. Section 5 illustrates the methodology through the development of a roadmap for IEC 62366 compliance before the paper is concluded in Section 6.

## 2 Related Work

### 2.1 Software Process Improvement Initiatives

There are many reasons why organisations may choose to undertake SPI initiatives. Studies have shown that SPI can offer a high return on investment in the form of productivity gains, reduced time to market and fewer defects reported by customers [1, 2, 3, 4, 5, 6, 11]. In addition to the benefits outlined above, there are many examples of specific successes achieved by organisations undertaking SPI initiatives. Through peer reviews of software requirements to detect defects prior to coding, one organisation was able to reduce the time spent on rework during the coding phase. In another organisation, improved configuration management practices allowed staff to replicate many errors encountered in the field, reducing the time and expense required to resolve problems [11].

The Software Engineering Institute has set out a roadmap for the undertaking of a software process improvement initiative [12]. This report identifies three main phases in which the software process improvement initiative should progress through. The first phase is to initiate the process improvement initiative which involves learning about SPI, committing initial resources and building a process infrastructure.

The next phase is to baseline the current state of the organisations software processes. This is achieved through the undertaking of a software process improvement assessment, such as ISO/IEC 15504-5:2012 [13] (SPICE) or Capability Maturity Model® Integration (CMMI®) [14]. During an assessment the organisations current processes are assessed and measured, and any weakness or shortcomings are identified. Both ISO/IEC 15504-5 and CMMI® contain capability levels which can allow an organisation to quantify the current state of their processes. These levels also facilitate the setting of targets which the organisation can reach through its process improvement initiative.

The final phase of the software process improvement initiative is to implement or deploy the software process improvements. This stage involves the identification of suitable methods for improving the software processes by addressing the weakness and shortcomings identified during the assessment and then implementing them within the organisation.

Software process improvement is not an overnight activity. It takes long-term commitment from all employees of the organisation, especially senior management, who must provide adequate resources for the implementation of the software process improvement [15]. In describing a usability maturity model developed by Nielsen, the Healthcare Information and Management Systems Society (HIMSS) usability task force noted that it can take decades to reach full maturity [16].

### 2.2 Software Process Improvement within the Medical Device Domain

As regulatory bodies only outline the regulatory requirements which must be complied with and not how they can be effectively achieved, medical device organisations have been compliance centric in their approach to software development. As a result, there has been very limited adoption of software process improvement within the medical device domain [17].

In addition existing generic SPI models, such as the CMMI® [14] and ISO/IEC 15504-5:2012 [13] (SPICE), do not provide sufficient coverage to achieve medical device regulatory compliance [18]. To address this issue a medical device specific SPI framework, titled Medi SPICE, is being developed [29].

The objective of undertaking a Medi SPICE assessment is to determine the state of a medical device organisation's software processes and practices, in relation to regulatory requirements and best practices with the goal of identifying areas for undertaking process improvement [18]. It can also be used as part of the supplier selection process when an organisation wishes to outsource or offshore part or all of their medical device software development to a third party or remote division [19].

Medi SPICE is based on ISO/IEC 15504-5:2012 [13], IEC 62304:2006 [7] and ISO/IEC 12207:2008 [20]. It is being developed in line with the requirements of

ISO/IEC 15504-2:2003 [21] and contains a Process Reference Model (PRM) and Process Assessment Model (PAM). It also incorporates the requirements of the relevant medical device regulations, standards, technical reports and guidance documents.

The Medi SPICE PRM consists of 44 processes and 15 subprocesses which are fundamental to the development of regulatory compliant medical device software. Each process has a clearly defined purpose and outcomes that must be accomplished to achieve that purpose.

Medi SPICE also contains a PAM which is related to the PRM and forms the basis for collecting evidence that may be used to provide a rating of process capability. This is achieved by the provision of a two-dimensional view of process capability. In one dimension, it describes a set of process specific practices that allow the achievement of the process outcomes and purpose, defined in the PRM; this is termed the process dimension. In the second dimension, the PAM describes capabilities that relate to the process capability levels and process attributes. This is termed the capability dimension [22].

### 2.3 Medical Device Regulations, Standards and Guidance Documents

In order to market a medical device within the European Union (EU), the medical device organisation must demonstrate that they are compliant with the regulations set forth by the EU to receive the CE Mark. Similarly, to market medical devices within the US the organisation must demonstrate compliance with the FDA regulations [23]. In order to help organisations achieve compliance with these regulations the EU and FDA have published guidance documents and also recommend compliance with harmonised or approved consensus standards. Medical device organisations may not follow these guidelines and standards and still achieve approval to market their device; however they must provide strong justification for not doing so.

One of the most fundamental requirements of a medical device organisation to achieve regulatory compliance is the implementation of a Quality Management System (QMS). A QMS ensures that the processes used during the development and production of a medical device are defined and monitored to ensure high quality products are developed. The requirements of a quality management system for medical devices have been outlined in ISO 13485:2003 [8]. This standard is harmonised in the EU with the Medical Device Directive (MDD) [24] and has recently been accepted by the FDA as adequate fulfilment of the requirements of a QMS.

As part of the QMS, organisations must perform risk management activities. To improve the quality of the medical devices and receive regulatory approval, the organisation should identify all possible risk and take appropriate action to help mitigate them. ISO 14971:2007 [9] describes the requirements of a risk management process for medical device development. This standard identifies 6 key stages; Risk Analysis, Risk Evaluation, Risk Control, Evaluation of overall residual risk acceptability, Risk Management Report, and Production and Post-Production information.

IEC 62304:2006 – Medical device software – Software life cycle processes [7], provides specific guidance on the processes to be performed for the development of medical device software. This is an EU harmonised standard and is recognised by the

FDA as an approved consensus standard. It is therefore used to develop medical device software for both the European and US markets as well as many other countries.

In 2007 the European Council amended the MDD [24], which governs the approval and marketing of medical devices in the European Union (EU). This amendment came into effect in March of 2010. As part of this amendment the EU recognized the importance of software and revised the directive to include the provision that software can now, in its own right, be classified as a medical device. As a result software can now be subjected to the same regulations and standards as other medical devices [25]. This means that some organisations that develop medical related software may now be developing medical devices and as such must adapt their processes to meet the requirements of the medical device standards outlined above.

# 3 SPI Roadmaps Towards Standards Compliance

## 3.1 Roadmap Structure

To assist medical device software development organisations achieve compliance with the required standards, we propose the development of a set of software process improvement roadmaps. For the purposes of this work we define a roadmap as: *A series of milestones, comprised of goals, that will guide an organisation, through the use of specific activities, towards compliance with regulatory standards.*

The roadmap is divided into two levels. The first level defines the goals, grouped into milestones, that the organisation should achieve throughout the SPI initiative. The first level of the roadmap is presented at a high level and does not contain any detail relating to how the goals should be achieved. This is done for two reasons. Firstly, by presenting the roadmap as a series of goals traceability to the relevant standard can be easily achieved. Secondly, the high-level roadmap can form a basis for communication across the industry as the same high-level roadmap can be applied to all organisations.

The second level roadmap contains specific guidance for organisations on how to achieve the goals outlined in the high level roadmap. The activities preformed, to meet the goals of the high level roadmap, can vary from organisation to organisation due to their nature, different abilities and resources. Each roadmap is comprised of multiple activities that can achieve each goal so that the most suitable activity can be presented to an organisation wanting to implement the roadmap.

## 3.2 Roadmap Implementation

The first stage in using the proposed roadmaps is to assess the organisations existing processes and to determine which goals they already meet. This can be done in a number of ways, including the use of existing process assessment models, such as Medi SPICE or assessment models developed from the standards through the transformation method presented in [26].

The next stage is to identify which goals the organisation needs to achieve in order to meet the requirements of the relevant standard. Due to the traceability provided by the roadmap development methodology, it is easy for an organisation to see which aspects of the standard are not being met.

Once the goals to be implemented have been identified, the next step will be to identify the relevant activities that will satisfy these goals. The identification of the correct activities will be based not only on the goals to be achieved, but also on the organisation itself. Factors, such as the organisation's size, the class of medical device being developed, and the distribution of the software development team or teams, can all impact the way in which an organisation will implement the standard.

Once the appropriate activities have been identified for the organisation, they will begin to implement these activities. The roadmap defines specific milestones in a progressive order that will guide this implementation.

## 4 Roadmap Development Methodology

The following methodology is proposed to provide a systematic approach to roadmap construction. This systematic approach will allow for other researchers to construct roadmaps for other regulated domains, such as the automotive domain or the aerospace domain. The following approach is similar to the transformation method presented in [26] for the construction of ISO/IEC 15504-2 compliant process assessment and process reference models.

There are a range of research methods that can be used with the following methodology. These techniques are used to validate the roadmap and to assist in the identification of a wide range of activities. By incorporating a wide range of activities, the generated roadmaps can provide more suitable guidance to implementing organisations.

The methodology used for the development of the roadmaps is as follows:

1. **Identify requirements of the standard.** The first step in the process of developing the roadmap is to identify all of the required activities of the standard. This step is similar to the first step in the transformation process presented in [26].
2. **Logically group all requirements.** The next step is to group the requirements. Requirements can be grouped based on the stage of the software development lifecycle at which they will occur. Some activities are performed throughout the lifecycle, independent of specific phases. In those cases these activities should be grouped together and placed at or before the first stage at which they are performed in the software development lifecycle.
3. **Separate grouped activities in line with ISO/IEC 15504 capability levels.** Once the requirements have been grouped, these groups should be separated based on the capability level at which the requirements should be performed. These groups form the milestones of the roadmap
4. **Order the milestones based on the capability level and logical groups.** All level 1 milestones should be implemented first in the order in which they will occur in the development process, followed by all Level 2 activities, and subsequently by all Level 3 activities until all of the milestones are in order.
5. **Validate generated roadmap.** The generated roadmap should be validated with industry experts. These experts could be individuals working in industry

implementing the standards, assessors regulating organisations using the standards or academics with the appropriate expertise. Members of the standards committee could also assist with the validation.

There are a number of methods that could be used to validate the roadmap. One approach could be to interview the experts after presenting the roadmap to them and providing sufficient time for them to review the material. Another approach could be a workshop in which the roadmap is presented to the experts and then a panel discussion is used to identify and rectify issues that may be present. A Delphi study could also be used.

A Delphi study involves multiple iterations and review by experts. In this case the experts are first asked to complete a questionnaire about the roadmap. Once the responses have been analysed, the roadmap is then revised and resubmitted to the experts for a subsequent review. This is repeated until a roadmap is agreed upon.

The validation should aim to ensure that:

- The goals are correctly grouped;
- The milestones are in the correct order for implementation; and
- The roadmap incorporates all aspects of the standard.

6. **Identify activities that can meet the identified goals**. The next step in the generation of the roadmap is to identify appropriate activities that can be used to fulfil the requirements of each goal in the roadmap. This can be done through a systematic literature review and/or case studies with organisations already implementing the standard.
7. **Validate activities in host organisation**. The final stage of the roadmap development methodology is to validate the roadmap within a host organisation. This will involve the generation of a roadmap for the host organisation and then undertaking a software process improvement initiative to implement the roadmap.

To date this work has developed high-level roadmaps for each of the standards; ISO 14971, ISO 13485 and IEC 62366. The following section will show how the above methodology has been applied during the development of a software process improvement roadmap for compliance with the IEC 62366 standard, which details the application of usability engineering to medical devices.

## 5 Roadmap to IEC 62366 Compliance

In four US hospitals more than 300 patients were over radiated by powerful CT scanners used to detect strokes and which had obtained FDA approval. One hospital, which detected the errors after 18 months when patients started losing their hair, found that the overdose was displayed on-screen however the technicians administering the scans did not notice it [27].

Similarly during an analysis of infusion pumps recalled by the FDA between 2005 and 2009, user interface errors were identified as one of the most common cause of the recalls [28]. It was found that on some devices the screen failed to make clear the units of measurement (pounds vs. kilograms) when entering patient data for calculating the dosage, leading to incorrect dosages being applied.

One way to reduce the likelihood of these errors occurring is through the use of usability engineering techniques. This is addressed by the international standard *IEC 62366:2007 – Medical Devices – Application of usability engineering to Medical Devices* which should be utilised during the implementation of a usability engineering process. This standard specifies "*a process for a manufacturer to analyse, specify, design, verify and validate usability, as it relates to safety of a medical device*" [10]

The standard places a strong focus on the identification and elimination of risks associated with the use of the medical device. As part of the usability engineering process, the standard highlights the importance of the identification of Hazards and Hazardous situations, a critical component of the risk management process. The standard (in Section 5.7 Note 2) also recommends an iterative development cycle, specifying the need to perform usability validation throughout the design and development of the medical device.

As part of the usability engineering process, IEC 62366 specifies the need to perform usability verification, ensuring the user interface meets the requirements of the usability specification, and usability validation, ensuring that the primary operating functions can be accomplished through the user interface.

The standard not only requires usability to be incorporated into the medical device, it specifies that usability engineering should also be applied to the development of the user manual and other supporting documentation as well as to the training of users in the use of the medical device and all material necessary to support this training.

Due to the importance of usability within the medical device domain, and the risks associated with the misuse of medical devices, this work has developed a software process improvement roadmap for the implementation of the IEC 62366 standard. This roadmap has been developed and is currently being validated by industry experts.

The first stage of the methodology is to identify the requirements of the standard. When this was performed on the IEC 62366 standard, 44 requirements were identified. These requirements were used as the basis for defining the goals of the roadmap. The following are example requirements taken from the standard.

- *Identify the frequently used functions that involve user interaction*
- *Identify the characteristics that relate to safety and focus on usability*
- *Design and implement the User Interface as described in the usability specification*

When all of the goals were identified, they were then grouped. During this stage, the goals were arranged into 9 groups, which represented the main components of the standard: verification, validation, training, documentation, implementation, the usability process, risk management, task orientated activities, and usability specification. Each group contained between 2 and 7 goals.

Once this was complete, the goals in each group were reviewed and associated with a capability level as defined in ISO/IEC 15504-2. It was found that 37 of the goals would be achieved at level 1 of the ISO/IEC 15504-2 capability level rating, while the remaining goals (7 goals) would occur at level 2. Based on this the groups were redefined, resulting in the identification of 10 milestones.

The final stage of the high-level roadmap generation is the ordering of the milestones. This was done based on the initial groupings and the capability levels defined in the previous step. The milestones containing level 1 goals were arranged in the order they would be performed in a typical software development iteration. Subsequently all milestones containing level 2 goals were arranged in the same order following all level 1 milestones. The resulting roadmap for IEC 62366 is as follows (Table 1):

**Table 1.** Roadmap for IEC 62366

| Step Number | Milestone Title | # of Goals |
|---|---|---|
| 1 | Task | 5 |
| 2 | Usability Specification | 5 |
| 3 | Risk Management | 7 |
| 4 | Implementation | 2 |
| 5 | Documentation | 6 |
| 6 | Training | 4 |
| 7 | Verification | 4 |
| 8 | Validation | 4 |
| 9 | Validation Management | 3 |
| 10 | Process | 4 |

The next step in this work will be to validate the above roadmap using industry experts. In order to do this a Delphi research method has been chosen. The Delphi research method allows for multiple reviews of the roadmap until the experts agree on a correct order of implementation for the roadmap. To perform the study experts will be asked to fill in an online questionnaire asking them questions relating to the order of the milestones, the appropriateness of each goal in each milestone and the completeness of the roadmap to meet all requirements of the standard.

## 6 Conclusions

Entering regulated domains, such as the medical device industry, is a difficult task due to the high level of regulations that must be adhered to. Regulations, standards and guidance documents outline what the organisation must do in order to achieve regulatory compliance; however these documents do not specify how the organisation should achieve it. This places additional stress and can be seen as a barrier for organisations wishing to enter the medical device domain. To address this issue, in the context of medical device software development, this paper presents a methodology for the development of software process improvement roadmaps for the implementation of medical device standards. The roadmaps presented are divided into two levels. The high-level roadmap outlines the main goals to be met to achieve regulatory compliance while the low level roadmap provides specific guidance on how to implement the processes necessary to meet these goals. The presented methodology allows researchers to generate such roadmaps directly from standards and guidance documents released by regulatory bodies. Although the approach requires substantial effort the resulting roadmap will benefit a large number of organisations and provide a foundation on which to build a comprehensive knowledgebase on software processes for regulatory compliance.

**Acknowledgement.** This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 10/CE/I1855.

## References

1. Humphrey, W.S., Snyder, T.R., Willis, R.R.: Software process improvement at Hughes Aircraft. IEEE Software 8(4), 11–23 (1991)
2. Dion, R.: Process Improvement and the Corporate Balance Sheet. IEEE Software 10(4), 28–35 (1993)
3. Herbsleb, J., Carleton, A.: Rozum J., Siegel J., Zubrow D, Benefits of CMM-Based Software Process Improvement: Executive Summary of Results. Software Engineering Institute (1994)
4. Clarke, P., O'Connor, R.V.: Business success in software sMEs: Recommendations for future SPI studies. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 1–12. Springer, Heidelberg (2012)
5. Harter, D.E., Krishnan, M.S., Slaughter, S.A.: Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. Management Science 46(4), 451–466 (2000)
6. Paulisch, F., Ebert, C.: Business Impact of Process Improvements Workshop ICSE Companion 2008, Companion of the 30th International Conference on Software Engineering, pp. 1073–1074 (2008)
7. IEC 62304:2006, Medical device software—Software life cycle processes, Geneva, Switzerland, IEC (2006)
8. ISO 13485:2003, Medical devices — Quality management systems — Requirements for regulatory purposes, 2nd edn., Geneva, Switzerland, ISO (2003)
9. ISO 14971 – Medical Devices – Application of risk management to medical devices, Switzerland, ISO (2007)
10. IEC 62366:2007 "Medical Devices – Application of usability engineering to medical devices"
11. Ziehe, T., Wohlwnd, H., Gettel, G., McGowan, D.: "Software Process Improvement (SPI) guidance for Improving software: Release 4.0" SEMATECH report. Technology Transfer # 95082943A-ENG (October 31, 1995)

12. McFeeley, R., McKeehan, D.: Software Process Improvement Roadmap, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, User's Guide CMU/SEI-95-UG-001 (1995), `http://www.sei.cmu.edu/library/abstracts/reports/95ug001.cfm`
13. ISO/IEC 15504-5:2012, Information technology - Process Assessment - Part 5: An Exemplar Process Assessment Model. Geneva, Switzerland, ISO (2012)
14. CMMI Product Team, Capability Maturity Model® Integration for Development Version 1.2. Software Engineering Institute, Pittsburch PA (2006)
15. Casey, V., Richardson, I.: A Practical Application of the IDEAL Model. Software Process Improvement and Practice 9(3), 123–132 (2004)
16. 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)
17. Denger, C., Feldmann, R., Host, M., Lindholm, C., Shull, F.: A Snapshot of the State of Practice in Software Development for Medical Devices. In: First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain (2007)
18. McCaffery, F., Dorling, A.: Medi SPICE Development. Software Process Maintenance and Evolution. Improvement and Practice Journal 22, 255–268 (2010)
19. Casey, V.: Virtual Software Team Project Management. Journal of the Brazilian Computer Society 16, 83–96 (2010)
20. ISO/IEC 12207:2008, Systems and software engineering - Software life cycle proc-esses. Geneva, Switzerland, ISO (2008)
21. ISO/IEC 15504-2, - Software engineering — Process assessment — Part 2: Performing an assessment. 2003: Geneva, Switzerland (2003)
22. Casey, V., Mc Caffery, F.: Medi SPICE and the development of a Process Reference Model for inclusion in IEC 62304. In: ICSOFT 2012 – The 7th International Conference on Software Paradigm Trends, Rome, Italy, July 24-27 (2012)
23. Burton, J., Mc Caffery, F., Richardson, I.: A risk management capability model for use in medical device companies. In: International Workshop on Software Quality (WoSQ 2006). ACM, Shanghai (2006)
24. European Council, Council Directive 2007/47/EC (Amendment). Official Journal of The European Union: Luxembourg (2007)
25. McHugh, M., McCaffery, F., Casey, V.: Standalone Software as an Active Medical Device. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A., et al. (eds.) SPICE 2011. CCIS, vol. 155, pp. 97–107. Springer, Heidelberg (2011)
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 2000. In: SPICE 2008, Nuremberg, Germany (2008)
27. Bogdanich, W.: As Technology Surges, Radiation safeguards lag, January 26. New York Times (2010)
28. Food and Drugs Administration, Infusion Pumps Improvement Initiative (2010), `http://www.fda.gov/medicaldevices/productsandmedicalprocedures/GeneralHospitalDevicesandSupplies/InfusionPumps/ucm205424.htm` (accessed: December 07, 2012)
29. Casey, V., McCaffery, F.: Development of the Medi SPICE PRM. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 265–268. Springer, Heidelberg (2012)

# Simulating Global Software Development Processes for Use in Education: A Feasibility Study

Miguel J. Monasor[1,2], Aurora Vizcaíno[2], Mario Piattini[2], John Noll[1], and Sarah Beecham[1]

[1] Lero, The Irish Software Engineering Research Centre, University of Limerick, Limerick, Ireland
MiguelJ.Monasor@gmail.com, {John.Noll,Sarah.Beecham}@lero.ie
[2] Alarcos Research Group, Institute of Information Technologies & Systems, Escuela Superior de Informática, University of Castilla-La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, Spain
{Aurora.Vizcaino,Mario.Piattini}@uclm.es

**Abstract.** VENTURE is a simulation-based training platform aimed at helping practitioners overcome process problems that arise in Global Software Development (GSD). VENTURE places practitioners in simulated GSD scenarios in which they play a role and interact with Virtual Agents, who represent team members from different nationalities. VENTURE makes it possible to simulate cultural, linguistic and GSD procedural problems gathered from experience and empirical studies. This paper reports on a Feasibility Study aimed to determine the potential of VENTURE to: 1) simulate GSD scenarios and processes of potential conflict, and 2) train practitioners to cope with these conflicts by interacting with virtual agents. A group of researchers and experts studied the platform and, through a survey-based method, they provided their endorsement of the concept. We received positive feedback and encouragement, in that the simulation of GSD processes will effectively provide training in industrial settings, helping practitioners to identify and resolve predefined problems.

## 1 Introduction

Global Software Development (GSD) implies new challenges for practitioners who have to collaborate with distant team members from a variety of cultures using a common language [1]. These challenges may often remain hidden until it is too late, and if ignored, may adversely affect the project and even lead to failure [2]. In traditional co-located development, work productivity is affected by social and individual factors, as well as by cooperation among software development teams [3]. However, in GSD these factors have even more impact, as cultural differences must also be taken into account [4]. To be effective therefore, GSD process development models need to consider aspects such as the employee's culture, and how the team will communicate and collaborate, as well as develop and maintain the team's common goal orientation [5-7].

Process improvement initiatives and techniques have been introduced to aim to identify specific software process improvement needs, e.g. Raninen et al. [8]. Training is a key factor for process improvement [9] as participants are required to develop specific

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 36–47, 2013.

competences and skills, that depend on their role in the software process [10]. For example, an e-learning experience to provide coaching in the use of best practices is presented by Messnarz et al. [11], and Cos et al. [12] explores how to adapt a e-learning platforms to fit international settings.

GSD introduces the need for employees to develop new competencies concerned with overcoming cultural and language barriers. Moreover, team members must be made aware of the additional interaction difficulties that arise as a result of the global distribution of work, new processes applied and time zone differences. The need for these new competencies calls for new training initiatives. However, providing training in the specific problems of GSD is not easy, due to the complexity of reproducing real environments in educational settings; this requires infrastructure, time and expertise [13]. Applying appropriate training requires some evaluation of the gaps in knowledge and specific needs of a particular student[1]. However, this evaluation of the student's current level of understanding is also difficult. As a result, the limited training in GSD tends to be classroom-based or paper-based [13], [14].

In this paper we introduce a new training initiative "VENTURE" (Virtual ENvironment for Training cUlture and language problems in global softwaRe dEvelopment) [15]. VENTURE is a simulation-based interactive training platform that aims to support practitioners with their culture and language process issues in GSD. In VENTURE's simulated GSD setting, students interact with Virtual Agents, who represent different cultures. The focus of this paper is to report how these aims are tested through a feasibility study conducted with a group of potential users.

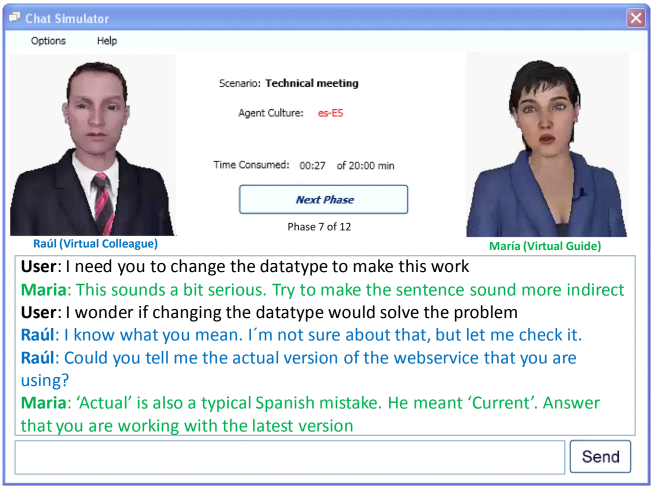



**Fig. 1.** VENTURE's chat simulator interface

Figure 1 shows VENTURE's chat simulator interface. This platform is intended to provide training in specific process model practices, GSD skills and GSD scenarios that reflect real interactions. A Virtual Guide ('Maria' in Figure 1) directs the conver-

[1] The term "student" is used to represent the role of 'practitioner, 'learner', 'trainee' or 'user' (as in Fig 1).

sation between the student and the virtual colleague (Raúl). Should the student make any errors in the conversation, the Virtual Guide will corrects the student in real-time. In this way, the student is trained in how to use negotiation and reward structures and in how to avoid conflict or generate trust. Supplementary information on cultural principles and known language problems are also given. For a detailed description of VENTURE go to http://global.lero.ie/venture.

### 1.1 Feasibility Study Objectives

The main objective of conducting the feasibility study was to obtain prompt feedback about the suitability of VENTURE for training Global Software Development (GSD) skills. The feasibility study results reported here are based on feedback from a sample of eight researchers and four practitioners with expertise in GSD. The results showed that the VENTURE platform has the potential to be engaging and useful for training in GSD processes, as long as the training scenarios are designed in a dynamic way, and provided the scenarios present truly representative situations. In response to the positive encouragement received, we plan to develop a revised platform that incorporates some new user requirements as suggested by our sample of experts. This feasibility study addresses four key research questions:

**RQ1:** Does the proposed tool help participants to develop skills needed in GSD?
**RQ2:** How should this tool be applied in educational settings?
**RQ3:** Would the tool be usable and effective for training purposes?
**RQ4:** What kinds of scenarios are suited to such a tool?

## 2 Methodology

The research methodology to date involves three phases: first we conducted a literature review to explore existing approaches to GSD training, and gain an understanding of their strengths and weaknesses. The results of the literature review, as reported in Monasor et al. [13] motivated the second phase of research: to develop a training platform that aims to address the identified gap in GSD training support [15]. Phase three of the method involves conducting a feasibility study of the suitability of the platform to meet the training needs of GSD practitioners and GSD instructors. It is this third phase of the research that is reported in this study.

To answer our research questions (as listed in section 1.1) we apply a survey methodology that includes a mixture of open and closed questions (see Appendix A). The survey method was selected as a good way to elicit a participant's own opinions and ideas without them being influenced by others taking part [16]. Since some participants involved in this study assessed the training tool as part of a group demonstration, this method appears appropriate. Prior to answering the survey questions, participants were given an explanation of VENTURE's architecture and operation. To illustrate the chat simulator's operation, participants were shown a series of snapshots. In these snapshots, a simulated Spanish student, playing the role of software analyst, chatted with a virtual customer in the elicitation of a set of software requirements. Several stages of the simulation were displayed, showing how the simulator presents cultural and linguistic differences with the help of the Virtual Guide. In this way, interviewees were able to form an impression on how the simulator operates. After the briefing, participants completed a survey consisting of structured questions that aim to answer our research questions. The questionnaire comprised yes/no closed questions (to gather data on experience and personal details), and open-ended questions (to elicit opinions about the use of the platform in their companies/universities; i.e. the platforms usefulness and usability). It was intended that an analysis of these responses would act as a guide to refine the next phase of tool design.

An opportunistic sample of four practitioners from three multinational companies participated in the survey, where selection was based on their expertise in GSD and their availability. As shown in Table 1, two participants were project managers, and two were developers. Experience in GSD projects ranged from three to eight years.

**Table 1.** Characteristics of the practitioners interviewed practitioners

| # | *Nationality* | *Age* | *Experience in GSD (years)* | *Current role* |
|---|---|---|---|---|
| **1** | Spain | 36 | 5 | Project Manager |
| **2** | Spain | 38 | 8 | Project Manager |
| **3** | Spain | 34 | 6 | Developer |
| **4** | U.S.A. | 35 | 3 | Developer |

**Table 2.** Background of researchers

| # | *Nationality* | *Age* | *Previous Knowledge in GSD* |
|---|---|---|---|
| **1** | Cuba | 26 | No |
| **2** | Italy | 27 | Medium |
| **3** | Mexico | 42 | Medium |
| **4** | Peru | 35 | Medium |
| **5** | Argentina | 26 | No |
| **6** | Spain | 27 | Advanced |
| **7** | Spain | 38 | No |
| **8** | Uruguay | 27 | Basic |

In addition, a group of eight researchers also completed the survey. Table 2 shows the characteristics of these researchers. As a whole, this population reflects the needs of end-users: practitioners can give their opinion from the perspective of the group designing and delivering the courses (allowing us to evaluate the scenario designer; and the applicability of the platform in industrial settings).

# 3 Results

The responses of the survey are analyzed by attending to the research questions:

## 3.1 Does the Proposed Tool Help Participants to Develop Skills Needed in GSD?

This question was addressed by the following points:

### 3.1.1 Usefulness of the Tool for Training in the Skills Required in GSD

In the words of a respondent: “It is very useful, because in a real project problems will never happen in a systematic, controlled way; with this environment, you can model many real situations and improve how you deal with them”. In general, researchers liked the idea that the tool can provide independent, customized training that focuses on specific problems, and also the idea of having a repository of predefined rules and scenarios. In this regard, two respondents suggested that a key factor for the platform to be successful would be for it to make available a large repository of different training scenarios, as well as a wide set of rules to ease the design of new scenarios based on specific problems.

#### 3.1.2 Weaknesses of the Tool and Ideas for Improvement

Respondents pointed out that real life is very complex; the training scenarios can represent only a small example of the problems that may arise. Moreover, in order to provide training adapted to each individual, with a specific student’s needs in mind, it is necessary to provide a sufficient number of training scenarios, which is not easy. The time required to design training scenarios is seen as being a very important aspect to take into account in the success of the tool. This suggests that the usability and flexibility of the scenario designer component are critical.

Respondents also believe that in the initial training stage, an explanation should be given on how the tool works to make them aware of the mechanism: “Users could lose interest when speaking to a machine. If this happens, they will make less of an effort. It is important for users to be aware of this, so that they get as much benefit out of the platform as possible”.

Respondents remarked that the simulations could appear artificial if the training scenario is not well designed, leading to a subsequent loss of interest on the part of the students. It was therefore suggested that interesting and fluid scenarios should be created, to mitigate this problem. The following suggestions were also provided:

- Include hyperlinks in the text. Virtual Agents could provide links to documents.
- Integrate the platform in real situations in which the interaction between real participants could be guided or supervised by the Virtual Guide.
- Include function to iteratively improve a scenario after each training cycle.
- Include a function to pause the simulation.

### 3.2 How Should This Tool Be Applied in Educational Settings?

The following points address this question:

#### 3.2.1 Applicability of the Tool in University Classes

In general, researchers considered VENTURE suitable for training certain concepts. They agreed that a deeper, corresponding theory should be provided in traditional classes to complement the virtual platform training. However, this add-on would depend on the course learning objectives.

#### 3.2.2 Applicability of the Tool to Their Companies

This question was answered only by practitioners, most of whom thought that, once instituted in a company, the platform would be useful not only for training in GSD, but also for other kinds of interaction between people, such as customer support. Two practitioners stated that the main problems companies encounter when organizing courses are: the difficulty in finding available experts in GSD; the time needed to develop these courses; and, the organizational difficulties of carrying out the courses in the company. Some practitioners agreed that, with a more complete database of problems and linguistic and cultural rules, this platform could be used, potentially, by inexpert developers. In the words of a practitioner: “In the future, when it has a sufficient amount of information, it will no doubt be a great tool”.

#### 3.2.3 Problems in the Application in the Company

The main concern of practitioners when applying this kind of training is the time and resources needed. Some comments also related to the operation of the platform: “Learning in this way could be kind of artificial, but it gives the user the chance to have experiences that it would otherwise be difficult to have”. “Obviously, a simulation is always different from reality, but in this case you can simulate lots of problems that might well appear in real life”.

#### 3.2.4 Time Required for This Kind of Training

Although there were a variety of responses to this question, practitioners seem to agree that two sessions a week would be reasonable. While the duration of the simulation should be close to real conversations, they concluded, taking 20 minutes (the average time suggested in the responses) might be the right length of time for a chat simulation. That could vary depending on the particular scope of the training scenario. One researcher suggested that the scenarios should focus on specific objectives, rather than focusing on the time taken to complete a scenario. So for example, the scenario would not finish until it had reached a certain phase at least, or when it had generated a certain number of mistakes. That said, we determined that time management was an important skill in GSD that we wanted users to learn.

### 3.3 Would the Tool Be Usable and Effective for Training Purposes?

Participants answered this question by highlighting the following three points:

#### 3.3.1 Look and Feel of the Virtual Environment

In general, practitioners and researchers feel that the use of the chat simulator is similar to any other chat application; this is appropriate from the point of view of the user’s experience. One practitioner indicated that when interacting with Virtual Agents, users are not going to react in the exact same way as they do with real people, but at the same time he agreed that Virtual Agents are perfectly valid for teaching purposes.

#### 3.3.2 Time Saving Benefit and Limitation

From the point of view of the instructor, a practitioner noted: “It minimizes the instructor’s workload…, it can reproduce difficult situations…, I think that the tool will be useful as support but the main concepts of the subject must always be taught by a teacher". Moreover, as regards the scenario designer, a respondent showed concern about the design of the scenarios: "The main problem is how to adapt the tool so that it provides suitable suggestions and feedback to the user".

#### 3.3.3 Engagement and Motivation

Researchers were asked about their interest in using the tool and how motivated they would be to do so. This question was intended to get feedback that could help to identify aspects of the simulator that needed to be improved if it was to be accepted more readily and completely. Most of the researchers valued the platforms ease of use for independent training. As the tool responds to a real training need in a practical way, they found engaging with the tool instructive and motivational. In their words: "When a developer faces a global project for the first time, he may suffer from stress and fear of failure…", "being able to practice beforehand and learn how to interact can reduce these problems in the initial stages of the project". One respondent also remarked that iterative improvement of the training scenarios would be necessary to create scenarios with enough quality to be attractive to real users.

### 3.4 What Kinds of Scenarios Are Suited to Such a Tool?

The following point provides answers to this question:

#### 3.4.1 Training Scenarios and Skills

Participants were asked which scenarios and skill training they would like added to the current platform. Having only been given an example of the training scenario that consisted of a requirements elicitation meeting, practitioners made the following suggestions for future development:

- Meetings to ask clients for specific information.
- Client support activities. Dealing directly with clients about issues that may arise.
- Interaction with a remote developer to solve a problem with the software.
- Asking an expert about a particular technology in order to solve a problem.
- Asking a client for access to their systems and for details of their requests.
- Provision of training in the use of the specific tools employed in the company.
- Dealing with an angry customer who is concerned about the software.
- Dealing with a colleague who has done a bad job.
- Real cases previously documented by the company.

Researchers, for their part, proposed the following interesting ideas:

- Formal meetings with a manager.
- Informal meetings to exchange information that could be interpreted by the user.
- Resolution of an urgent situation that must be dealt with in a short period of time, where there is no room for mistakes.
- Delivery of software to a client. Providing training and assistance to its users.

- Asking for clarification of requirements to solve a certain problem.
- Discussing a reported error with a client. In some cases these errors are not really errors as such; sometimes these errors are complicated to reproduce.

This list clearly indicates that there is great scope for future development of the tool for new environments and new functionality.

# 4 Discussion

Our aim in this study is to identify whether our proposed VENTURE tool has the potential to fill the gap identified in the literature. The literature calls for the provision of GSD training that is accurate, flexible, and easy for students to engage in and instructors to develop [13]. Conducting the feasibility study reported in this paper helped us to determine whether VENTURE has the potential to meet our aims. The feasibility study was designed to identify the strengths of our current solution and identify how the tool could be improved at this early stage in the development cycle.

## 4.1 Potential Application

The platform was considered by both practitioners and researchers to be applied both in universities and in companies. Respondents agreed that the kind of training provided by this proposal can help to focus the training on specific objectives that can indeed be systematically reproduced in an accurate manner. Using both researchers and practitioners in our feasibility sample proved useful in gaining ideas from a variety of perspectives. For example researchers, in general, liked the idea that the tool can provide independent, customized training that focusses on specific problems, that they could engage in when needed (without having to attend a formal class). As the tool reflects and simulates a real training need, they found the experience instructive and motivational. On the other hand, practitioners took a more pragmatic stance focussing on the time and resources which could be saved through this form of training platform. Compared with the traditional training approaches [13], VENTURE minimizes the instructor's workload and the time required for organizing courses and looking for experts. Both the flexibility of the tool for reproducing different kinds of scenarios and the independence of the training were seen as being of great value. Some of those surveyed even found that the platform would be useful for designing other kinds of interactions that are unrelated to GSD training.

Both groups of participants felt that the virtual platform was familiar, and that the use of the Chat Simulator is similar to any other chat application. However they did note that interacting with a Virtual Agent does have its limitations as users are not going to react in exactly the same way as they would do with real people, although the approach is perfectly valid for teaching purposes.

The main weak points reported were the problem of providing a sufficient number of training scenarios and the anticipated time required to create new training scenarios. Another problem reported by a participant was related to the motivation of the students: they might lose interest when interacting with Virtual Agents. "If this happens, they will make less of an effort. It is important for users to be aware of this, so that they get as much benefit out of the platform as possible". Moreover, the Virtual Guide might be too intrusive on some occasions and stop the flow of conversation. This means that the course designer (instructor) must seek to obtain a careful balance between realism and training scopes, and should give the student just enough guidance to complete the scenario within an appropriate amount of time. Feedback provided by the Virtual Guide must also be carefully planned, in order to avoid disturbing the student too much.

Finally, analysis of the responses to the feasibility study survey provided useful insights that address our research questions. Viewing responses from a group of potential training platform users in terms of current strengths and weaknesses of the concept, will inform the next phase of development. However, having refined the tool; future evaluations should ideally be undertaken in real educational (or industrial) settings with real students and instructors. In this way we will gain a more accurate picture of how effective VENTURE is in improving the skills of the users.

## 5 Limitations

This work has some limitations with regards to construct, internal and external validity of the evaluation [17]:

- **Construct Validity:** There may be some bias in the responses since all the researcher participants were from the same course and university. Also, only one type of research instrument was used: a survey with a limited number of questions.
- **Internal Validity:** There may be some bias, since the participants handed their responses directly to the researchers undertaking and reporting this study.
- **External Validity:** We cannot generalize these results to the wider due to the small sample, which is not necessarily representative of the population of practitioners and researchers likely to use VENTURE. However, the sample does represent a cross section of countries, experiences and user groups.

## 6 Conclusions

This paper describes a feasibility study of VENTURE we conducted to assess whether our simulation tool could potentially provide training in GSD-specific skills. The study also examined whether simulating real processes would give the student increased confidence, and lead them to effectively communicate with participants from different cultures and languages. The analysis of the results obtained indicates that our training platform meets our objectives. Indications are that VENTURE has the potential to give the student increased confidence for effective communication in GSD, and is able to reproduce realistic scenarios and provide feedback that focuses on specific skills. Contentious, sensitive, and error-prone scenarios on specific process areas can be designed based on previous experiences or knowledge. Participants can thus become familiar with these specific processes and problems in a secure off-line way and they can learn how to tackle them effectively, without fear of committing mistakes that could impact adversely on real projects. The feasibility study served to conclude that VENTURE has a potential application in industrial settings, as a tool for improving GSD processes. Our results indicate that we need to add an iterative improvement function to ensure that high quality training scenarios are to be provided. There should also be some effort dedicated to introducing students to the context of the training scenario and the operation of the environment, prior to their first interaction with it. Finally, the feasibility study of our prototype training platform, as reported in this paper, proved useful in highlighting the strengths and weaknesses of the platform and e-learning virtual training concept; it has also shown us the direction for future VENTURE development. The use of a cross-section of participants, all of whom have experience in GSD or in education/training, added particular strength and relevance to our study.

**Acknowledgements.** This work was supported by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie). It has also been funded by the GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01). Support also came from ORIGIN (IDI-2010043 (1-5)) funded by CDTI and FEDER, as well as GLOBALIA (PEII11-0291-5274), Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha.

## References

1. Monasor, M.J., Piattini, M., Vizcaíno, A.: Challenges and Improvements in Distributed Software Development: A Systematic Review. Advances in Software Engineering 2009, 1–16 (2009)
2. Espinosa, J.A., Carmel, E.: The impact of time separation on coordination in global software teams: a conceptual foundation. Software Process: Improvement and Practice 8, 249–266 (2003)
3. Yilmaz, M., O'Connor, R.V.: An Empirical Investigation into Social Productivity of a Software Process: An Approach by Using the Structural Equation Modeling. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 155–166. Springer, Heidelberg (2011)
4. MacGregor, E., Hsieh, Y., Kruchten, P.: Cultural patterns in software process mishaps: incidents in global projects. In: Proceedings of the 2005 Workshop on Human and Social Factors of Software Engineering, pp. 1–5. St. Louis, Missouri (2005)
5. Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R., Koenig, F.: Agility Meets Systems Engineering: A Catalogue of Success Factors from Industry Practice. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 245–256. Springer, Heidelberg (2010)
6. Richardson, I., Casey, V., McCaffery, F., Burton, J., Beecham, S.: A process framework for global software engineering teams. Information and Software Technology 54, 1175–1191 (2012)

7. Noll, J., Beecham, S., Richardson, I.: Global software development and collaboration: barriers and solutions. ACM SIGCSE Bulletin - Special Section on Global Intercultural Collaboration (2010)
8. Raninen, A., Ahonen, J.J., Sihvonen, H.-M., Savolainen, P., Beecham, S.: LAPPI: A light-weight technique to practical process modeling and improvement target identification. Journal of Software: Evolution and Process (accepted)
9. Ringstad, M.A., Dingsøyr, T., Brede Moe, N.: Agile Process Improvement: Diagnosis and Planning to Improve Teamwork. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 167–178. Springer, Heidelberg (2011)
10. Riel, A.: Integrated Design – A Set of Competences and Skills Required by Systems and Product Architects. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 233–244. 99, Heidelberg (2010)
11. Messnarz, R., Bachmann, O., Ekert, D., Riel, A.: SPICE Level 3 - Experience with Using E-Learning to Coach the Use of Standard System Design Best Practices in Projects. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 213–221. Springer, Heidelberg (2010)
12. Cos, J.A., Toval, R., Toval, A., Fernández-Aleman, J.L., Carrillo-de-Gea, J.M., Nicolas, J.: Internationalization requirements for e-learning audit purposes. In: 2012 IEEE Global Engineering Education Conference (EDUCON), pp. 1–6 (2012)
13. Monasor, M.J., Vizcaíno, A., Piattini, M., Caballero, I.: Preparing students and engineers for Global Software Development: A Systematic Review. In: International Conference on Global Software Development (ICGSE 2010), pp. 177–186. IEEE Computer Society, Princeton (2010)
14. Carlson, P., Nan, X.: Experience and recommendations for distributed software development. In: Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), pp. 21–24 (2012)
15. Monasor, M.J., Vizcaíno, A., Piattini, M.: Cultural and linguistic problems in GSD: a simulator to train engineers in these issues. Journal of Software Maintenance and Evolution: Research and Practice (Special Issue on Global Software Engineering) 24, 707–717 (2011)
16. Krosnick, J.A.: Survey research. Annual Review of Psychology 50, 537–567 (1999)
17. Hoyle, R.H., Harris, M.J., Judd, C.M.: Research Methods and Social Relations. Wadsworth Publishing (2009)

## Appendix A: Survey

Practitioners were asked the following specific questions:

1. Users will not react in the exactly the same manner when interacting with Virtual Agents as when dealing with real participants. Do you consider this difference negative? Can this difference create a barrier to training?
2. Do you think that the tool could be really useful for training the skills and knowledge required in GSD?
3. Do you think it would be feasible to train members of your company/university by applying this training platform environment? Do you envisage any problems or inconveniences in its application?
4. By considering the example of the Requirements Elicitation training scenario shown, which other training scenarios would you like to be designed?

5. Do you think that it would be worth creating training scenarios for training particular problem recognition or skill development? Which ones?
6. How long would you consider the students could dedicate to these courses in your company? (give time scale)
7. Do you find any weak point in the environment not mentioned previously? What improvements would you suggest?

In the case of the researchers, they were also asked for personal data. Moreover they were asked to give their opinion from the students' point of view:

1. Age, 2. Nationality, 3. University where you obtained your degree.
4. Do you have theoretical knowledge in Global Software Development or Distributed Software Development? How did you get that knowledge?
5. Have you ever practiced GSD in your professional life? For how long? In how many projects? How many different cultures were involved?
6. From the perspective of the student, do you find the environment useful for developing the skills required in GSD?
7. By considering that the student must get as much information as possible and commit as few errors as possible during the simulations, how long would you consider a training scenario should last?

# Research Prototypes versus Products: Lessons Learned from Software Development Processes in Research Projects

Dietmar Winkler, Richard Mordinyi, and Stefan Biffl

Institute of Software Technology and Interactive Systems, CDL-Flex,
Vienna University of Technology
Favoritenstrasse 9/188, 1040 Vienna, Austria
{dietmar.winkler,richard.mordinyi,stefan.biffl}@tuwien.ac.at

**Abstract.** Software and systems development in industry typically focus on constructing high-quality products by using traditional or agile software processes and applying established tools and methods. Most projects have to handle more or less stable requirements but usually build on a proven architecture. On the other hand, research projects typically aim at investigating new ideas, facing promising research directions, showing feasibility of novel approaches or building prototypes for demonstration purposes. Obviously there seems to be a big gap between industrial projects and research projects. Anyway – after a period of research – there is the need to enable the transition from prototype to real products, comparable to industrial developed software products. The main challenge is bridging the gap between research prototypes and industry products, typically out of scope of a research project. As we have to handle these challenges in a long-running research project, this paper aims at identifying risks, challenges and candidate solutions to identify how to bridge the gap from research to industry. Main result of this paper is an adapted software engineering process that has been initially evaluated in context of our research project.

**Keywords:** Software Development Processes, Research Projects, Industry Product, Prototyping.

## 1 Introduction

Typical industry projects follow defined software engineering processes [16], e.g., traditional and/or agile development approaches, using well-defined methods and tools [1][6] covering important steps of the software development project, e.g., requirements elicitation, architecture definition, code construction, testing, and delivery/deployment of new and updated software products. Based on standardized software processes, process tailoring approaches support the application and adaption of software processes (out of the box) to a specific organizational or project context [5]. Based on adapted processes, methods, and tools companies typically implement and use well-established approaches within an organization, following a common goal: i.e., delivering high-quality (software) products to customers at an optimum cost/benefit ratio for all involved stakeholders.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 48–59, 2013.

On the other hand, research projects typically focus on different topics and different goals, e.g., investigating new ideas or facing promising research directions. Based on several stakeholders, e.g., funding organizations, principal and application industry partners, researchers, and developers, various and partly conflicting interests and goals are observable. Unclear, unstable, and frequent changing requirements are additional challenges to be addressed within a research project: new ideas come up frequent and can result in fundamental changes of the solution concept and – as a consequence – of the prototype solutions. However, typical outcomes of research projects are concepts, feasibility studies of novel approaches, and prototype applications for demonstration purposes. An important issue focuses on the empirical evaluation of concepts and research prototypes [3], i.e. demonstrating that the solution work like expected. In contrast to industry projects and products, research projects and prototypes typically have strong limitations regarding application capabilities in industry, and, thus, require appropriate processes, methods, and tools to transfer prototype solutions to industrial products. Based on these basic differences, we derived an important research questions:

*How can we bridge the gap between (a) research projects and industry projects and (b) research project prototypes and industry products*?

In this paper we report on challenges and candidate solutions based on experiences from our research project, i.e., CDL-Flex[1], a seven-year research project, started in 2010. The main objective of the project is to support engineers in large-scale engineering projects to (a) better collaborate and exchange data between different disciplines and (b) to improve the engineering process [4]. The project focuses on the automation systems domain (e.g., hydro power plants and steel mills) where engineers coming from various disciplines, e.g., the mechanical, electrical, and software domain, have to collaborate and exchange data efficiently. Efficient data exchange is a pre-condition for change management [18], even if different disciplines are involved. Please see Section 3 for a more detailed description of the research project. After three years of research, our industry partners claim to apply selected use cases in (his) industry environment. Nevertheless, the outcome of the research project is still classified as a (research) prototype. Thus, we need some mechanisms to transfer the prototype use cases to an industry product. In this paper we address these challenges (over time) and present a candidate solution for a software development process to support the transfer from research results to industry solutions, and report on findings and lessons learned after a three year period of our research project.

The remainder of this paper is structured as follows. Section 2 presents some related work in context of software development processes. Section 3 introduces to the CDL-Flex research project in more detail. We highlight research challenges in section 4 and present our candidate solution and first results in section 5. Finally, section 6 concludes and identifies future work.

---

[1] CDL-Flex: Christian Doppler Laboratory „Software Engineering Integration for Flexible Automation Systems", `http://cdl.ifs.tuwien.ac.at`

# 2 Related Work

This section summarizes related work on engineering processes in Section 2.1, methods and tools in Section 2.2, and product/process maturity levels in Section 2.3.

## 2.1 Software Engineering Processes

Research projects typically focus on an experimental development of new process, methods, and tools to gain knowledge in defined areas by using scientific methods [2]. Nevertheless, it has to be shown that concepts, derived from research activities work in a defined context. Thus, candidate outcomes of research in software engineering result in creating software solutions and prototypes focusing on defined purposes. Based on the nature of research projects, i.e., handling (a) highly instable requirements and (b) instable architecture approaches, the flexibility of engineering processes is a key requirement for research projects. Thus, an important question is which software processes are flexible enough to enable the construction of prototypes and products in context of research projects and – after successfully evaluating these prototypes – which processes aims at bridging the gap between research visions, prototype products, and industry products. In industry, several software process approaches, either traditional or more flexible (agile) process approaches are available for constructing industry products [16]. Among others, the Rational Unified Process [8] or V-Model XT [14] are candidates for application in research projects. Nevertheless, the structure of these processes and the pre-condition of more or less stable requirements might hinder successful application in research projects, where concepts, architecture, and implementation may change frequently. Agile processes, e.g., Scrum [15] or eXtreme programming, seem to fit well to research projects as they focus on user interaction and flexibility and support fast feedback-loops of individual stakeholders. Nevertheless, a stable baseline of tools, methods, and development environments is a pre-condition for software development. Unfortunately (early) research projects do not provide this kind of stable baseline. More current approaches, e.g., Lean Development or Kanban [9] can also provide an organizational framework for software construction – nevertheless, similar critics apply for research project application – there must be something stable to build on.

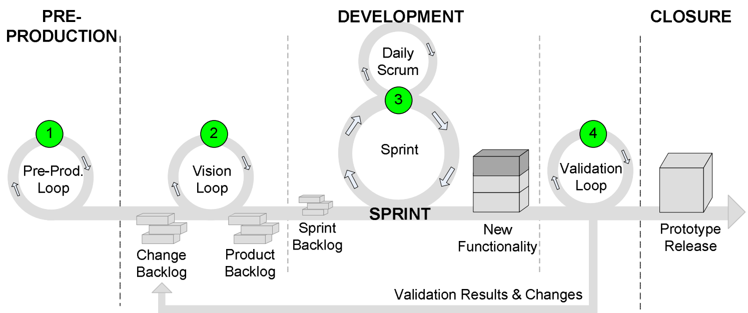



**Fig. 1.** Extended Scrum Process Model according to [10]

From the authors' perspective, research projects also include creative work to find new and promising solutions – similar to game development domain [10]. Thus, processes derived from game development are promising candidates for application in research projects. In Musil *et al.* we introduced a modified agile process approach based on Scrum including (1) pre-production, (2) vision, (3) product development, and (4) validation loops [10]. Figure 1 presents an overview of this adapted software engineering process:

(1) *Pre-Production Loop*. Goal is (a) identifying candidate use cases (e.g., during workshops with industry partners and researchers) based on visions and current needs and (b) cost-value considerations [3] for selecting most valuable use cases.
(2) The *Vision Loop* focuses on the product backlog maintaining product vision and changes from industry partner/researcher feedback.
(3) The *Sprint* enables developers in constructing and evolving the use case according to product and sprint backlogs.
(4) Finally, the main goal of *Validation Loops* include (a) use case verification and validation, (b) in-depth industry partner/researcher feedback, and (c) stimulation of new ideas and visions as a baseline for updating backlog for next iterations.

Based on our previous work [10] the extended Scrum process approach seems to be the most promising approach for handling research projects. Nevertheless, this process does not specify the exit/transition point from prototypes to industry products.

### 2.2 Methods and Tools

In industry projects methods and tools represent the foundation for product development. Basically, constructive methods and tools support engineers in building software documents (e.g., model-based or test-driven) while analytical approaches support defect detection, verification, and validation (e.g., reviews, inspection, and testing). Typically standardized methods and tools are available organization-wide in repositories for selecting and reusing them within the organization. In research projects, where several stakeholders collaborate (maybe in different research organizations), every researcher is using his/her own toolbox, which fits best to the individual requirements or individual preferences. Thus, there is a large base of different tools in a heterogeneous research landscape. While this approach might be suitable for research prototypes, where a small subset features are in the scope of research, this heterogeneity hinders efficient product development in industry. As a consequence an agreement on the most relevant methods and tools has to be established to support the transition from research prototypes to industry products.

### 2.3 Prototype and Product Maturity

A main issue of prototypes and products focus on the maturity of deliverables. Maturity levels are well-known in context of processes based on CMMI [7] or ISO 15504 (SPICE) [17] to estimate the capability levels of individual processes and the maturity of products and/or organization. Nevertheless in context of prototypes and products maturity levels are typically based on verification and validation results (e.g., results from test runs and acceptance tests). However, maturity levels based on the quality of prototypes and products seem to be a reasonable approach to assess individual work products, i.e., prototypes or products.

To support the transition from research prototypes to industry projects we see the need to introduce (a) defined software engineering process approaches, enabling flexible handling of requirements and stakeholder needs, (b) defined sets of methods and tools for application in research projects as well as industry projects, and (c) assessment approaches of prototype/product maturity levels with respect to apply the solution in an industry context.

## 3 The CDL-Flex Research Project

This section introduces to the CDL-Flex research project including the basic goals of the project, addressed research areas, and involved stakeholders.

### 3.1 Project Goals

Engineering projects in the automation systems domain, e.g., hydro power plants and manufacturing systems, depend on the knowledge of experts from a wide range of different disciplines and domains, e.g., mechanical, electrical and software engineering [12]. Individual knowledge is embodied in a heterogeneous set of domain-specific tools and data models. Weakly integrated tools and data models hinder efficient collaboration and data exchange between disciplines [4]. Main goal of the project is to support engineers and managers in large-scale engineering projects to overcome this technical gap between individual tools and the semantic gap of individual data models to better collaborate and exchange data in heterogeneous engineering environments.

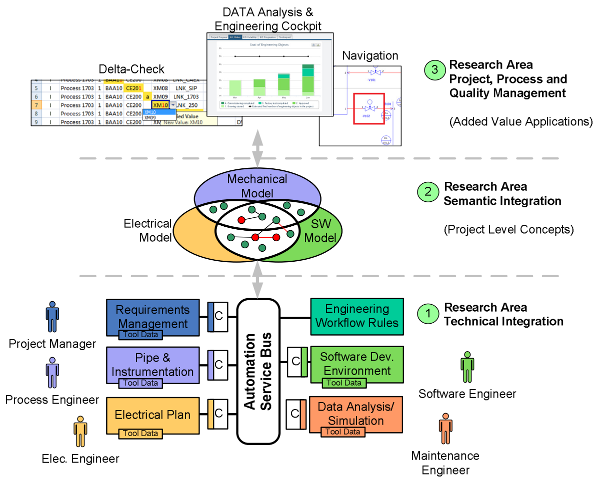



**Fig. 2.** Application Level-Concept of the Automation Service Bus

Based on the technical and semantic foundation, added-value components support end users and project managers in applying specific use cases in context of engineering projects in a heterogeneous environment. Figure 2 presents the application level-concept of the project with focus on three different research areas:

(1) *Technical Integration of Tools* represents the technical foundation to enable individual tools interacting with each other. For instance, a change request by the electrical engineer will be propagated across a middleware platform (i.e., the *Automation Service Bus, ASB*) to other affected engineers. See Winkler *et al.* for a more detailed description of this basic change management process [18].
(2) *Semantic Integration of Data Models.* Semantic heterogeneity of data models, caused by individual tools, hinders efficient data exchange. Note that the circles in Figure 2 represent discipline-specific data models. Common concepts, i.e., overlapping areas, are the foundation for mapping data models to each other. See Moser *et al.* for a more detailed description the common concept approach [13].
(3) *Added Value Application.* The technical and semantic integration of tools and data models enables added-value components, e.g., change management across domain and tool borders [18], project observation and control as well as comprehensive data analysis with the Engineering Cockpit [12], or efficient navigation between different tools, which are typically not connected to each other [11].

Note that these applications have been built as research prototypes (proof-of-concept) to show the feasibility of the underlying architecture (technical and semantic integration) and to enable added-value components for research and industry partners.

### 3.2 Project Stakeholders

Research projects typically involve a set of different stakeholders who have to collaborate to achieve defined (but different) goals, for instance:

- *Project Sponsors, e.g., public agencies,* typically focus on basic research, drive research goals forward, and enable good publications in the field(s) of research.
- *Principal Industry Partners* contribute with resources, e.g., additional funding, knowledge in the application domain, and customer contacts. Added values are the permission to use research results in their own business area as a product or as input for consulting activities after project completion.
- *Additional industry partners* who support researchers with real-world use cases, business domain knowledge, and test data. Main interest is getting challenges solved for application in their own business domain.
- *Researchers* in addressed areas focus on research challenges and publication in related communities.
- *Open Source Community (OSS).* Parts of the middleware platform are available as open source contribution[2] with the goal to make research findings public and usable within the OSS community. Note that there are ongoing plans for publication under Apache license.

---

[2] Open Engineering Service Bus (OpenEngSB): `http://www.openengsb.org/`

- *Developers*. Based on the project setting, we introduced two main development groups, i.e., an open source development team and an industry development team responsible for (principal) industry partner related use cases.
- *Power-Users* are application domain experts, who provide (a) domain knowledge and (b) feedback to early prototypes, and (c) support researchers in finding most valuable use cases based on industry needs for future research (research vision).
- *End-Users* should work with the application, i.e., the product, in daily business. Note that end-users typically require stable and working software products.

These basic groups of stakeholders have been introduced quite early in the project to enable effective and efficient prototype development, as required by research contracts. After three years of research and prototype building new roles, e.g., product management and quality assurance teams (QA team) have been established.

# 4 Research Challenges and Questions

Based on the related work and research/industry best practices we identified two main research challenges in context of the research project:

*RQ1. How can we bridge the gap between research projects and industry projects?* While systematic and established processes, methods, and tools are available in industry context, research projects have to be more flexible (e.g., changing architecture and requirements). Thus, a main challenge is to find a well-defined process to handle research and industry projects to support prototypes and products development.

*RQ2. How can we transfer research project prototypes to industry projects?* Research prototypes typically include strong limitations for industry application regarding stability, performance, and user acceptance. Thus, this question focuses on a classification and/or assessment for prototypes/products evaluation.

# 5 Solution Approach, Results, and Lessons Learned

This section summarizes the CDL-Flex solution approach, initial results, and presents lessons learned after three year of research and prototype construction.

## 5.1 Prototype and Product Maturity

A first step towards a successful transfer from research prototypes to industry projects is identifying maturity steps (levels) in prototypes and products. Nevertheless, the contribution and the “quality” of basic research (i.e., prototype development) and industry product development vary over time. Figure 3 presents the five basic maturity levels (or steps of development), implemented in the CDL-Flex research project:

- *Level 1 – Research Vision.* This development step includes creative processes, brainstorming, and workshops with industry partners to get ideas, visions and current needs of industry partners to be addressed in the research project.
- *Level 2 – Research Concept.* Based on initial ideas and visions basic concepts (concrete use cases including test data and test cases) are developed – mainly by researchers. Feedback cycles on the concepts enable early validation of the ideas

and visions and ensure that concepts meet individual requirements of industry partners. These prototypes are mocked (i.e., demos without real functional behavior) to simulate the expected behavior based on the initial concepts. Main goal is to receive feedback, e.g., on the user interface and the planned behavior.

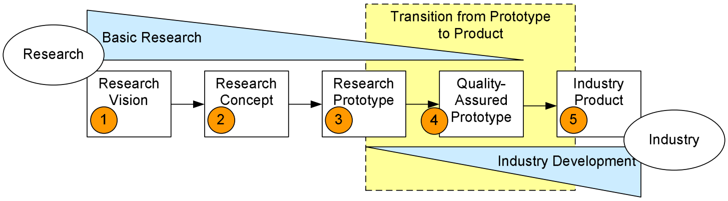



**Fig. 3.** Maturity Levels in Research/Industry Projects

- *Level 3 – Research Prototypes* include real functional behavior based on the research concept and industry partner feedback. An initial research prototype is based on agreed use cases, test cases, and test data to show the feasibility of the solution. Although basic quality assurance activities have been applied, these prototypes focus on providing the basic functionality with strong limitations to stability, robustness, usability, and fault tolerance.
- *Level 4 – Quality-Assured Prototype.* To enable more stable, robust and fault tolerant systems, additional implementation effort and extended quality assurance approaches, e.g., integration and acceptance testing, are required. Typically, these tasks are out of scope of researchers (who want to show the feasibility of the concept) and have to be executed by other stakeholders, i.e., industry development and quality assurance teams. Note that both teams usually have to be paid.
- *Level 5 – Industry Product.* The final maturity step focuses on real industry products where industry-related methods for development and quality assurance apply. In our project industry products are typically developed by our principal industry partner, supported by the industry development team at the CDL-Flex.

Note, that the impact of basic research contributions decreases over time and impact of industry contributions increases (see Figure 3). The most interesting part is the transition phase, involving (early) research prototypes (level 3, quality-assured prototypes (level 4), and – finally – industry products (level 5).

### 5.2 Software Development Process

To support individual requirements, derived from individual prototype/maturity levels, we applied the extended Scrum process model (see Figure 1) to the individual development steps (maturity levels) and highlight main contributions of involved stakeholders. Figure 4 presents the proposed extended process approach to enable prototype and product development and the transition between prototypes and products. Basically, researchers, industry partners, and power users derive a set of research vision use cases during workshops and discussions (1; “Pre-Production Loop”). Based on selected use cases researcher develop concepts and implement initial (mockup) prototypes for discussion and feedback (2; “Vision loop”). Note that these initial research prototypes are typically developed by students during their university work (e.g., diploma thesis or internships). Thus, the quality of the prototypes varies and the prototypes are not usable in industry context. Nevertheless, main results of this step are (a) prototypes for feasibility studies and (b) sets of more concrete requirements and features for research prototype implementation.

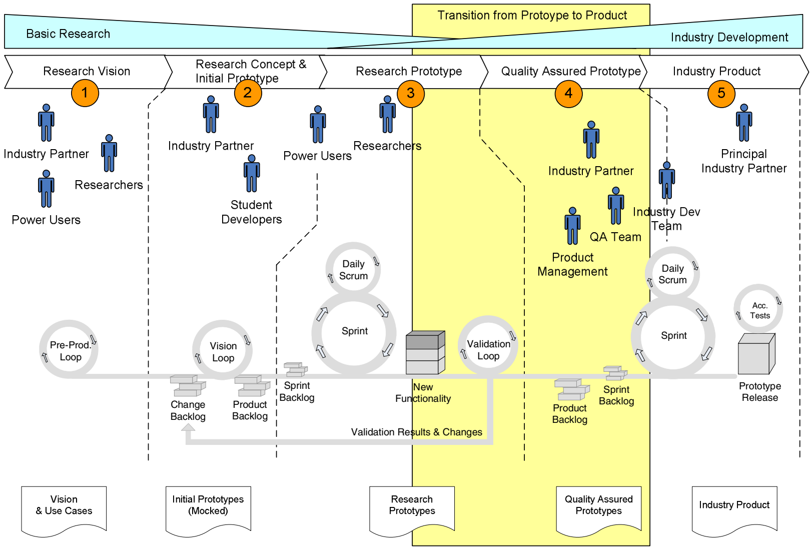



**Fig. 4.** Proposed Development Process

Based on these more or less stable requirements (derived from the vision loop) research prototypes (3) are developed by student developers. Additional feedback cycles (supported by industry partners, power users, and researchers) enable the improvement of research prototypes ("Validation Loop"). Nevertheless, these type of prototypes lack in robustness, stability, and fault tolerance because the goal is to show the feasibility of the prototype on functional level. To enable application in industry context, more stable and robust prototypes are necessary (4; "quality assured prototypes"). Thus, we nominate product owners, industry development and quality assurance teams for individual use cases and/or industry partners. Note that team members are responsible for the quality-assured prototype and are recruited as professionals. After a successful pilot application the quality assured prototype will be transferred to our principal industry partner, who is responsible for the product and have to integrate the solution in his product portfolio.

### 5.3 Software Engineering Environment

In early phases of prototype development, researchers apply methods and tools which seem to fit best to the requirements and individual preferences. Nevertheless, when starting implementing a research prototype a more stable development environment is necessary. Main artifacts and tools in context of our project are:

- *Use Cases*. Use cases represent the most important artifacts in context of the research project, i.e., (a) vision use cases that represent rough ideas and (b) concrete use cases for prototype implementation. Typically use cases are high-level goals (from the perspective of industry partners) which have to be split into several features, represented in backlogs assigned to different maturity levels. Note that a use case includes a brief description, real-world data sets for testing purposes, and success criteria for industry partner acceptance. We use Confluence[3], a collaboration tool, for managing use case and related engineering documents.

- *Features, Issues, Bugs.* We use Jira[4], a project tracking software, to manage individual aspects of the use case, e.g., user stories, issues and bugs. Note that all use case related information are linked from use case descriptions (provided in Confluence) to individual issues (provided in Jira) and individual developers.
- *Jenkins.* To support continuous integration and test processes, we applied Jenkins[5], an open source server to monitor and control the project progress including quality assurance checks with respect to quality-assured prototypes, e.g., by implementing Checkstyle[6] and CodeCover[7].
- *Testing of Prototypes/Products.* Beyond developer testing based on unit tests, we implemented acceptance tests for features and use cases. Acceptance tests, maintained by the QA team, are used to (a) establish an early and common understanding of the use case and (b) to enable automation-supported testing of implemented research prototypes. Based on the maturity of prototypes/products, we applied manual tests during research prototype development and automation supported tests with Selenium[8] at the level 4 ("quality-assured prototypes") latest. This quality assurance strategy enables us in (a) separating individual maturity levels and (b) fast feedback in case of changes.

**Table 1.** Lessons Learned and Key Findings

| | Vision | Concept | Research Prototype | Quality Assured Prototype | Industry Product |
|---|---|---|---|---|---|
| **Outcome** | Research Vision | Reserach Concept<br>Mock-Up Prototype<br>Proof of Concept<br>Feasibility Study | Use Case / Features<br>Functional Prototype | Use Case / Features<br>Prototype: robust, stable, and fault tolerant | Use Case / Features<br>Industry product |
| **Maturity Level** | n/a | low | low | medium | high |
| **QA approaches** | informal feedback | systematic feedback<br>test case definition | test case definition<br>manual tests | automated tests<br>QA metrics | According to engineering process |
| **Users** | Researcher | Researcher<br>Developers | Researcher<br>Developers<br>Power Users | Industry Partners<br>Power Users<br>End Users | Industry Partners<br>Power Users<br>End Users |
| **Evaluation** | informal discussion | interviews and feedback | basic tests | Automated tests<br>QA metrics<br>Acceptance Tests | Automated tests<br>QA metrics<br>Acceptance Tests |
| **Cost/Value evaluation** | Estimation of experts and researchers. | Expected benefits based on state of the practice (Experts) | Basic measurement results from pilot applications, | Comparative evaluations in real world settings (pilot | Comparative evaluations in real world settings (pilot |

[3] Atlassian Confluence: http://www.atlassian.com/en/software/confluence
[4] Atlassian Jira: http://www.atlassian.com/software/jira/overview
[5] Jenkins: www.jenkins-ci.org
[6] Checkstyle: www.checkstyle.sourceforge.net
[7] Codecover: http://codecover.org/
[8] Selenium: http://docs.seleniumhq.org/

### 5.4 Lessons Learned

After three years of research and observing/analyzing engineering processes, we identified a set of challenges, risks, and candidate solutions for prototype and product development, which can be addressed by (a) flexible software development processes, based on an extended Scrum process (Section 5.2), (b) a set of tools within the development environment (Section 5.3), and (c) a five-level maturity concept to estimate and asses the maturity of development steps and deliverables. Table 1 presents a brief summary of our key observation and practices applicable in every development step of a single use case. Note that we also include the level of quality assurance, involved stakeholders/users, candidate prototype/product evaluation approaches, and cost/value considerations from industry partner perspective.

## 6 Conclusion and Future Work

In this paper we reported on our experiences from three years of research work of a seven year research project. The main challenge was to bridge the gap between research prototypes, typically constructed based on visions and research ideas, and industry prototypes, usable in industry context. Based on different requirements and involved stakeholders there is a need for (a) engineering processes that support research projects and industry projects (RI 1) and (b) a maturity concept that enables an efficient classification of deliverables to support a smooth transfer from research prototypes to industry products (RI 2).

Lessons learned from previous process improvement initiatives in creative application domains, e.g., game development, can help addressing visions and instable requirements as well as an instable architecture. Figure 4 presented the application of an adapted Scrum process approach [10] in context of research projects. In addition we learned that different method approaches apply in different stages of use case development. Table 1 presented the most important findings derived from project observations. Another important finding was the involvement of different stakeholders, especially in the development process where students can work on individual use case in a defined scope. If research prototypes evolve towards quality-assured research prototypes for pilot applications and industrial products (i.e., an increasing maturity level), professionals are required to enable the construction of high-quality prototypes and/or products. Based on our experience, we believe that the proposed engineering process and the suggested maturity levels can help in better addressing the need of individual expectations coming from research and industry to bridge the gap between research and industry project and research prototypes and industry products.

Future work includes a more detailed evaluation (i.e., a case study) of the purposed process approach to (a) get a more detailed view on the effects of the process and (b) to continue improving the proposed engineering process. In addition we have to evaluate the defined tool-set with respect to applicability in research and industry context in the next phase of our research project.

## References


[1] Abran, A., Moore, J.W.: Guide to the Software Engineering Body of Knowledge. IEEE (2004)

[2] Bell, J.: Doing your Research Project: A Guide for first-time Researchers in Education, Health and Social Science, 5th edn. Open University Press (2010)

[3] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (eds.): Value-Based Software Engineering. Springer (2005)

[4] Biffl, S., Schatten, A., Zoitl, A.: Integration of heterogeneous engineering environments for the automation systems lifecycle. In: Proc. of 7th Conf. on INDIN, Cardiff, Wales, pp. 576–581 (2009)

[5] Biffl, S., Winkler, D., Höhn, R., Wetzel, H.: Software Process Improvement in Europe: Potential of the new V-Model XT and Research Issues. Journal Software Process: Improvement and Practice 11(3), 229–238 (2006)

[6] Chemuturi, M.K., Cagley, T.M.: Mastering Software Project Management: Best Practices, Tools, and Techniques. J. Ross. Pub. Inc (2010)

[7] Chrissis, M.B., Konrad, M., Shrum, S.: CMMI for Development: Guidelines for Process Integration and Product Improvement. SEI-Series in SE (2011)

[8] Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley Longman (2003)

[9] Ladas, C.: Scrumban - Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press (2009)

[10] Musil, J., Schweda, A., Winkler, D., Biffl, S.: Improving Video Game Development: Facilitating Heterogeneous Team Collaboration Through Flexible Software Processes. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 83–94. Springer, Heidelberg (2010)

[11] Mordinyi, R., Moser, T., Winkler, D., Biffl, S.: Navigation between Tools in Heterogeneous Automation Systems Engineering Landscapes. In: 38th Annual Conference of the IEEE Industrial Electronics Society, IECON (2012)

[12] Moser, T., Mordinyi, R., Winkler, D., Biffl, S.: Engineering Project Management using the Engineering Cockpit: A collaboration platform for project managers and engineers. In: 9th Int. Conf on Industrial Informatics, INDIN (2011)

[13] Moser, T., Biffl, S., Sunindyo, W.D., Winkler, D.: Integrating Production Automa-tion Expert Knowledge Across Engineering Domains. Int. J. of Distributed Systems and Technologies (IJDST), SI 2(3), 88–103 (2011)

[14] Rausch, A., Bartelt, C., Ternite, T., Kurmann, M.: The V-Modell XT Applied – Model-Driven and Document Centry Development. In: Proc. of the 3rd World Congress for Software Quality, pp. 131–138 (2005)

[15] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press (2004)

[16] Sommerville, I.: Software Engineering, 9th revised edition. Addison-Wesley Longman, International Version (2010)

[17] Van Loon, H.: Process Assessment and ISO/IEC 15504: A Reference Book, 2nd edn. Springer (2007)

[18] Winkler, D., Moser, T., Mordinyi, R., Sunindyo, W.D., Biffl, S.: Engineering Object Change Management Process Observation in Distributed Automation Systems Projects. In: 18th EuroSPI Conference, Roskilde, Denmark (2011)

# An Analysis of the Software Development Processes of Open Source E-Learning Systems

Aarthy Krishnamurthy[1] and Rory V. O'Connor[2]

[1] School of Electronic Engineering, Dublin City University (DCU), Dublin Ireland
aarthy.krishnamurthy2@mail.dcu.ie
[2] School of Computing, Dublin City University, Dublin Ireland
roconnor@computing.dcu.ie

**Abstract.** In recent years there has been a rapid increase in demand for e-learning systems. The software development process plays a crucial role in the design and development of a high-quality e-learning system. However, to date, there is no comprehensive comparative study of open source software (OSS) development process for different OS e-learning systems. This hinders the development of a generalized OSS development process, a key requisite for rapidly developing high-quality OS e-learning systems. This paper provides a full analysis of different existing and successful OS e-learning software systems and the best practices followed in the e-learning development. In particular, this paper investigates the software development activities of Moodle, Dokeos and ILIAS. An activity flow representation that describes their current development practices is constructed individually for all three OS e-learning systems. Further, a comprehensive comparative analysis is carried out that leads to an explicit identification of various development stages of the three OS e-learning systems.

**Keywords:** Activity flow diagram, e-Learning, open source software, software development processes.

## 1 Introduction

E-learning can be broadly defined as transfer of knowledge and skills electronically, through different communication medium and devices [1]. Further, in e-learning systems, the learner is not always at a fixed, predetermined location. The learner can take advantage of the opportunities offered by mobile technologies [2]. The principal benefit is the ability to provide users the flexibility of learning and efficiently communicating anytime and from anywhere.

There are many e-learning systems that are developed successfully (Moodle, ILIAS, Blackboard, etc). They are developed either as open source software (OSS) or as closed source software (CSS) systems. Most of the commercial CSS have been developed based on either a traditional software process or some form of tailored traditional process, in order to accommodate local needs. These development processes have associated standards/guidelines that are followed for high quality

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 60–71, 2013.

software development. On the other hand, OSS systems are developed by a community of like-minded developers, who are geographically distributed, yet work together closely on a specific software product [5].

OSS development (OSSD) has gained significant attention in recent years and is widely accepted as reliable products (e.g. Moodle, Apache, Linux, etc.). In today's times, the educational pattern evolves continuously with time. Hence, in order to keep pace with this evolution, the next generation e-learning systems need to evolve with the educational patterns. Before developing a generalized OSS process for e-learning systems, it is imperative to analyze and understand the existing and successfully running OS e-learning systems. To the best knowledge of authors, there has been no comprehensive study of the development activities of different OS e-learning systems. Hence, this paper carries out this study. In particular, this paper focuses on the development activities/process of three successful and highly popular OS e-learning systems - Moodle, ILIAS and Dokeos.

The development activities of these three OS e-learning systems were identified using two different approaches.

i. The first approach was to collect information from their websites, blogs, wiki pages and/or from any social network/media used by the community to broadcast the information. In addition, information was also collected from bug tracking system (or any other tracking systems), as some of the OSS communities track each of its development activities in such systems.

ii. The second approach was adopted only when the information collected from the first approach was either incomplete and/or ambiguous. It is a direct method wherein, questions were posted in public OSS community forums. The idea was to seek response directly from the community members, either through the same forums or through e-mails/private messages. However, the main disadvantage of this approach was that, there was no clear consensus from the contribution of the community members, on many occasions. In such scenarios, separate e-mails had to be sent to the core members and other experienced developers within the OSS development communities. This helped in identifying many nuances of the current development activities. Importantly, no analysis was done until all the information was gathered. This was strictly followed to avoid any ambiguity due to incorrect assumption of the current development practices.

Once the information on the development practices of each of the three e-learning systems was gathered, an in-depth analysis was carried out. The results of the analysis were then modeled using activity flow diagrams. The activity flow diagram representation was used because it provided a dynamic aspect of an overall flow of the development practices followed by the OS communities. The activity flow diagram was preferred over other approaches like state diagrams and event driven process diagrams as it would indicate an overall flow of activities carried out within the community for its software development.

The paper is organized as follows: Section 2 describes in detail the development activities of the each of the three OS e-learning system. Subsequently, section 3 compares the development practices of the three e-learning systems. Finally, Section 4 provides a brief conclusion along with the next research steps in this direction.

## 2 Development Activities

Over the years, the development activities of OSS have become openly visible to all and the development artifacts are publicly available over the web. Further, there is little need for formal project management, virtually no budget and often a very flexible schedule. OSSD is oriented towards the joint development by a community of developers [6]. All the three systems considered for analysis in this paper are OSS and follow their own development practices. In this paper, they are investigated through a case-study since there is very little literature available that discusses the OS e-learning systems and their development activities. The case study approach was selected as it will result in a detailed analysis of the developmental activities; thereby leading to comparisons which would help in understanding the similarities and differences on the practices followed by OS e-learning system development community. All three selected OS e-learning systems are quite popular worldwide and importantly, receive significant and frequent contributions from volunteers for performing various development activities regularly. The development activities of Moodle, ILIAS and Dokeos are described as follows.

### 2.1 Development Activities of Moodle

Moodle stands for 'Modular Object-Oriented Dynamic Learning Environment' and is one of the early and most successful OS e-learning platforms which follow strong pedagogical principles. It has the largest user-base and has the benefit of having the largest market share and highest satisfaction in small companies, educational institutions and government organization's LMS [3]. The different activities of Moodle development are shown in Fig. 1. It should be noted that Moodle does not have a clear bifurcation between various development stages. Contrastingly however, the development always starts with selecting the right candidate feature [7].

The first step of development involves selecting the right candidate feature. For selecting a candidate feature, the community pools the entire feature requests raised in the Moodle moot discussions, user's feature request from forums and feature request from moodle vendors. These candidate features are then voted for entering into the release roadmap list.

Any developer interested in developing the new feature listed in the release road map could initiate a discussion with other fellow community developers, in order to ensure that no one else is working on that requirement/feature. The developer(s) would then discuss their ideas with others, confirm the merits and the need for the particular feature, and importantly, evaluate theirs and other's ideas. Once the feature is selected for development by a Moodle developer, he/she is expected to come with design documents along with other specification documentations. These documents are then posted in the Moodle wiki. In addition, a tracker item is created for the feature and assigned to the developer. Subsequent changes are then carried out, based on the feedback received by the developer in the respective documents which are then updated in the wiki. Once the changes made are agreed by the Moodle community, the developer begins coding. When the development is completed or a major milestone is reached, it is the responsibility of the developer to advertise the feature for testing. Testing could be done by interested candidate(s) within the Moodle community. Subsequently, any bugs found are then reported and fixed. It is then integrated with the main version of Moodle and then released as a new version, which would be open and freely available.

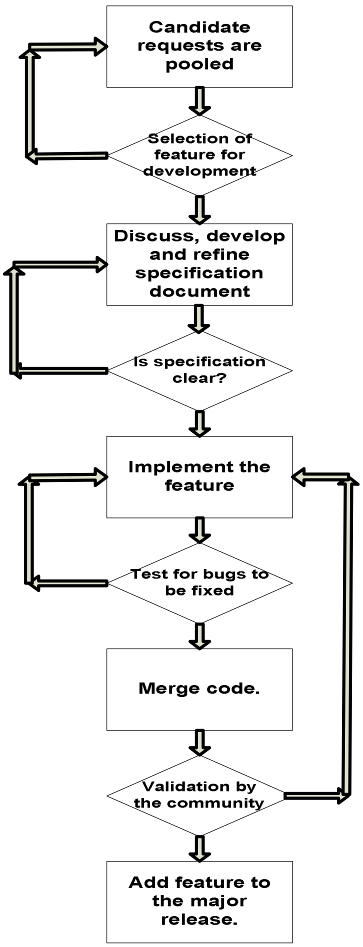



**Fig. 1.** Activity flow representation of Moodle

Additionally, a tracker item is created for the feature and assigned to the developer. Subsequent changes are made, based on the feedback received by the developer in the respective documents which are then updated.

### 2.2 Development Activities of ILIAS

ILIAS stands for Integrated Learning Information and co-operAtion System). It is a popular web based learning management system (LMS) / OS e-learning systems and comprises of six stages of development [8]. They are; Vision/Concept, Specification, Implementation, Documentation, Testing and Release & Maintenance. In each of these stages, the OSS community performs various developmental activities which can be observed clearly in Fig. 2. The details of each stage are described as follows:

- ***Vision/Concept Stage:*** This is the first stage of development wherein, ideas are proposed and published in wiki. The core development team will then decide on how to start the development. If the idea is already been put on to the feature wiki, people with similar interest are requested to work with them and develop the feature collaboratively.
- ***Specification Stage:*** In this stage, all major development is expected to have corresponding use cases or mock up screen-shots. For other minor developments/enhancements, developers would start with the feature wiki where it will describe the feature in detail, the purpose, etc.
- ***Implementation Stage:*** The third stage which is the implementation stage, where the coding/programming is done by the developers. Each module that is developed in this stage is tested by the developer who also fixes the initial bugs that comes across. Further, the developer would either perform a unit-testing using PHP Unit, or get it done by a tester. Subsequently, the code is then merged with CVS.
- ***Documentation Stage:*** There are two types documentation prepared for a feature developed for ILIAS - technical documentation and user documentation. The technical documentation consists of the class and functional documentation generated by PHP Doc. The user documentation will be mainly instructions for the average user on how to use it. The user documentation is only released at the time of release of the product._
- ***Testing Stage:*** The testing stage mainly follows the implementation stage. In this stage, once the unit-testing and code merger is done, an alpha release is carried out for further testing and bug fixing. It is the responsibility of the developer to appoint a tester to test the module developed by him. If the developer is unsuccessful in finding a tester to test his/her module, then the core team would carry out the required testing. However, in any case, the developer himself cannot be a tester for his own developed module
- ***Release Stage:*** The final stage during the development is the release stage wherein, the new modules that have undergone alpha testing are released under the beta version. Errors/bugs encountered after the beta release are then entered into the bug tracker (Mantis bug tracker). These bugs are then fixed and released as the main stable version.

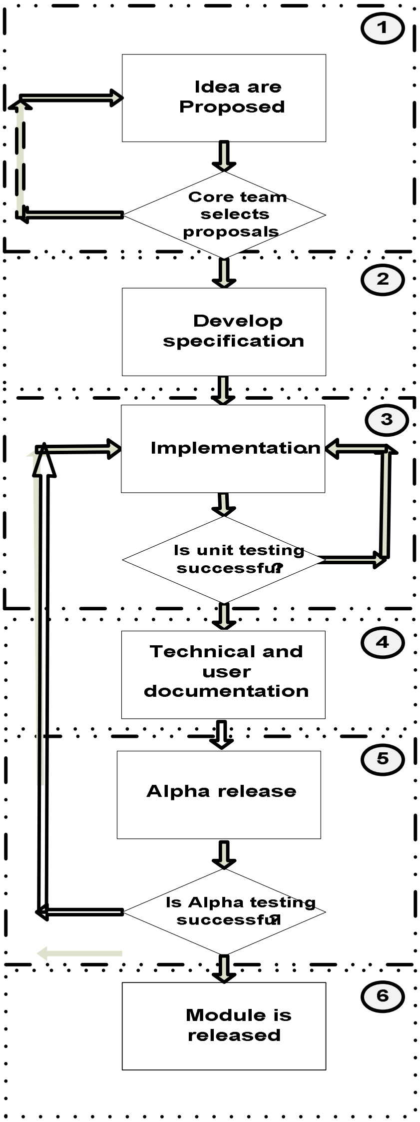



**Fig. 2.** Activity flow representation of ILIAS

## 2.3 Development Activities of Dokeos

Dokeos is developed both as commercial and OSS version. The development of OSS version is the responsibility of the Dokeos community – from initiation of idea through release. Although there are two different existing systems, the OS version does provide all the basic features for free without any licensing cost to its users.

Dokeos community does not follow any defined stages as in ILIAS, but often, they do perform some activities in a particular order as shown in Fig. 3. Development of a feature starts with feature selection where the selected feature is added to the roadmap for development. The feature is then developed by the community of developers. The features are first tested before it is given it to the users for further testing. Any anomaly, if and when found are fixed and passed on to the users for user testing.

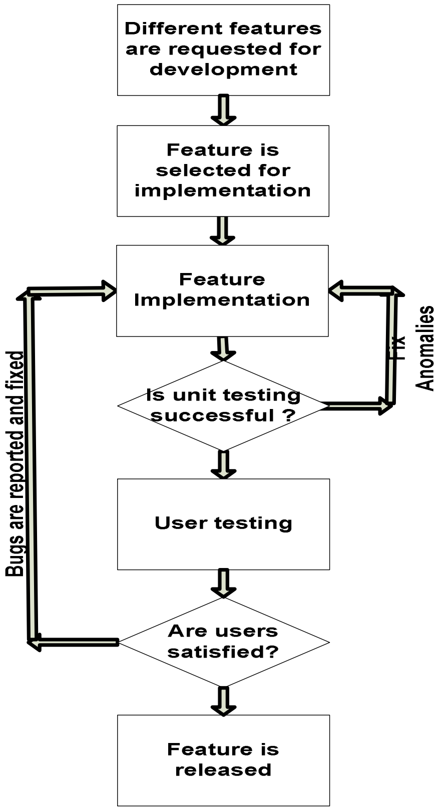



**Fig. 3.** Activity flow representation of Dokeos

The users would test the developed feature and if there is any bug(s), it would be reported. These bugs are then fixed and sent to the user again for testing. Once the user is satisfied with the features, they are subsequently released to the community as a stable version. All the users could then download it for free and use the same.

## 3 Comparison of OS E-Learning System Development Practices

The individual analysis of the three OS e-learning systems carried out in section 2 provided details and insight into the different activities carried out at different stages of development. Notably, the manner in which each stage is carried out would depend entirely on the expertise, experience and availability of resources and skills. There are distinct similarities and differences between Moodle, ILIAS and Dokeos on different aspects. These are summarized in Table 1. The comparative analysis begins with differences in number of developmental stages. The common developmental activities in each of the stages are then compared, based on different factors like, how it has been performed, who performs it, etc. Each of these differences and similarities are discussed briefly and is described as an observation and critique.

**Table 1.** Comparison between three OS e-learning system development

| LMS / Parameters | Moodle | ILIAS | Dokeos |
|---|---|---|---|
| **Number of development stages** | Do not categorize development stages | Does categorize six development stages | Does not categorize stages |
| **Who validates the proposed idea** | Anyone can validate the idea and comment on it | Only the core team validates the proposed idea | Does not validate the proposed idea at this stage |
| **Detailed development plan** | No plan is produced | No plan is produced | No plan is produced |
| **Person(s) responsible for development** | A person who volunteered initially & the team that was formed latter on the fly. | A person who volunteered initially & the team that was formed latter on the fly. | Any interested volunteer engages in developing the software. |
| **Testing** | Anyone can test at any time. | Anyone can test at anytime. | Anyone can test till the product is released. |
| **Release** | Two stage release process is followed. | Two stage release process is followed. | Once the testing is done & bugs are fixed, the product is released. There is no beta release. |

- **<u>Number of Software Development Stages</u>**

  *<u>Observation:</u>* In ILIAS, it is easy to identify different development stages /phases during development. However, Moodle and Dokeos do not categorize different software development stages, even though it has many tasks similar to ILIAS.

  *<u>Critique:</u>* Having defined stages or phases of development are important as it aids in easy tracking of the development activities as well as assists in planning and testing different phases independently.

- **Scrutiny of the Proposed Idea**

  *Observation:* New ideas proposed to Moodle and ILIAS is scrutinized immediately after its proposal. At the same time, there is one major difference between Moodle and ILIAS. In case of Moodle, anyone who is interested in the new idea, including the core team, co-developers, testers, users, etc. can read the proposal document and comment on it. Based on the received feedback, the core team or the core members will signal the development. However, in case of ILIAS, only the core members will review the idea/feature and would decide its future. On the other hand in Dokeos, specifications are not detailed or developed for idea scrutinization.

  *Critique:* Assessing the features credibility and need even before the specifications are developed might lead to inappropriate judgment with regard to the features need and importance.

- **Person(s) Responsible for Specification/Scrutiny**

  *Observation:* In case of Moodle, the entire community could scrutinize the specification by reading the proposal document and commenting on it. Based on the feedback the core team/ members would either agree/ disagree with the idea. On the other hand, as compared to Moodle, ILIAS has a different approach. In case of ILIAS, only the core members would scrutinize the idea/feature decide its future. On the other hand, Dokeos does not have any such activity and therefore the community is not responsible for the same.

  *Critique:* Being open source and built by users for users, the specification validation should be kept open. This will make sure that the specification is acceptable from the OSS user's point of view. This is very important because, in all cases, development happens based on the specification. If the specification happens to be wrong, then the developed feature would go wrong. This is true for all the software products including OS e-learning systems, irrespective of the development method followed.

- **Developmental Plan**

  *Observation:* In all three systems i.e., Moodle, ILIAS and Dokeos, there are no explicit plans portrayed for its development. It is the responsibility of the person in-charge to develop the feature as agreed upon. At the same time, it is the individual or team's responsibility to answer all queries regarding the module/feature development.

  *Critique:* Even though having a defined plan is beneficial in tracking the development; it is very complicated to come up with plans and follow it strictly in the OSS environment where the volunteers develop the product during their free time.

- **Person Responsible for Development**

*Observation:* In Moodle and ILIAS the person who agreed to develop the feature takes responsibility of its implementation. Further, the team formation happen on-the-fly based on the personal interest of the community member(s). If anyone is interested in its implementation, testing, documentation, etc. they volunteer to the working group/person.

*Critique:* Even though having a defined plan for developing a feature may seem to be a 'failsafe' approach, it is not practical to follow such a structure in an OSS environment. This is especially so, when a feature is developed by geographically distributed community members who volunteer to do the same in their spare time not just for themselves but also for others.

- **Testing**

*Observation:* In all the three OS e-learning systems, any individual from the community who is interested in a particular feature can test the developed code for any potential bug(s). However, there is one notable difference. In case of Moodle and ILIAS, the common ground testing could be carried out even after new versions are released. On the other hand, in case of Dokeos, this type of common ground testing could be done only till the product is released.

*Critique:* Testing is one of the important activities in producing a quality software product. OS software products are usually well-tested due to the large number of user-base/testers, who are geographically distributed, have varied skill sets and could test the module/feature independently.

- **Product Release**

*Observation:* A two-stage testing process is employed in case of Moodle and ILIAS. Once the initial testing is over, both Moodle and ILIAS release their features as a 'beta' version. Subsequently, this is tested again. Once the testing is completed, the features are then finally released along with other items as final version of the major product release. On the other hand, Dokeos does not have any beta release. The feature(s) are tested by users/community once it is developed and the bugs are reported. Once the encountered bugs are fixed, the feature is subsequently released.

*Critique:* Having a beta test stage will enable identification of problems before the integration to the stable version. This would potentially save any additional costs (in most cases it's the time spent by the OSS community) that might have to be incurred if the stable version is corrupted.

## 4 Conclusion

This research work carried out a detailed individual analysis and a comprehensive comparative analysis of the development activities in the three major open source e-learning systems - Moodle, ILIAS and Dokeos [11]. The results of the analysis were presented using an activity flow diagram representation. The comparison demonstrated the clarity and explicitness of each development stages carried out in the three OS e-learning systems. Further, for any differences identified in the development practices, a corresponding critique has been presented. This resulted in a better understanding of the best practices followed in the three OS e-learning environment.

Significantly, there were two major limitations encountered in the activity flow representation. Firstly, it did not identify the actors involved in carrying out various tasks and secondly, it did not explicitly specify the outcome of a development activity. Hence, having identified the different development stages, the next step in the design of generalized OSSD process for e-learning systems would be to construct a high level abstract model which would focus on the actors performing the activity and their outcome.

**Acknowledgment.** The author would like to thank Irish Research Council (IRC) - Embark Initiative Program for their support.

## References

[1] Tavangarian, D., Leypold, M., Nölting, K., Röser, M.: Is e-learning the Solution for Individual Learning. Journal of e-Learning 2(2) (2004)

[2] MOBIlearn (2003), Guidelines for learning/ teaching/ tutoring in a mobile. Technical report, `http://www.mobilearn.org/download/results/guidelines.pdf.`

[3] Wexler, S., Dublin, L., Grey, N., Jagannathan, S., Karrer, T., Martinez, M., Mosher, B., Oakes, K., van Barneveld, A.: "LEARNING MANAGEMENT SYSTEMS (LMS)" in GUILD Research 360 degree report (2007), `http://www.elearningguild.com/research/archives/index.cfm?id=120`

[4] Clarke, P., O'Connor, R.: Towards the identification of the influence of SPI on the successful evolution of software SMEs. In: Dawson, et al. (eds.) Proceedings of the 18th Int. Conf. on Software Quality Management (2010)

[5] Scacchi, W., et al.: Understanding Free/Open Source software Development Process. Softw. Process Improve. Pract. 11 (2006)

[6] Scacchi, W.: Software Development Practices in Open Source Software Development communities: A Comparative Case Study. In: 1st Workshop on Open Source Software Engineering (May 2001)

[7] Moodle, Moodle Development: Overview (2011), `http://docs.moodle.org/en/Development:Overview`

[8] ILLIAS, ILIAS Development Guide (2011), `https://www.ilias.de/docu/goto_docu_lm_42.html`

[9] Sakai, Sakai development work (2011), https://confluence.sakaiproject.org/display/MGT/How+Sakai+Development+Works

[10] Nichols, D.M., Twidale, M.B.: The usability of open source software. First Monday 8(1) (2003)

[11] Krishnamurthy, A., O'Connor, R.V.: Using ISO/IEC 12207 to Analyze Open Source Software Development Processes: An E-Learning Case Study. In: O'Connor, R. (ed.) SPICE 2013. CCIS, vol. 349, pp. 107–119. Springer, Heidelberg (2013)

# Identifying Process Problems with the SAWO Functional Defect Classification Scheme

Tanja Toroi, Anu Raninen, Hannu Vainio, and Lauri Väätäinen

University of Eastern Finland, School of Computing, Kuopio, Finland
{tanja.toroi,anu.raninen,hannu.vainio,lauri.vaatainen}@uef.fi

**Abstract.** In this paper we present the first official version of SAWO, a functional defect classification scheme developed to enable the usage of defect data for Software Process Improvement (SPI) purposes. Defect data is one of the most important, although nowadays perhaps least discussed management information sources for SPI decisions. Applying our scheme, defects can be classified with accuracy needed to generate practical and targeted process improvement suggestions. The SAWO scheme classifies defects on three levels. On the first level, the focus is on software defects in general. The second level focuses on functional defects and the third level brings more detail to the functional level. Further, we present the validation results of SAWO with three software companies' defect data consisting of 6363 defects.

**Keywords:** SAWO defect classification scheme, defect data analysis, process improvement.

## 1 Introduction

Defect data is rarely utilized in Software Process Improvement (SPI) [1]. Defect data is neglected even though defect data analysis is recognized as an effective and important approach for process improvement [2]. In addition, defect analysis, tracking and removing the major sources of defects offer the greatest short-term potential for improvements [3]. Previous research has shown that the defect classifications can be used to identify product and process problems [4] and to improve the testing and/or inspection activities [5]. In this paper we present a way of making defect data applicable as an input for practical process improvement.

We've developed the SAWO defect classification scheme to enable defect data analysis based process improvement. SAWO is a functional defect classification scheme, which classifies defects on three levels. The first level classifies defects at a general level. On the second level, the focus is on functional defects and the third level deepens the functional level, i.e. defects in control flow, processing, calculations and/or functional logic. In addition, missing duplicated, or overlapped features are considered functional defects. The SAWO scheme is based on an analysis of 11879 defects, see [6, 7]. Initial version of SAWO has been first published in [7]. The scheme presented in this paper is an improved version and is considered here as the official version. The representation style and descriptions of defect types have been

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 72–83, 2013.

evolved based on feedback from our case organizations. In addition, the scheme has now been validated with three software companies defect data.

In this paper we show how the SAWO scheme can be applied in software process improvement. The results presented are based on applying the SAWO scheme for 6363 defects of three software companies.

Our preliminary analysis [6] showed that defect distributions are surprisingly similar in different case organizations. The preliminary analysis was based on defect classification derived and combined from existing defect classification schemes [8, 9]. In addition, we noticed that 65% of the defects stored in the companies' databases are functional defects [6]. Hence, it appeared that the existing classification schemes [8, 9] were not detailed enough to produce meaningful results to be applied in SPI. To be able to truly benefit from the defect data, a more detailed defect classification was needed. However, such defect classification schemes are practically nonexistent in literature. The SAWO scheme was created to fill this gap.

Applying the SAWO classification scheme has been encouraging: the result of the classification is detailed enough to enable the identification of practical inputs for process improvement.

In comparison to our initial literature based scheme [6], the result of the new SAWO scheme is able to show the differences in software companies. The defect distributions reported in [6] were fairly similar as opposed to those generated applying the SAWO scheme. The differences in software companies defect data seem to lie on the functional level. The result of SAWO would appear to show the real problematic areas of the process in question. Hence, it would appear that by applying the SAWO scheme on the functional defect level we are able to learn something from the defects that we have not been able to make visible before.

The overall structure of this paper is: Research setting is described in section 2. In section 3, we present our defect classification scheme, SAWO. Section 4 describes the results of applying the SAWO scheme. Section 5 presents process improvement suggestions based on SAWO defect data analysis. The results are discussed in section 6 and section 7 provides the conclusion.

## 2 Research Setting

In this section we present the research problem and the case organizations. The research method used was case study [10].

### 2.1 Research Problem

It is shown that software defect data is one of the most important available management information sources for software process improvement decisions [3]. We conducted a preliminary study in spring in 2011 to find out what the most common defect types are and how this information can be used in process improvement [6]. The study was conducted using defect data from three software companies consisting of 11879 defects in total. Based on the results of the preliminary study it was noticed that further research was needed. The defect classification scheme applied was too general to provide detailed information to be applied for process improvement purposes.

In order to utilize the defect data for process improvement purposes, defects had to be understood in more detail. Hence, the research problem of the study is: Does the SAWO defect classification scheme provide practical inputs for software process improvement?

### 2.2 The Case Organizations

In this section we describe the case organizations of the study. The case organizations are presented in table 1.

**Table 1.** Case organizations

| | **Company A** | **Company B** | **Company C** |
|---|---|---|---|
| **Market** | Farming | Metal industry | Telecommunications |
| **Size** | Small | Large | Medium |
| **Employees in development / system testing** | 9 / 4-6 | 24 / 2 | 30 / 1 |
| **Country** | Finland | Multinational | Finland |
| **Defect tracking system** | Mantis | Jira | HP Quality Center |
| **Language of the defect descriptions** | Finnish | English | Finnish |
| **Analyzed defects, in total** | 2938 | 554 | 2871 |
| **Functional defects** | 1826 | 185 | 1788 |
| **Coding language** | Delphi | C# | Java |

The case organizations in this study are dissimilar in many ways. For example, they produce software products for very different business domains. In addition, the companies differ in size. The study presented in this paper consists of 6363 defects in total, 2938 defects in company A, 554 defects in company B and 2871 defects in company C. The amount of defects is fewer than in our earlier studies [6, 7] because in this study we chose the newest defects (from 2008 to 2011) from the defect databases. The total amount of functional defects is 3799, 1826 defects in company A, 185 in company B and 1788 in company C.

## 3 The SAWO Defect Classification Scheme

In this section we present the SAWO defect classification scheme. The initial version of the scheme was developed in 2012 and is presented in [7]. In this paper we present an improved version of the scheme. In addition, the scheme has been validated with two new companies' defect data. The main difference to the one presented in [7] is that the scheme classifies defects on three levels presented in three different tables. This makes the scheme easier to understand and apply. In addition to improving the representation style, we have clarified the defect descriptions and changed one defect type's title (i.e. 6.6 Printing) and description to be more descriptive.

### 3.1 The SAWO Defect Classification Scheme Explained

The SAWO defect classification scheme classifies defects on three levels. The first level of the scheme is a combination of the schemes by Beizer [8] and Humphrey [9] and it divides defects in ten types. The second level of the scheme is applied to classify the functional defects, i.e. defect type Function is divided into sub-types. The second level adapts Beizer's functional defect classification. It consists of six defect types. The third level of the scheme classifies Feature/Function correctness defects in more detail. The third level has been developed and validated with our case organizations. There are six defect types in the third level. The structure of the SAWO defect classification scheme is depicted in figure 1.

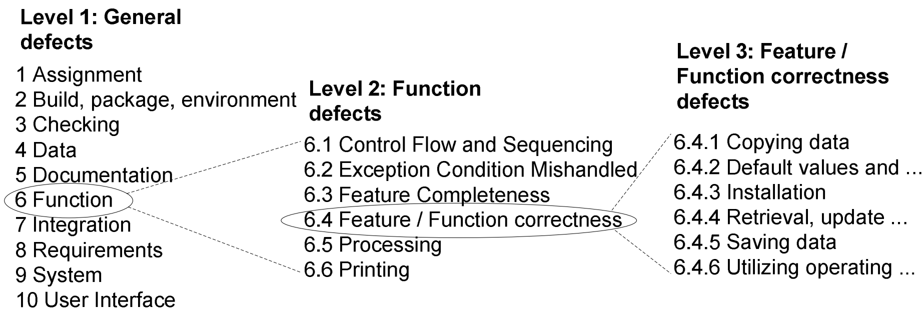



**Fig. 1.** The structure of the SAWO defect classification scheme

The three levels of the SAWO defect classification scheme are described in detail in tables 2-4, levels 1-3 respectively.

**Table 2.** SAWO defect classification scheme, level 1

| ID | Defect Type | Description |
|---|---|---|
| **1** | Assignment | Declaration, duplicate names, scope, limits |
| **2** | Build, package, environment | Change management, library, version control |
| **3** | Checking | Error messages, inadequate checks |
| **4** | Data | Database structure and content |
| **5** | Documentation | Comments and messages |
| **6** | Function | Logic, pointers, loops, recursion, computation, function defects |
| **7** | Integration | Integration problems, component interface errors |
| **8** | Requirements | Misunderstood customer requirements |
| **9** | System | Configuration, timing, memory, hardware |
| **10** | User Interface | Procedure calls and references, I/O, user formats |

**Table 3.** SAWO defect classification scheme, level 2

| ID | Defect type | Description |
|---|---|---|
| **6.1** | Control Flow and Sequencing | Defects in control flow (e.g. path left out, unreachable code, improper nesting loops, loop termination criteria incorrect). |
| **6.2** | Exception Condition Mishandled | Defects in exception handling. Exception conditions are not correctly handled, wrong exception-handling mechanisms used. |
| **6.3** | Feature Completeness | Feature is executed inadequately. Missing feature, duplicated, overlapped feature. |
| **6.4** | Feature/Function correctness | Implementation of feature / function is incorrect. Feature not understood, feature interaction. |
| **6.5** | Processing | Defects in processing, calculations. Algorithmic, arithmetic expressions, initialization, cleanup, precision. |
| **6.6** | Printing | User messages are incorrect. Printing on screen / paper, defects in reports. |

**Table 4.** SAWO defect classification scheme, level 3

| 6.4 | Defect type | Description |
|---|---|---|
| **6.4.1** | Copying data | Defects in copying data between systems / databases. Difficulties in making backups. |
| **6.4.2** | Default values and initial states | Defects in programs default values e.g. programs default selection causes failures in software. |
| **6.4.3** | Installation | Problems during installation of the developed program. |
| **6.4.4** | Retrieval, update and removal of data | Relates to refreshing the screen. Data inputs from user doesn’t update properly to the screen. |
| **6.4.5** | Saving data | Data doesn’t save to system. Data can’t be saved when it should be possible or it can be saved when it shouldn't be able. |
| **6.4.6** | Utilizing operating system services | Problems related to operating systems (e.g. Windows), e.g. mouse commands, tab order, and other features provided by the OS. |

# 4 Applying The SAWO Scheme

In this section we present the results of the defect classification applying the SAWO scheme. Data from the years 2008-2011 is classified.

## 4.1 SAWO Defect Distribution, Level 1

We applied the SAWO defect classification scheme level 1, presented in table 2, for three software companies defect data consisting of 6363 defects. The result of the defect classification is presented in figure 2.

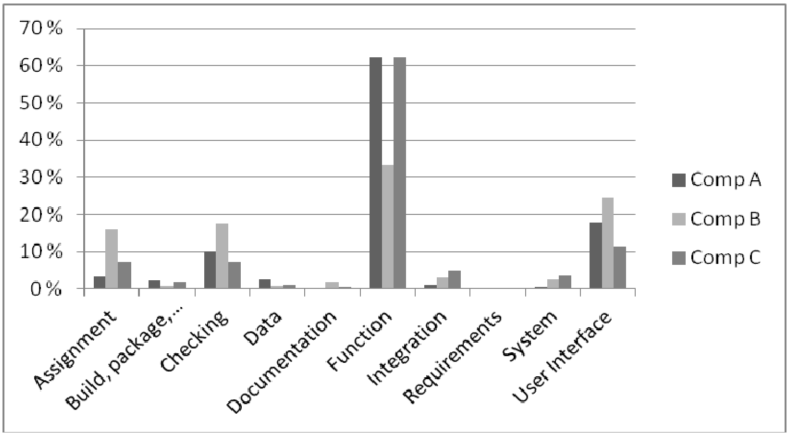



**Fig. 2.** SAWO defect distribution level 1

From figure 2, it can be seen that by far the most common defect type in every company is “Function” defect type (total of 3799, 59.7%). The second most common defect types are “User Interface” (total of 991 defects, 15.6%), and “Checking” (total of 601 defects, 9.4%). “Requirements” (total of 14 defects, 0.2%) and “Documentation” (total of 27 defects, 0.4%) are the rarest defect types.

## 4.2 SAWO Defect Distribution, Level 2

Despite the SAWO Level 1 made the problem points of the software processes visible, it is reasonable to study functional defects in more detail due to their large amount. Hence, we applied the SAWO defect classification scheme level 2 (see table 3) to classify the defects in a more precise manner. The classification was conducted for the functional defect data consisting of 3799 functional defects (see section 4.1). The defect distribution is presented in figure 3.

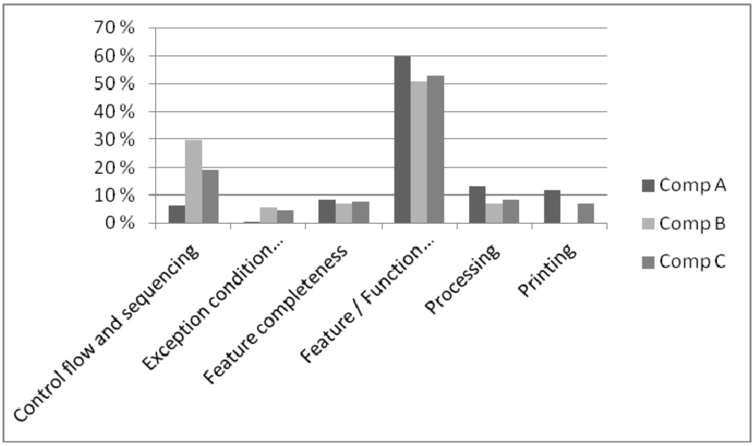



**Fig. 3.** SAWO defect distribution level 2 (Functional defects)

From figure 3, it can be seen that the defect type “Feature/Function correctness” is remarkably more common than the other defect types. “Feature/Function correctness” includes 56.3% of the defects (2139 defects in total). The second most common defect types are “Control flow and sequencing” (total of 512 defects, 13.5%) and “Processing” (total of 406 defects, 10.7%). “Exception condition mishandled” is the most uncommon defect type (2.6% of the defects).

## 4.3 SAWO Defect Distribution, Level 3

The case organizations wanted to find out what the “Feature/Function correctness” issues are, in order to improve their development and testing processes. In order to figure this out, we applied SAWO level 3 (see table 4) for the “Feature/Function correctness” defects, 2139 defects in total. The results can be seen in figure 4.

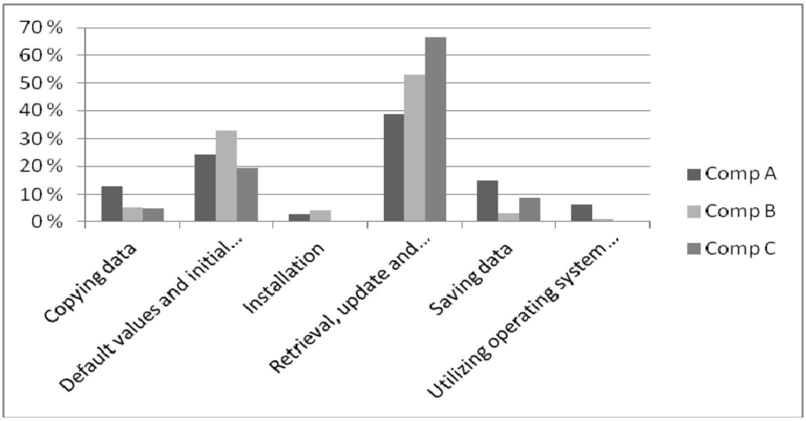



**Fig. 4.** SAWO defect distribution level 3 (Feature/Function correctness defects)

The most common defect type is “Retrieval, update and removal of data” (total of 1107 defects, 51.8%). The second most common defect types are “Default values and initial states” (total of 480 of defects, 22.4%) and “Saving data” (total of 249 of defects, 11.6%). “Installation” is the most uncommon defect type (only 1.6% of the defects).

# 5 Process Improvement Suggestions Based on SAWO

Applying SAWO, it was possible to provide practical and targeted improvement suggestions for the case organizations. The defect distributions of the case organizations vary, most likely because of the special characteristics of the organizations. Hence, the improvement suggestions are company-specific and offer solutions to current problems.

Based on the defect data classification, the case organizations are able to see their software engineering problem points from the defect point of view. The classification shows that the most common defects in all the case organizations are functional defects. In addition, user interface defects and checking defects are also common. When the functional defects are studied in more detail, it can be noticed that the most troublesome functional issues are related to retrieving, updating and removing data, default values of the variables and forms, and control flow and sequencing. The most common defect types of the case organizations are summarized in table 5. The amount of Function defects and Feature/Function correctness defects have been divided into sub defect types in SAWO levels 2 and 3.

**Table 5.** The summary of the most common defect types of all the defects in each company

| | **Company A** | **Company B** | **Company C** |
|---|---|---|---|
| **1** | 6 - Function | 6 - Function | 6 - Function |
| **2** | 6.4 - Feature/ Function correctness | 10 - User interface | 6.4 - Feature/Function correctness |
| **3** | 10 - User interface | 3 - Checking | 6.4.4 - Retrieval, update and removal of data |
| **4** | 6.4.4 - Retrieval, update and removal of data | 6.4 - Feature/ Function correctness | 6.1 - Control flow and sequencing |
| **5** | 3 - Checking | 1 - Assignment | 10 - User interface |
| **6** | 6.4.2 - Default values and initial states | 6.1 - Control flow and sequencing | 3 - Checking |

Based on the results of the SAWO defect classification detailed improvement suggestions were given to each case organization. The suggestions related to the most common defect types are presented in table 6.

**Table 6.** Improvement suggestions related to the most common defect types

| SAWO level 1 | | Company | | |
|---|---|---|---|---|
| **Defect type** | **Improvement suggestions** | **A** | **B** | **C** |
| Function | See SAWO level 2. | x | x | x |
| Feature/ Function correctness | See SAWO level 3. | x | x | x |
| User Interface | Conduct more thorough **user interface design and testing**. In addition, create **instructions for the user interface design,** review user interfaces, and provide developers with **check lists** for the important issues which must always be checked. Further, make use of **product family engineering approach** in which new interfaces/systems reuse common product family components. | x | x | x |
| Checking | **Error messages** should be more **accurate**. It must be checked in user interface testing if the error messages are relevant in all the situations. Use **uniform error messages** throughout the system. | x | x | x |
| Assignment | Conduct **code inspections** in order to reduce the amount of bugs due to carelessness. **Review also the requirements** and/or the specifications where the values of the parameters are derived from (see below Default values and initial states). | | x | |
| **SAWO level 2 (Function defects)** | | **A** | **B** | **C** |
| Control flow and sequencing | Conduct **code inspection**. Code coverage analysis and control flow tracking can be assisted with an **automated tool**. Complex systems should be **documented more properly**. Documentation helps in recalling the functionality when new version will be developed or new developers take charge of the project. | | x | x |
| **SAWO level 3 (Feature/Function correctness defects)** | | **A** | **B** | **C** |
| Retrieval, update and removal of data | **Stress the importance of unit testing**. Data retrieval, updating and deletion defects could be detected already in the unit testing phase during which it would be cheaper to fix them. **Conduct pair programming**. Previous research has found that programmers working in pairs produce fewer bugs, than programmers working alone [11]. Establish more precise **naming practice of the fields of the database** and improve the **database design process**. Check the content of the database after insert/delete/update operations. | x | | x |

**Table 6.** (*continued*)

| | | | | |
|---|---|---|---|---|
| Default values and initial states | Conduct **code inspections and pair programming**. **Take test automation in use.** Use automation tool for the management of the parameters. **Initial states and conditions must be defined in design phase** and they must be reviewed. If the **values of the parameters** are received outside the company (e.g. by legislation) they must be **transformed electronically** to the phase in which they are used (e.g. design, coding, inspection). | x | | |

## 6 Discussion

Our preliminary analysis with three software companies' defect data (11879 defects) showed that 65% of the defects were functional defects [6]. Even though the practical improvement suggestions could be given based on the preliminary analysis, a more detailed classification was needed because of the huge mass of functional defects. We wanted to find out what the real problems are behind these functional defects in order to enable process improvement based on defect data. Defect data is one of the most important available management information sources for SPI decisions [3]. Yet, defect data is rarely utilized properly in process improvement efforts [1].

In addition to the SAWO scheme, there are not many functional defect classification schemes available in the literature. The example of one can be found in [8]. Further, applying the defect taxonomies is somewhat challenging because the schemes are not detailed enough to enable the identification of tangible targets for process improvement.

The SAWO defect classification scheme provides a more detailed classification on the functional level than any of its predecessors. It has been developed to more accurately identify the problem areas of software process and to help software companies allocate improvement resources to justifiable targets. The first two levels, general level and the first functional level of the SAWO scheme are based on those presented by Humphrey and Beizer [9, 8]. We developed the third level of the SAWO based on our defect data analyses [6, 7]. We have applied our scheme for three software company's defect data and learned that the three levels of classification show differences between the defect distributions of the case organizations.

Based on the classification results presented, practical process improvement suggestions could be provided to the case organizations. The defect distributions clearly show that certain areas are more error prone than others. Hence, it is reasonable to allocate improvement resources to those areas. For example, all of the case organizations were suggested to conduct code inspections. This was suggested due to the fairly large amount of Assignment (6.1%), Control Flow and Sequencing (8.0%), and Default values and initial states (7.5%) defects. These defect types suggest that the programmers are a bit careless and might benefit from inspections where the quality of the code was monitored.

Further, there were also company-specific differences in the defect distributions. Hence, there were also differences in the improvement suggestions provided. For example, company A was suggested to conduct more thorough user interface design and testing and create instructions for the user interface design due to the large amount of User Interface defects (18.0%). Company B was suggested to check in user interface testing if the error messages are relevant in all situations. In addition, they were suggested to use uniform error messages throughout the system. This was suggested due to large amount of Checking defects (17.7%). Further, company C was suggested to improve the database design process and pay attention to the database naming conventions. The suggestion is related to the Retrieval, update and removal of data defects (22.0%).

There may be some limitations to this study. Firstly, we may have misinterpreted some defect descriptions due to lack of domain knowledge and language of the defect descriptions. In one case company the language of the defect descriptions is English while others used Finnish. English text of a non-native English-speaker may sometimes be difficult to understand and may cause misunderstandings. However, the amount of the defects in this study is so large that single misclassified defects do not influence the reliability of the results. Secondly, we do not know yet if the proposed improvement suggestions improve processes in the long run. However, it appears that case companies gained from the improvement suggestions: two companies initiated unit testing improvements and one company started to implement a test automation tool.

Based on the results of the study, SAWO would appear to provide practical input for SPI efforts by making the most problematic areas of software products visible. Further, SAWO enables the utilization of defect data, one of the most important data sources on which to base SPI decisions on. The three levels of SAWO enable us to understand software defects on a level of detail that appears not to have been possible with the existing defect classification schemes. SAWO is especially beneficial at the early stages of SPI projects when visible results are needed quickly to maintain motivation for SPI [12].

## 7 Conclusion

In this paper we present how defect classification can be applied as an input for software process improvement. The main contribution of the paper is the first official version of the SAWO classification scheme, initial version of which is first introduced in [7]. In addition, the SAWO scheme is validated via classifying three software companies' defect data. Further, based on the results of the defect data classification practical, company-specific process improvement suggestions are provided. Applying the SAWO scheme, the problem areas of software development and testing processes can be identified. Further, process improvement actions can be targeted to the real problem areas identified based on the defect data classification. The SAWO scheme enables software companies to utilize defect data, one of their most important, and nowadays perhaps least used management information sources for SPI decisions.

**Acknowledgements.** This research was funded by the Finnish Funding Agency for Technology and Innovation (Tekes) with grant 70030/10 for METRI (Metrics Based Failure Prevention in Software Engineering) project.

## References

1. Fredericks, M., Basili, V.: Using Defect Tracking and Analysis to Improve Software Quality. In: DoD Data & Analysis Center for Software, DACS (1998)
2. Vinter, O.: Experience-Based Approaches to Process Improvement. In: Proceedings of the 13th International Software Quality Week, San Francisco, USA (2000)
3. Grady, R.B.: Practical software metrics for project management and process improvement. Prentice Hall, New Jersey (1992)
4. Bhandari, I., Halliday, M.J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J.S., Lepori-Costello, C., Jasper, P.Y., Tarver, E.D., Lewis, C.C., Yonezawa, M.: In-process improvement through defect data interpretation. IBM Systems Journal 33(1), 182–214 (1994)
5. Freimut, B.: Developing and using defect classification schemes. Fraunhofer IESE IESE-Report No, 72 (2001)
6. Raninen, A., Toroi, T., Vainio, H., Ahonen, J.J.: Defect Data Analysis as Input for Software Process Improvement. In: Dieste, O., Jedlitschka, A., Juristo, N. (eds.) PROFES 2012. LNCS, vol. 7343, pp. 3–16. Springer, Heidelberg (2012)
7. Toroi, T., Raninen, A., Vainio, H.: Using Functional Defect Analysis as an Input for Software Process Improvement: Initial Results. Communications in Computer and Information Science 301, 181–192 (2012)
8. Beizer, B.: Software Testing Techniques. International Thomson Computer Press (1990)
9. Humphrey, W.: A discipline for software engineering. Addison-Wesley (2007)
10. Yin, R.K.: Case study research: Design and methods. Sage publications, INC. (2009)
11. Cockburn, A.: Williams. L.: The Costs and Benefits of Pair Programming. In: Succi, G., Marchesi, M. (eds.) Extreme Programming Examined, pp. 223–243 (2001)
12. Zahran, S.: Software process improvement: practical guidelines for business success. Addison-Wesley, Reading (1998)

# An Automated Approach for Fault Injection Testing of BPEL Orchestrations

Dessislava Petrova-Antonova[1,2], Sylvia Ilieva[1,2], Vera Stoyanova[2], Ilina Manova[3], and Valentin Pavlov[3]

[1] IICT-BAS, [2] Sofia University, [3] Rila Solutions
1113 Sofia, Bulgaria
d.petrova@fmi.uni-sofia.bg, sylvia@acad.bg, vera.stoyanova@yahoo.com, {ilinam,vpavlov}@rila.bg

**Abstract.** Web service orchestrations are widely adopted solution for development of loosely-coupled distributed applications. In addition to traditional defects causing failures in the software systems, their quality is also affected by additional problems such as network latency, interface inconsistency or communication issues. The fault injection testing is useful for validation the behavior of the web service orchestrations when such problems occur. That is why we propose an approach based on fault injection technique for generation and execution of fault tolerance test cases. The approach is automated through implementation of two software tools for fault injection and test case generation and execution Those tools are integrated in a common testing framework, called TASSA, presented in this paper. They are validated on a case study through simulation of different type of failures and fault tolerance testing of a web service orchestration.

**Keywords:** BPEL orchestrations, fault injection testing, web services, test automation.

## 1 Introduction

The wide adoption of Service-Oriented Architecture (SOA) and its web service based implementation brings new research challenges related to development of complex business processes and their subsequent testing [1]. Fault tolerance of such processes is critical since the proper work of many client applications depends on them. The business processes have to provide high-quality software services even if the execution conditions are abnormal and crashes occur.

The widely used testing tools and frameworks are dealing with logic and programming errors in the development process causing defects in the software systems. However, the software quality is affected also by additional external factors such as hardware crashes, communication interruption, unexpected user behavior and so on. A well known testing technique that can be applied in such situations is the fault injection. Its main objective is to test the fault tolerance through injecting faults and verify whether the software system can handle the faults and recover from them.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 84–95, 2013.

There are several challenges regarding business process testing, which affect the usage of fault injection technique:

- The complexity of the SOA architecture and therefore of its implementations – the typical service-based systems and in particular business processes integrate a large number of services, that are developed with various software technologies and provided by different providers. The usage context and the quality requirements of these services are not known at the time of development. That is why the testing process requires simulation of large number situations with unexpected condition that is possible through usage the fault injection technique.
- The latency of faults – Some faults can exist in the business process without causing failures. For example, the business process may invoke certain service returning invalid responses in a very few situations, which might not happen in a long time. Such situations are useful to be simulated through fault injection.
- The dynamic nature of SOA – The business processes based on SOA orchestrate services that are changed dynamically. The service providers can produce new versions of their services, temporally stop or even undeploy them. Additionally some of the consumed services can be replaced with new ones providing the same functionality. This in turn could cause unexpected behavior of the business process that can be simulated through fault injection.

This paper presents an approach for fault injection testing of business processes described with Business Process Execution Language for Web Services (WS-BPEL). It is implemented in two software tools, namely Fault Injection Tool (FIT) and Test Case Generation and Execution Tool (TCGET) that are parts of a common testing framework for BPEL orchestrations, called Testing as a Service Software Architecture (TASSA) [2]. The goal of the framework is to provide a set of tools and common platform for end-to-end testing of WS-BPEL orchestrations. It can be used in different phases of software process, namely implementation, integration, verification and validation, and maintenance phase.

The rest of the paper is organized as follows. Section 2 describes the FIT and TCGET. Section 3 presents a case study showing fault injection testing of a sample business process. Section 4 outlines the related work. Section 5 concludes the paper and gives directions for future work.

## 2 Tool Support for Fault Injection Testing of TASSA Framework

The TASSA framework provides a methodology and set of tools for functional as well as non-functional testing business processes described with WS-BPEL. The functional testing includes path coverage according to functional requirements, full branch coverage with shortest path length of the business process, and path coverage according to new functionality. The non-functional testing is achieved through robustness and scalability testing. The robustness testing is conducted by simulations of possible failures of unreliable service(s) and tests compositions of both reliable and unreliable services. The scalability testing covers many configuration changes.

### 2.1 Fault Injection Tool

The main function of FIT is to make modifications in a way that the BPEL process can be tested against fault injection. To provide a mechanism for such testing the FIT modifies the communication channel between the BPEL process and its partner web services as depicted in Fig 1.

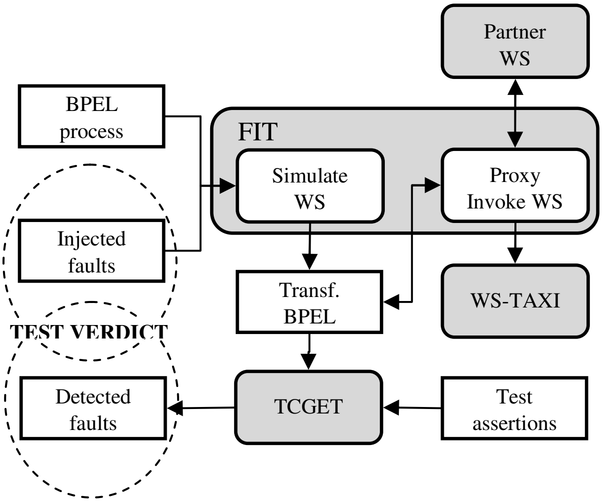



**Fig. 1.** Fault Injection Approach

FIT gets as input the original BPEL process and a set of configuration parameters describing the injected faults. Then, it transforms the BPEL process to a new one using a Simulate web service. During transformation the invoke activity where the fault will be injected is changed so that the corresponding message is sent to a Proxy Invoke web service instead to the process's partner web service. The Proxy Invoke web service is created by FIT between the two parties of the communication. While the original BPEL process communicates directly with the partner web service through message exchange, the transformed BPEL process calls Proxy Invoke web service to transmit the request message to its original receiver or to another web service, called TAXI-WS [3]. Then the FIT simulates a failure and deliver the response message to the transformed BPEL process. The following faults can be simulated by FIT:

- Message delay;
- Noise in the communication channel;
- Interruption of the communication channel;
- Wrong business logic.

In case of message delay, noise and interruption of the communication channel IPWS calls the partner web service, injecting the faults in the response message generated from it. In case of wrong business logic it calls the TAXI-WS. TAXI-WS is developed by a research team of Software Engineering Research Laboratory at the

ISTI (Istituto di Scienza e Tecnologie dell'Informazione) in Pisa. WS-TAXI generates compliant XML instances from a given XML Schema by using well-known Category Partition technique. The data in the XML instances could be randomly generated or predefined. Using WS-TAXI, the Proxy Invoke web service returns to the BPEL process a message with modified values, sufficient to testing its robustness. Thus, the tester is allowed to put the BPEL process under a variety of test cases. When random data are generated, a randomly distributed coverage of the service responses space is provided. If predefined, intelligent data are specified an invalid or boundary values are covered.

### 2.2 Test Case Generation and Execution Tool

The TCGET is a solution for WSDL-based testing of both single and composite web services, described with WS-BPEL. It automatically identifies the web service operations as well BPEL variables in case of composite web services testing. During test case generation TCGET provides SOAP request templates that are used for sending request messages to the web service under test. After test case execution it shows the response messages from the web service as well as the content of the BPEL variables if the web service is composite one. The test data can be manually defined or obtained from a data source. Currently, TCGET supports CSV and JDBC data sources. Its relation with FIT is shown in Fig. 1.

TCGET tool provides assertions at HTTP, SOAP and BPEL variable level. The supported assertions are as follows:

- HTTP Status Code – verifies the status code of the HTTP response;
- Response Time – verifies that the response is received in a given period of time;
- SOAP Fault – verifies that the HTTP response contains a valid SOAP Fault message;
- SOAP response valid – verifies that the HTTP response contains a valid SOAP message;
- XPath Equals – verifies that when a certain XPath expression applied to a SOAP body or a BPEL variable, its result is equal to a particular value. It is possible to make it case insensitive and/or match it with regular expression.
- XPath Exists – verifies if a certain XPath expression applied to a SOAP body or a BPEL variable is not empty;
- Contains – verifies that a certain string exists within a SOAP body or a BPEL variable.

In addition, the TCGET tool provides Negated assertion, which can be applied to all assertions defined above.

## 3 Case Study

This section presents the results from testing a sample composite web service, which calculates the area limited by a rectangle and ellipse inscribed in it. The business process under test (BPUT) has three partner web services and five invoke activities through which the web services' operations are invoked. The *Rectangle* web service and *Ellipse* web service compute respectively the area of a rectangle and an ellipse. The *Calculator* web service provides operations for summation, multiplication, subtraction, division and square of integer and double numbers. Fig. 2 shows the graphical representation of the WS-BPEL description of the BPUT before injection of faults.

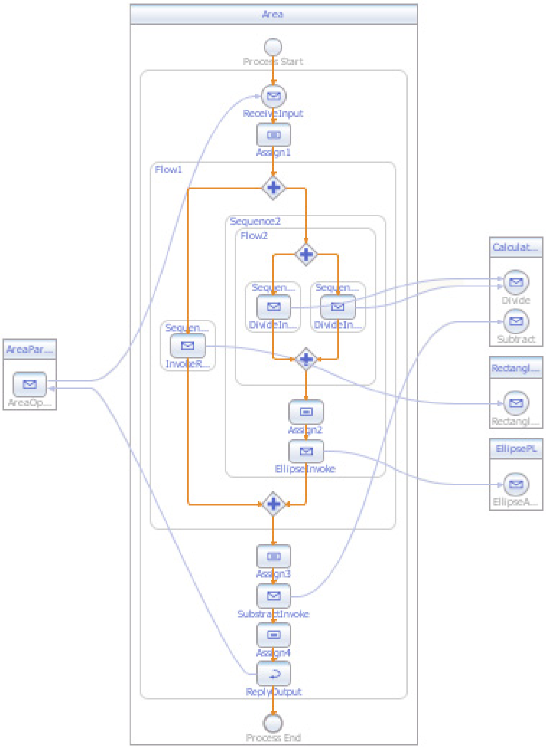



**Fig. 2.** The business process under test

### 3.1 Fault Injection

The BPUT is transformed so that the invocation of *Rectangle* web service is replaced with invocation of *Proxy Invoke* web service. The transformation is provided by the FIT, which requires the following configuration data that describes the simulated failure as follows:

- Wait interval – an integer value that defines the delay of message seconds in seconds;
- Error factor – an integer value that that defines the kind of error will be injected (1÷100: insert random errors in the data, which would possible break the XML structure; 0: usually used with Wait interval to delay the message; - 1: replace the original values in the message; -2: interrupt the message)

- End point address – an end point address of the partner web service;
- Activity variables – input and output variables of the activity that will be injected. Sample configuration data are presented in Table 1.

**Table 1.** Configuration data

| Data | Description |
|---|---|
| wait=0 | Error factor |
| error_ratio=40 | Wait interval |
| http://localhost:8022/RectangleService.asmx?wsdl | End point address |
| $RectangleAreaIn.parameters=$RectangleAreaOut.parameters | Activity variables |

During transformation of the BPUT, the invoke activity of *Rectangle* web service is replaced with two assign activities and one invoke activity of *Proxy Invoke* web service. Listing 1 shows the invoke activity that is responsible for invocation of Rectangle web service in the original BPEL process.

**Listing 1.** Invoke activity of Rectangle web service

```
<invoke name="InvokeRectangle"
partnerLink="RectanglePL" operation="RectangleArea"
xmlns:impl="http://rectangle.bple.tps"
portType="tns:RectangleSoap"
inputVariable="RectangleAreaIn"
outputVariable="RectangleAreaOut"
xmlns:tns="http://Rectangle.org/"/>
```

Listing 2 shows the transformation of the invoke activity of *Rectangle* web service after execution of FIT.

**Listing 2.** Fault injection during invocation of Rectangle web service

```
<assign name="Assign5">
<copy>
<from>sxxf:doMarshal($RectangleAreaIn.parameters)</from>
<to>$ProxyInvokeOperationIn.operationIn/tassaP:part1</to>
</copy><copy>
<from>'http://localhost:8022/RectangleService.asmx?wsdl'
</from>
<to>$ProxyInvokeOperationIn.operationIn/tassaP:endpoint</to>
</copy><copy><from>0</from>
<to>$ProxyInvokeOperationIn.operationIn/tassaP:wait</to>
</copy>
<copy><from>40</from>
<to>$ProxyInvokeOperationIn.operationIn/tassaP:errorsFactor</to>
</copy>
</assign><invoke xmlns:tns="http://www.rila.com/tassa/ProxyInvoke"
inputVariable="ProxyInvokeOperationIn"
name="InvokeRectangle" operation="ProxyInvokeOperation"
outputVariable="ProxyInvokeOperationOut" partnerLink="PartnerLink1"
```

```
portType="tns:ProxyInvokePortType"/>
<assign name="Assign6">
<copy>
<from>sxxf:doUnMarshal($ProxyInvokeOperationOut.part2)</from>
<to part="parameters" variable="RectangleAreaOut"/>
</copy>
</assign>
```

As can be seen from Listing 1 and Listing 2, the invoke activity of *Rectangle* web service, named *InvokeRectangle*, is enclosed with two additional assign activities, named Assign 5 and Assign 6. The first assign activity initializes the input parameters of the corresponding operation of *ProxyInvoke* web service as follows:

- Serialized input arguments of Area operation of *Rectangle* web service;
- End point address of the *Rectangle* web service;
- Wait interval initialized with 0;
- Error factor initialized with 40.

The second assign activity copies deserialized result from invocation of *Proxy Invoke* web service to the output variable of the Rectangle web service. In addition, *InvokeRectangle* activity invokes *Proxy Invoke* web service instead actual *Rectangle* web service.

### 3.2 Test Case Generation and Execution

FIT is used for generation of four test suites that correspond to the supported types of faults. Due to the limited space of the paper, each test suite is presented by a single test case as follows:

- Test Case 1: Message delay;
- Test Case 2: Interruption;
- Test Case 3: Invalid message data;
- Test Case 4: Wrong but valid message data.

To prove the fault injection against normal behavior of the BPUT first the standard use case should be observed. That is why an additional test case (Test Case 0) is defined showing the behavior of the BPUT when fault is missing.

The expected outputs from each test case are described in Table 2.

**Table 2.** Expected response messages from test cases

| Data | Description |
|---|---|
| Test case 0 | Meaningful, well formed message that is executed in t ms. |
| Test case 1 | Meaningful, well formed message that is executed in time interval t + T ms, where T is the delay injected by FIT. |
| Test case 2 | Error message due to interruption in communication channel. |
| Test case 3 | Error message due to invalid data in response message. |
| Test case 4 | Meaningful, well formed message with wrong data. |

The SOAP request generated by TCGET during test case execution is shown in Listing 3.

**Listing 3.** SOAP request to the BPUT

```
<soapenv:Envelope xsi:schemaLocation="..." xmlns:xsi="..."
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="..."
xmlns:area="...">
  <soapenv:Body>
    <area:AreaOperation>
       <a>30</a>
       <b>20</b>
    </area:AreaOperation>
  </soapenv:Body>
</soapenv:Envelope>
```

The SOAP response obtained by TCGET in case of normal behavior of the BPUT is shown in Listing 4.

**Listing 4.** SOAP response from the BPUT in case of missing faults.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="...">
  <SOAP-ENV:Body>
    <m:AreaOperationResponse xmlns:m="...">
       <result xmlns:msgns="...">128.99999999999994</result>
    </m:AreaOperationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 3 presents the assertions defined for the test cases. The goal of the test case execution is to verify whether the faults injected by FIT will be detected by TCGET.

**Table 3.** Assertions defined for the test cases

| Data | Description |
|---|---|
| A_1 | HTTP Status Code (200) |
| A_2 | Response Time = (600 ms) |
| A_3 | Not SOAP Fault |
| A_4 | SOAP response valid |
| A_5 | Response XPath Equals (xpath: " Envelope/ Body/AreaOperationResponse/result", value:" 128.99999999999994", case sensitive: "false", is regex "false") |
| A_6 | BPEL variable "RectangleAreaOut" XPath Equals (xpath: " message/part/RectangleAreaResponse/RectangleAreaResult", value:"600", case sensitive: "false", is regex "false") |

Table 4 presents the results for assertions after execution of the test cases. The test verdicts, which differ from those obtained during normal behavior testing, are in bold font.

**Table 4.** Assertion results after execution of test cases

| Data | TC0 | TC1 | TC2 | TC3 | TC4 |
|---|---|---|---|---|---|
| A_1 | Passed | Passed | **Failed** | **Failed** | Passed |
| A_2 | Passed | **Failed** | **Failed** | Passed | Passed |
| A_3 | Passed | Passed | **Failed** | **Failed** | Passed |
| A_4 | Passed | Passed | Passed | Passed | Passed |
| A_5 | Passed | Passed | **Failed** | **Failed** | **Failed** |
| A_6 | Passed | Passed | **Failed** | **Failed** | **Failed** |

**The test case TC1** simulates a delay of the response message from *Rectangle* web service. As it is shown in Table 1, only the assertion A_2 is failed. This assertion checks the response time of the BPUT, which is greater than 600 ms due to delay caused by *Proxy Invoke* web service.

**The test case TC2** simulates an interruption in the communication channel through missing response message from *Rectangle* web service. The results from test case execution show that all assertions except assertion A_4, failed. The response of the BPUT is a fault message shown in Listing 5.

**Listing 5.** SOAP response from the BPUT in case of missing response from Rectangle web service

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="...">
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Server</faultcode>
         <faultstring><![CDATA[BPCOR-6135: A fault was not
         handled in the process scope;...
         </faultstring>
         <faultactor>sun-bpel-engine</faultactor>
         <detail>
            <detailText> ...
            Caused by: HTTP transport error:
            java.net.ConnectException:
            Connection refused: connect
            Caused by: Connection refused: connect]]>
            </detailText>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The assertion A_1 is failed since the obtained HTTP status code of the response is 500 instead of 200. Since the BPUT waited to receive a response from *Rectangle* web service, the assertion A_2 is also failed. The reason for the failure of the assertion A_3 and assertion A_5 is obvious – the BPUT replies with fault message shown in Listing 5, containing information about the nature and the cause of the fault. The assertion A_6 is failed, since the BPEL variable *RectangeAreaOut* is not initialized due to missing result from invocation of the *Area* operation of *Rectangle* web service.

**The test case TC3** simulates an invalid response from *Rectangle* web service. The response from the BPUT is shown in Listing 6.

**Listing 6.** SOAP response from the BPUT in case of invalid response from Rectangle web service

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="...">
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Server</faultcode>
         <faultstring>...
           There is an error in XML document (1, 215) ---
           &amp;amp;gt; System.FormatException: Input string
           was not in a correct format. ...
         </faultstring>
         <faultactor>sun-bpel-engine</faultactor>
         <detail>
            <detailText>... Caused by: BPCOR-6131: An Error
            status was received while doing an invoke
            (partnerLink=CalculatorPL,
            portType={http://Calculator.org/}CalculatorSoap,
            operation=Subtract)...
         </detailText>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Again the assertion A_1 is failed since the obtained HTTP status code of the response is 500 instead of 200. The reason for the failure of the assertion A_3 and assertion A_5 is obvious – the BPUT replies with fault message shown in Listing 6, containing information about the nature and the cause of the fault. The fault details show that the invocation of the Subtract operation of *Calculator* web service is failed due to invalid input parameter. This is due to the simulated wrong response from *Rectangle* web service. What actually happens is that *Proxy Invoke* web service replaces the numeric result obtained by *Rectangle* web service with invalid string result. Thus, the BPEL variable *RectangleAreaOut* is initialized with incorrect data and its value is passed to the BPEL variable *SubtractIn* used for invocation of the *Subtract* operation of *Calculator* web service. This also explains the failure of the assertion A_6.

**The test case TC4** simulates a wrong response from *Rectangle* web service. The result received from that service has correct data type but its value is wrong. That is why the value of the BPEL variable *RectangleAreaOut* is not properly initialized, which in turn leads to the incorrect calculation of the final result obtained from the BPUT. Thus, the BPUT returns on time a valid SOAP response without faults as evidenced by the success of the assertion A_2, assertion A_3 and assertion A_4. However the values of the BPEL variable *RectangleAreaOut* and the numeric result in the response of the BPUT differ from the expected ones, causing the failure of assertion A_5 and assertion A_6.

## 4 Related Work

In this section we outline the approaches related to testing of BPEL processes. An extensive overview of the recently proposed approaches and tools for functional, structural and security testing of web services can be found in [4], whereas [5] surveys current proposals for testing web service compositions.

Concerning fault injection testing, existing approaches propose testing frameworks for injecting faults into service implementations in order to carry out white-box coverage testing of error recovery code [8] or to analyze the quality of the composed services in terms of fault tolerance capability [9]. A fault injection approach for testing of BPEL processes is presented in [11]. It uses web service stubs, which generate faults by simulating unexpected behavior of partner web services of the BPEL process. The work in [10], presents a framework for the generation and execution of robustness test cases for service compositions based on BPEL. Like proposed solution in Section 2, it takes into account invalid data and exceptional situations occurred in the internal logic and in the participant services. The framework differs from ours in that it does not change the BPEL specification but uses virtual services consisting of complex stubs with internal logic, to simulate the abnormal behavior or errors of the real services.

The authors of [11] define a model-based testing framework for web services composition. This framework, as the TASSA framework, consists of a set of tools that cover all different validation phases of web services compositions including robustness testing, but differently from TASSA it uses formal approaches for deriving test cases. Both frameworks include tools performing robustness testing by fault injection but these tools work very differently. The script driven tool, called WSInject, included in the framework of [11], intercepts all the SOAP messages exchanged between service partners and injects them with communication and interface faults according to dedicated rules that allow deleting, duplicate and corrupt messages or increasing the delay response. Differently from WSInject, the FIT of TASSA has a more complex behavior. It modifies the communication channel between the BPEL process and the partner services and transforms the original BPEL into a new one.

A general taxonomy of service oriented architecture faults is presented in [6], whereas a more specific one for service compositions is that of [7]. Differently from [10], [11] we addressed the fault categories presented in [7], focusing on the subcategories that are most interesting and likely to occur.

## 5 Conclusion

The presented work is focusing on fault injection testing of the web service orchestrations, described with WS-BPEL. While the traditional fault injection testing aims to detect problems causing by the end user or system components, fault injection testing of BPEL processes provide additional challenges due to dynamic nature of partner web services, fault latency, interface compatibility or interruption in the communication channel. Furthermore, the fault injection testing complements the other testing approaches. Even if orchestrated web services are free of defects and have stable interfaces, a network delay for example can cause their unexpected behaviour. That is why potential failures need to be simulated in order to provide robustness web service orchestration. To address such problem, we have developed TASSA framework, which can be used both for positive and negative testing.

In this paper, a case study that shows the usage of TASSA framework for fault injection testing is designed. It covers most likely to occur failures in the BPEL processes caused by message delay, interruption in communication channel and invalid or wrong message data. The obtained results show the efficiency of FIT to simulate failures as well as the possibility of TCGET to detect the causes of these failures. It is possible to extend TASSA framework to support other possible problems such as SLA violations or interface inconsistency. We also plan to prove the proposed approach against performance issues.

**Acknowledgments.** This work is supported by the National Scientific Fund, BMEYS Research Project, agreement n. DOO2-182.

## References

1. Morris, E., Anderson, W., Bala, S., Carney, D., Morley, J., Place, P., Simanta, S.: Testing in Service-Oriented Environments. Technical Report CMU/SEI-2010-TR-011, Carnegie Mellon University (2010)
2. Pavlov, V., Borisov, B., Ilieva, S., Petrova-Antonova, D.: Framework for Testing Service Compositions. In: 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing Timisoara, Romania, September 23-26, pp. 557–560 (2010)
3. Bartolini, C., Bertolino, A., Marchetti, E., Polini, A.: WS-TAXI: A WSDL-based testing tool for web services. In: International Conference on Software Testing Verification and Validation, Denver, Colorado, USA, pp. 326–335 (2009)
4. Bartolini, C., Bertolino, A., Lonetti, F., Marchetti, E.: Approaches to functional, structural and security SOA testing. In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions, pp. 381–401. IGI Global (2011)
5. Bucchiarone, A., Melgratti, H., Severoni, F.: Testing service composition. In: 8th Argentine Symposium on Software Engineering, ASSE 2007 (2007)
6. Bruning, S., Weissleder, S., Malek, M.: A fault taxonomy for service-oriented architecture. In: High Assurance Systems Engineering Symposium, pp. 367–368 (2007)
7. Chan, K., Bishop, J., Steyn, J., Baresi, L., Guinea, S.: A fault taxonomy for web service composition. In: Service-Oriented Computing - ICSOC 2007 Workshops (WESOA), pp. 363–375 (2007)
8. Fu, C., Ryder, B.G., Milanova, A., Wonnacott, D.: Testing of Java Web Services for Robustness. In: International Symposium on Software Testing and Analysis, pp. 23–34 (2004)
9. Fugini, M.G., Pernici, B., Ramoni, F.: Quality analysis of composed services through fault injection. Information Systems Frontiers 11, 227–239 (2009)
10. Kuk, S.H., Kim, H.S.: Robustness testing framework for web services composition. In: Asia-Pacific Services Computing Conference (APSCC), pp. 319–324 (2009)
11. Cavalli, A., Cao, T.-D., Mallouli, W., Martins, E., Sadovykh, A., Salva, S., Zai anddi, F.: WebMov: A Dedicated Framework for the Modelling and Testing of Web Services Composition. In: IEEE International Conference on Web Services (ICWS), pp. 377–384 (2010)
12. Wang, Y., Ishikawa, F., Honiden, S.: Business semantics centric reliability testing for web services in BPEL. In: Proceedings of the 6th World Congress on Services, SERVICES 2010, pp. 237–244. IEEE Computer Society, Los Alamitos (2010)

# A Framework to Support Software Quality Trade-Offs from a Process-Based Perspective

Gabriel Alberto García-Mireles[1], Mª Ángeles Moraga[2], Félix García[2], and Mario Piattini[2]

[1] Departmento de Matemáticas, Universidad de Sonora
Blvd. Encinas y Rosales s/n col. Centro, 83000 Hermosillo, Sonora, México
mireles@gauss.mat.uson.mx
[2] Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Paseo de la Universidad 4, 13071, Ciudad Real, España
{MariaAngeles.Moraga,Felix.Garcia,Mario.Piattini}@uclm.es

**Abstract.** Organizations are attempting to provide software that will meet stakeholders' quality requirements. Experts recognize that interactions between quality requirements might be conflictive. A trade-off study is an approach that can be carried out in order to resolve this issue. Since a trade-off study is a kind of decision process, we have reviewed the decision processes in CMMI and ISO/IEC 12207 in order to identify the process requirements. As we wished to deal with only one set of requirements, we have applied a harmonization technique whose results show that tasks of the ISO/IEC 12207 decision process could be embedded in practices from the CMMI decision process. We have then developed a proposal for a process framework to deal with these issues, which includes a trade-off quality process. We depict the elements taken into account to build the framework, and the trade-off process is presented at a generic level.

**Keywords:** harmonization, mapping, trade-off study, CMMI-DEV, ISO/IEC 12207, quality requirements conflict, decision process.

## 1 Introduction

Software quality is a fundamental feature that must be addressed throughout the product development life cycle. Software quality is defined as "the extent in which software has a combination of desired attributes" [1]. These desired attributes can be found in software product quality models such as ISO/IEC 9126, FURPS, McCall, and others. Indeed, one important goal in software development is to achieve a balance among the desired quality attributes [2], but some of them may be very difficult, or even impossible, to implement in the software product when they contradict each other. This situation is termed as conflict [3]. Some of the factors that may cause conflict among quality attributes to arise are the individual's perception of quality [4], inconsistency among quality models [5], and the lack of appropriate methods and techniques [6]. Software engineers must consequently carry out a trade-off study in order to balance quality requirements and build a better system [1]. A trade-off

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 96–107, 2013.

study is described as being a systematic approach through which to analyze the advantages and disadvantages of each proposed requirement or design alternative [7].

Various studies have addressed the question of software quality trade-offs. For instance, the WinWin approach considers the conflicting requirements and uses a tool to inform relevant stakeholders about the possible strategies that can be deemed to resolve the situation [2]. The NFR framework provides a modeling approach in which quality requirements can be modeled as softgoals and the mechanisms proposed to achieve them can also be included [8]. In a recent mapping study, Barney et al. [9] found diverse solution proposals that can be used to tackle software quality trade-offs. The results showed that the majority of papers are focused on methods that support architecture trade-offs. As a conclusion, the authors pointed out that the research area is still maturing [9].

The literature reviewed depicts a number of methods and techniques for use in dealing with software quality tradeoffs, but their scope is limited to a particular research area. The aim of this paper is to propose a framework to deal with software quality tradeoffs at earlier software product development phases aligned with CMMI [10] and ISO12207 [11]. In previous work, we carried out a harmonization effort in order to identify practices that support product quality characteristics, and discovered that process improvement models address them in the analysis and design phases [12]. Since a trade-off study is a kind of decision process, we reviewed the requirements for the decision process from CMMI and ISO12207 by applying a harmonization approach [13]. The requirements identified have contributed to the process framework presented herein.

The process framework supports the tailoring of quality models in order to refine the understanding of quality terms with regard to the kind of software product that an organization develops. It also provides support to deal with interactions among quality characteristics, and when negative interactions are identified in the software project, the framework provides a software quality trade-off process.

The remainder of this paper is structured as follows: Section 2 describes the works existing in literature that concern the decision-making approach. In Section 3 the comparison between the CMMI and ISO12207 processes is presented. Section 4 depicts the framework proposed to deal with software quality trade-offs. Finally, Section 5 shows our conclusions and future work.

## 2 Related Work

Many decision-making situations occur during software development. In practice, decision makers rely on their experience, attitude and intuition, and this depends on the context, such as the budget and the time available to make a decision [14]. In an empirical study concerning how software engineers make design decisions, Zannier et al. [15] reported that they can apply either a rational decision making approach when the problem under consideration is well-structured or a naturalistic decision making approach when the problem is perceived as ill-structured. Software project decisions can also be made at strategic, tactical and operational organizational levels [16].

Despite the importance of decision-making in software engineering, little empirical research has been reported [15]. Ruhe [17] summarizes the major concerns as regards decision-making and concludes that decisions are often poorly understood or described, made under time pressure, based on intuition, and consider only a few relevant stakeholders. Indeed, there are a variety of contextual factors that could affect software quality and these must be considered when making decisions [18].

Strategic and operational decisions concerning products, process, technologies or tools and other resources are far from being mature. Any stakeholder can perceive, interpret and evaluate the quality characteristics with regard to his/her own experience [4]. This subjectivity could produce conflicting quality requirements. Uncertainty and incompleteness are inherent characteristics of software quality requirements at the beginning of software development [19]. When conflicts emerge among quality requirements, software engineers should manage them. Indeed, as Robinson et al. [6] point out, requirements interaction management is a critical area.

Some proposals with which to manage conflicting quality requirements have also appeared, such as the WinWin approach [2], the NFR framework [8] and KAOS [20]. Various researchers have reported conflicting relationships among quality requirements [2, 3, 21-24].There are, however, different opinions as to the source of conflictive dependencies. Some authors have stated that conflict is inherent to a pair of quality requirements, while others emphasize that conflictive interactions depend on the software architecture and coding [23].

Several types of methods can be used to carry out a trade-off study, some of which depend on expert judgment, while others use semi-formal and formal models to compare alternatives [3]. The mapping study results in [9] reported that almost 50% (of 168 papers) deal with software architecture decision. Moreover, 25% of the papers address product quality and software process from a generic perspective. There are very few studies dealing with software coding and testing phases. The most common methods reported were the Analytical Hierarchy Process (AHP), model building, the Architecture Tradeoff Analysis Method (ATAM), algorithm-based and metric-based methods, expert opinion, Quality Function Deployment (QFD) and prototypes. At the analysis stage, the authors of the mapping study found specific techniques such as the Quality Performance (QUPER) model, prototyping and negotiation. With regard to the design stage, they reported additional techniques such as goals models, metrics, expert opinion and the automated construction of architecture alternatives. However, little empirical support is provided when software quality trade-offs are involved [9].

Software architecture trade-off methods have also been studied in order to understand the benefits and shortcomings of each one. Falessi et al. [25] compared decision-making techniques at the software design stage, taking into account the difficulties involved in using it. They found that there is no the best decision-making technique for the resolution of trade-offs in architecture design. In addition, Babar [26] proposed a framework with which to compare and evaluate various software architecture evaluation methods. Of the nine methods evaluated, only ATAM has the goal of analyzing trade-offs.

With regard to the development of decision-making processes based on ISO/IEC 12207 [11] or CMMI-Dev1.3 [10], we found that the Decision Analysis and Resolution (DAR) process area has been considered to define decision-making processes as regards the domains of both commercial-off-the-shelf (COTS) components and outsourcing companies. In the former category, Vantakavikran and Prompoon [27] described a process model with three layers, and they mapped each activity with DAR process area goals and practices. Phillips and Polen [28] described the Comparative Evaluation Process (CEP). They suggested set of criteria included the commonly evaluated characteristics (function, costs, maintainability and installation) and others that impact on management, architecture and strategic goals. They also considered contextual project factors and the credibility of data source. In the latter category, Hayshi [29], established a decision-making process with which to select outsourcing companies by considering the DAR process, and the AHP technique was used to prioritize criteria. The last author cited also classified criteria as being either absolute or relative in order to reduce the number of alternatives when they did not meet the absolute criteria.

Finally, with regard to the harmonization approach, Pino et al. [13] reported that the Decision Management (DM) process from ISO/IEC 12207 has a partial relationship with practices from the DAR process area, but did not include the details of this comparison. Indeed, a current research line in harmonization is that of mapping models to discover the common practices among models [13, 30]. However, we wish to enrich this research area by using these mappings to build new processes.

## 3 Comparison between CMMI and ISO/IEC 12207 Decision Processes

Harmonization is an approach whose goals consist of deploying diverse quality improvement models in organizations, optimizing resources, and simultaneously obtaining the expected benefits of each model and achieving business goals [31]. In particular, we are interested in the identification of requirements for the decision-making process based on CMMI [10] and ISO12207 [11] , in order to define a process whose goal is to support quality trade-off decisions. Requirements were extracted from models by carrying out a comparison between the decision processes of both models by adapting the techniques proposed in García-Mireles et. al. [12] and Pino et al.[13]. The activities involved in this comparison were:

1. Analyze models. The purpose of this is to understand the improvement models' goals, structure, and requirements. In this study, we describe the decision processes involved in the comparison.
2. Design mapping. The purpose of this is to set out a comparison procedure and to design mapping templates. We are interested in the details of the implementation

of quality requirements. We then consider additional information such as sub-practices, notes and process outcomes.
3. Execute mapping. The purpose of this is to apply a comparison procedure in order to achieve mapping results.
4. Establish quality requirements. The purpose of this is to report requirements identified with regard to each model and to propose a solution that will allow them to be integrated into a process definition.

### 3.1 Analyze Models

The Decision Analysis and Resolution (DAR) Process Area of CMMI-DEV1.3 belongs to the support category and appertains to the defined process level (maturity level 3). The process relies on a systematic evaluation process and established criteria to evaluate identified alternatives. Although it is recognized that uncertainty is one of the principal risks when making a decision, the model emphasizes a rationalistic approach to select, monitor and control either methods or selection criteria. The process also seeks new alternatives when the alternatives evaluated do not meet the stated requirements. The process area describes the specific goals, practices, sub-practices and exemplary of possible outcomes. There are also informative notes.

The Decision Management Process (DM), on the other hand, belongs to project processes within the main system context process group of the ISO/IEC 12207. The process analyzes project alternatives in order to then select one of them. The project must confirm that the preferred alternative resolves the issue that a request for a decision has identified. The process relies on the decision-making strategy to deal with decisions. The process specifies activities, tasks, and outcomes. There are also informative notes.

### 3.2 Design Mapping

We are interested in identifying common actions and the differences among process elements in order to provide support when an organization wishes to define a decision process to support trade-off studies. We use a matrix in which we can see the relationships between the elements of both models, and include the work products (outcomes) used or produced in each practice or activity. The heading of Table 1 depicts the template elements.

### 3.3 Execute Mapping

The results attained after comparing both models are presented in Table 1. We found that all the tasks and activities from the DM process are related to the DAR process practices. The processes' outcomes are aligned with DAR specific practices.

**Table 1.** Mapping between DAR and DM processes (Legend: SUB: sub-practices, SP: Specific practice; DP: Decision planning, DA: Decision analysis, DT: Decision tracking)

| CMMI specific practices | ISO/IEC 12207:2008 activities and tasks | CMMI outcomes | ISO12207 outcomes |
|---|---|---|---|
| **SP 1. Establish guidelines for decision analysis (100%)** | | Guidelines for when to apply a formal evaluation process | Decision-making strategy, applicable policies and procedures |
| SUB 1.<br>SUB 2. | DP Task 1<br>6.3.3.3 | | |
| **SP 2. Establish evaluation criteria (33%)** | | Documented evaluation criteria, rankings of criteria importance | Decision-making strategy |
| SUB 1.<br>SUB 2.<br>SUB 3, 4, 5,6 | DP Tasks 1, 2, 3<br>DA Task 1 | | |
| **SP 3. Identify alternative solutions (33%)** | | Documented evaluation criteria, identified alternatives | Alternative courses of action, decision-making strategy |
| SUB 1.<br>SUB 2.<br>SUB 3. | DP Task 3 | | |
| **SP. 4. Select evaluation methods (33%)** | | Selected evaluation methods | Decision-making strategy |
| SUB 1.<br>SUB 2, 3 | DA Task 1 | | |
| **SP 5. Evaluate alternative solutions (66%)** | | Documented evaluation criteria, rankings of criteria importance, identified alternatives, selected evaluation methods, evaluation results | Decision-making strategy, alternative courses of action, resolution, decision rationale and assumptions, (preferred courses of action) |
| SUB 1.<br>SUB 2.<br>SUB 3.<br>SUB 4, 5<br>SUB 6. | DA Task 2<br>DT Task 1<br>DT Task 1<br><br>DT Tasks 1, 2 | | |
| **SP 6. Select solutions (50%)** | | Recommended solutions | Resolution, decision rationale and assumptions, (preferred courses of action) |
| SUB 1.<br>SUB 2. | DT Task 2 | | |

### 3.4 Establish Quality Requirements

The comparison results show that there is an overlapping between the DM and DAR processes. Indeed, the task of the DM process can be 100% addressed by DAR practices. However, the DM process covers only 52% of DAR practices. The main differences between the process outcomes are the terms used to name informational elements in both models. DM includes a decision-making strategy as a basis to capture all relevant data related to carrying out a decision-making process, while the DAR process provides more details as regards dealing with process outcomes. Since the DM process can be embedded in the DAR process, we use the latter as a basis to describe process quality requirements. Although we rely on the DAR specific practices, the proposal can help to understand the activities required to support trade-off studies.

## 4 Framework to Support Software Quality Trade-Offs

The research presented in this paper has been conducted in the context of the industrial project MEDUSAS (Improvement and Evaluation of Software Maintainability, Security, Usability and Design) whose goal is to build an ISO25000-based environment to support quality control and quality management [32]. The project scope includes the assessment of both code and design models in order to determine the maintainability, security and usability of software products. The quality models were built in order to link terms, concepts, measures and heuristics to the aforementioned software quality characteristics.

One important issue that emerged once the quality models had been proposed was how to suitably manage the interactions between quality characteristics. This resulted in our proposal for a process framework that would manage the interactions between quality requirements. The goal is for the company to have a common definition of relevant quality characteristic terms, and their respective expected values, in order to support decision-making when conflictive interactions between quality requirements arise in the project. The framework is composed of conceptual, methodological and technological elements. This paper focuses on the methodological component in which the processes required to make a trade-off decision are delineated. Fig. 1 shows the principal processes, in addition to a repository of the models, methods and techniques used to perform trade-offs, and the dependence matrices among quality requirements.

The processes at the top of Fig. 1 correspond to the roles responsible for improving processes. These processes lead to the tailoring of a software quality model that is appropriate for the kind of software products that a company develops in which critical quality attributes are established and suitable measures are identified and evaluated. The establishment of a product quality goal process then takes place to diagnose the quality of both the company's products and those of the competition in order to set a benchmark to permit the identification of future quality levels of critical attributes, and the mechanisms employed to meet those quality levels are documented.

The processes depicted at the bottom of Fig. 1 support conflict management when software engineers are developing a software product. The first attempts to achieve a common understanding as regards the software quality vocabulary. The second process seeks potential conflicts among quality requirements. The process identifies these by using the interaction matrices to check dependencies among quality requirements. If a conflict is detected, then the software quality trade-off process is performed.

We present the workflow of the activities in the software quality trade-off study process (Fig. 2) which is the process at the bottom-left of Fig. 1. The activities correspond to DAR specific practices. The right-hand side of Fig. 2 shows the work products built throughout the process, while the left-hand side depicts the products built by the other framework processes, with the exception of negative interaction in which the problems among quality characteristics are documented in the MEDUSAS repository.

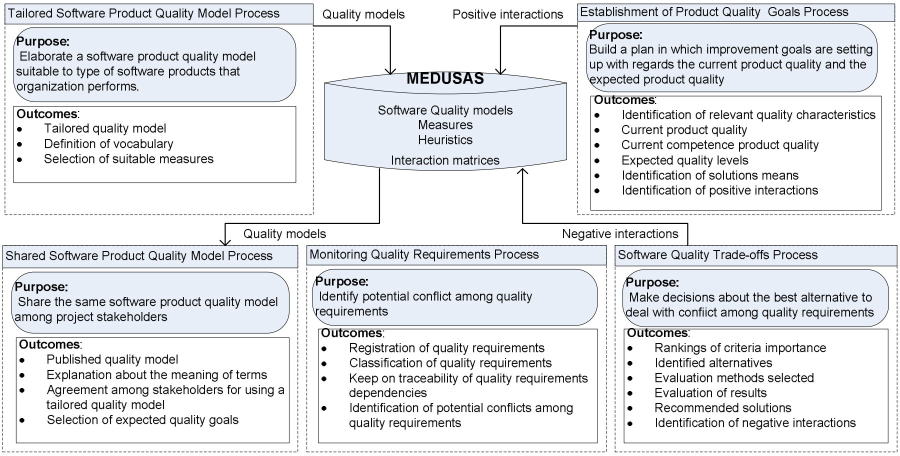



**Fig. 1.** Framework used to deal with software quality trade-offs

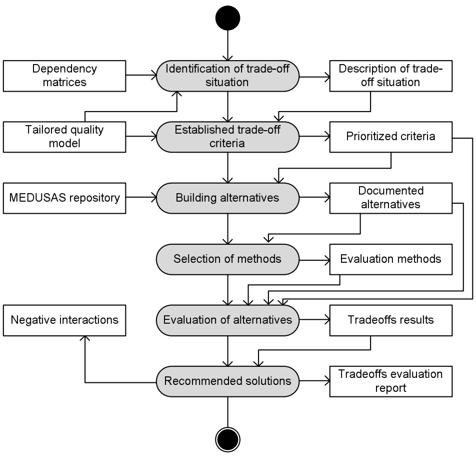



**Fig. 2.** Activities and products of software quality tradeoff process

Each activity is composed of several tasks. For instance, the establishment of trade-off criteria activity consists of the tasks described in Table 2. We also suggest some techniques that can be used to carry out these tasks. Maintainability, security and usability are the main quality characteristics that have been considered in MEDUSAS, and we have reviewed literature in order to identify conflictive interactions among them. We are currently surveying the project leaders from the companies that are working on this project in order to explore how they are dealing with conflicts between quality characteristics. Two companies are developing information systems and the other is developing a software tool. The code size is in the range of 1.4KLOC to 160KLOC and the projects are carried out by small teams. They reported the duration of the project to be in the range from 3 to 36 months. An excerpt of the contents of this survey, focusing on managing interactions, is shown in Annex A.

**Table 2.** Tasks from establishment of trade-off criteria activity

| Tasks | Products | Methods or techniques |
|---|---|---|
| Select quality characteristics from tailored quality models as criteria | Input: tailored quality models, description of trade-off situation Output: selected quality characteristics | GQM in order to identify goals from relevant stakeholder |
| Add additional criteria for evaluation of alternatives | Output: additional criteria | Meetings or interviews to add other criteria, e.g. costs and utility |
| Identify compulsory criteria | Output: compulsory criteria | Interviews or meetings to identify the quality criteria that are compulsory to all alternatives |
| Determine criteria' priorities | Output: weight of each criterion | This could be determined using the AHP technique |
| Pilot criteria and ranking | Output: result of testing criteria and ranking | Apply criteria in an alternative to verify that criteria are useful. Verify that criteria are traceable to requirements, business objectives or other sources |

## 5 Conclusions

If software development organizations are interested in improving their product quality, then they must consider how to deal with quality requirements, and particularly with any negative interactions among them. We have reviewed the main contributions from the quality requirement research area that are focused on the analysis and design stages of the software lifecycle. Although this research area is still in the development phase, software organizations should improve the management of dependencies among quality requirements.

In order to support software organizations, we have proposed a framework to deal with issues regarding quality attribute interactions. This framework relies on requirements extracted from process improvement models, such as CMMI and ISO/IEC 12207, and from issues to be considered when a project attempts to manage software product quality attributes. Since tasks from the DM process are covered by DAR process practices, as the harmonization task showed, we decided to take the latter as our primary source of requirements.

The process framework addresses the tailoring of product quality models and the establishment of quality goals. At a project level, the processes attempt to ensure that stakeholders maintain an agreed meaning of quality attributes, monitor potential negative interactions among quality attributes, and finally, perform the software quality trade-off supported by rational selection of the alternatives that lead to a balance among quality attributes. The framework maintains the data regarding quality attribute interactions in order to assist with decision-making, and particularly trade-off processes. We are currently studying how companies identify and resolve conflicts among quality requirements, in the context of MEDUSAS project.

As future work, we are considering whether the proposed framework will be sufficiently flexible to deal with different quality standards. The conformance requirements could also be used to classify diverse trade-off methods. In order to attain the benefits of interaction management, it is necessary to propose a method with which to capture the knowledge concerning both positive and negative types of interactions. Traceability, another important issue, must be implemented in order to manage interactions and dependencies. It is also necessary to analyze the trade-off methods and understand how they can be linked to the proposed framework. We are currently exploring companies' awareness as regards interactions among quality requirements in order to refine our proposal.

**Acknowledgments.** This work has been funded by the GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01) and PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN and FEDER, TIN2009-13718-C02-01).

## References

1. Barbacci, M., Klein, M., Longstaff, T., Weinstock, C.: Quality Attributes (CMU/SEI-95-TR-021) (1995), http://www.sei.cmu.edu/library/abstracts/reports/95tr021.cfm
2. Boehm, B., In, H.: Identifying quality-requirement conflicts. IEEE Software 13(2), 25–35 (1996)
3. Berander, P., et al.: Software Quality Attributes and trade-offs (2005), http://www.uio.no/studier/emner/matnat/ifi/INF5180/v10/undervisningsmateriale/reading-materials/p10/Software_quality_attributes.pdf
4. Paech, B., Kerkow, D.: Non-Functional Requirements Engineering - Quality is essential. In: Regnell, B., Kamsties, E., Gervasi, V. (eds.) 10th Anniversary International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ 2004), pp. 237–250 (2004)
5. Chung, L., do Prado Leite, J.C.S.: On non-functional requirements in software engineering. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S., et al. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 363–379. Springer, Heidelberg (2009)
6. Robinson, W.N., Pawlowski, S.D., Volkov, V.: Requirements Interaction Management. ACM Computing Surveys 35(2), 132–190 (2003)
7. Alexander, I.: Initial industrial experience of misuse cases in trade-off analysis. In: IEEE Joint International Conference on Requirements Engineering, pp. 61–68 (2002)

8. Chung, L., Nixon, B.A.: Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In: 17th International Conference on Software Engineering, pp. 25–37 (1995)
9. Barney, S., et al.: Software quality trade-offs: A systematic map. Information and Software Technology 54(7), 651–662 (2012)
10. CMMI, P.T. CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033) (2010), http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm
11. ISO, ISO/IEC 12207 Systems and software engineering — Software life cycle processes (2008)
12. García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M.: Towards the Harmonization of Process and Product Oriented Software Quality Approaches. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 133–144. Springer, Heidelberg (2012)
13. Pino, F.J., et al.: Mapping software acquisition practices from ISO 12207 and CMMI. Journal of Software Maintenance and Evolution: Research and Practice 22, 279–296 (2010)
14. Jedlitschka, A., Pfahl, D.: Towards Comprehensive Experience-Based Decision Support. In: Dingsøyr, T. (ed.) EuroSPI 2004. LNCS, vol. 3281, pp. 34–45. Springer, Heidelberg (2004)
15. Zannier, C., Chiasson, M., Maurer, F.: A model of design decision making based on empirical results of interviews with software designers. Information and Software Technology 49(6), 637–653 (2007)
16. Aurum, A., Wohlin, C.: The fundamental nature of requirements engineering activities as a decision-making process. Information and Software Technology 45(14), 945–954 (2003)
17. Ruhe, G.: Software Engineering Decision Support – A New Paradigm for Learning Software Organizations. In: Henninger, S., Maurer, F. (eds.) LSO 2003. LNCS, vol. 2640, pp. 104–113. Springer, Heidelberg (2003)
18. Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: Towards a comprehensive reference framework. Information and Software Technology 54(5), 433–447 (2012)
19. Ngo-The, A., Ruhe, G.: Decision Support in Requirements Engineering. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 267–286. Springer, Heidelberg (2005)
20. Van Lamsweerde, A., Darimont, R., Letier, E.: Managing conflicts in goal-driven requirements engineering. IEEE Trans. on Software Engineering. 24(11), 908–926 (1998)
21. Egyed, A., Grünbacher, P.: Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. IEEE Software 21(6), 50–58 (2004)
22. Zulzalil, H., Ghani, A., Selamat, M., Mahmod, R.: A Case Study to Identify Quality Attributes Relationships for Web-based Applications. IJCSNS International Journal of Computer Science and Network Security 8(11), 215–220 (2008)
23. Henningsson, K., Wohlin, C.: Understanding the Relations between Software Quality Attributes - A Survey Approach. In: 12th International Conference for Software Quality, Ottawa, Canada, pp. 1–12 (2002)
24. Sadana, V., Liu, X.F.: Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In: Society, I.C. (ed.) 31st Annual International Computer Software and Applications Conference, COMPSAC 2007, pp. 215–218 (2007)
25. Falessi, D., Cantone, G., Kazman, R., Kruchten, P.: Decision-making techniques for software architecture design: A comparative survey. ACM Comput. Surv. 43(4), 1–28 (2011)

26. Babar, M.A., Liming, Z., Jeffery, R.: A framework for classifying and comparing software architecture evaluation methods. In: Proceedings of Software Engineering Conference, Australian, pp. 309–318 (2004)
27. Vantakavikran, P., Prompoon, N.: Constructing a Process Model for Decision Analysis and Resolution on COTS Selection Issue of Capability Maturity Model Integration. In: 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2007, pp. 182–187 (2007)
28. Phillips, B.C., Polen, S.M.: Add decision analysis to your COTS selection process. CrossTalk, 21–25 (2002)
29. Hayshi, A.: Establish decision making process for selecting outsourcing company. In: 21 International Conference on Softwre Engineering and Knowlegde Engineering, SEKE 2009, pp. 666–671 (2009)
30. Pardo, C., et al.: From chaos to the systematic harmonization of multiple reference models: A harmonization framework applied in two case studies. Journal of Systems and Software 86(1), 125–143 (2013)
31. Siviy, J., Kirwan, P., Morley, J., Marino, L.: Maximizing your Process Improvement ROI through Harmonization (2008), http://www.sei.cmu.edu/library/assets/whitepapers/multimodelExecutive_wp_harmonizationROI_032008_v1.pdf
32. ISO, ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQauRE) - System and software quality models (2010)

## Annex A. Partial Questionnaire

Section 4 of this paper mentions a survey. This excerpt addresses interactions between quality characteristics.

1. What kind of dependency did you observe in each pair of quality attributes? Positive, negative, independent or unidentified?
2. What rationale was used to determine this type of dependency?
3. In which life cycle stage was the dependency identified?
4. What means were used to meet this pair of quality characteristics?
5. What measures did you apply to verify the dependency?
6. What elements did you consider to evaluate the dependency as negative?
7. What procedures did you use to resolve the negative dependency?
8. What was the impact of negative dependencies on quality product requirements?
9. What was the impact of negative dependencies on software design, coding and testing?
10. Which participants were involved in the negative dependency identification and conflict resolution?

# Discovering and Studying Collaboration Networks in Software Repositories

Andrejs Jermakovics, Alberto Sillitti, and Giancarlo Succi

Free University of Bolzano-Bozen
Piazza Domenicani 3, 39100, Bolzano-Bozen, Italy
{ajermakovics,asillitti,gsucci}@unibz.it

**Abstract.** Collaboration is important to software development processes and collaboration networks help us understand its structure and patterns. A common problem, however, is that these networks are not known and need to be discovered. In this work we study collaboration networks of five projects using an existing method that mines these networks from version control systems. The method is based on Recommender System techniques and finds similar developers by analyzing commits that are made to common files. These similarities are then used to automatically construct the network and it is visualized using a force directed graph layout algorithm. Two of the studied projects come from industry and are closed source while the other three are open source. In each study we learn some of the project's collaboration form and organization. We also were able to find various aspects of these projects that were previously not known.

**Keywords:** Collaboration networks, software repositories, visualization, social network analysis.

## 1 Introduction

Collaboration between developers is important in both closed source and open source projects and collaboration networks is one way of understanding its existing structure and patterns. In a development environment they can provide decision support for improving the software development process. Collaboration networks have already been widely used and been applied to developer networks for exploring collaboration [32], predicting faults [25], studying code transfer [22] and many other activities [3], [6], [29]. The analysis of these networks is often leveraged using visualizations and their appearance plays a significant role in how people interpret the networks [11], [20]. It is, therefore, important that the network visualizations are easy to interpret and represent the actual network as closely as possible.

In a lot of cases the actual social networks are not known and need to be discovered. Many existing approaches rely on communication archives to discover the networks [1], [5], [27]; however these sources are not always available. Another convenient source of developer networks is the Version Control Systems (VCS) [18], [11], [31]. The underlying idea is that frequent access and modification on the same

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 108–118, 2013.

code implies communication and sharing. The advantages of using VCS is that they are commonly available for all software development activities, can be mined automatically without human involvement and directly reflect collaboration on code.

In this work we study the collaboration networks of two closed source and three open source projects in order to learn their organizational structure and collaboration patterns. To do so we use an approach [15] that mines collaboration networks from version control systems and computes similarities between developers based on commits to common files. Once the similarities are computed the network is visualized using a force-directed graph layout. Due to non-disclosure agreements we omit the names of companies and their products.

## 2 Related Work

There has been a lot of research on the usage of social network analysis in software engineering. Since our work is focused on the discovery and visualization of developer networks, in the related work, we look at other approaches involving these aspects.

Similarly to our work Lopez-Fernandez et al. [18] collect developer links from CVS repositories and propose social network analysis for characterizing open-source projects. The difference is that they focus on analyzing characteristics of developer networks and not on analysis using visualizations.

Also Huang and Liu [11] use CVS repositories to mine developer networks and discover core and peripheral developers in open-source projects using distance centrality measure. A link between developers is defined if they contributed to the same module and, unlike our approach, the links are unweighted.

In Ariadne [31] a call-graph is extracted from Java programs and is annotated with author information to build sociograms of developers. Thus, the dependencies among modules are translated to dependencies of their corresponding authors and the strength of the developer dependency is based on code dependencies.

Tesseract [27] is a system for exploring socio-technical relationships and extracts developer links from version control systems. While force-directed layout is used here as well, the weight of links is not based on the number of changes but on the amount of communication between developers (represented by edge width).

In [1], Bird et al. analyze social networks of open-source project developers to determine patterns of organization. They extract social networks from email communication and then find communities by computing a graph partitioning that maximizes a modularity measure [23].

Ohira et al. [24] use collaborative filtering and social network visualizations to show developers of open-source projects. The difference between our work and theirs is that they use developer participation in projects to identify links, while we use file change history.

Augur [30] is a visualization tool that, together with other repository visualizations, shows a developer network in the form of a graph. The similarity between developers is based on changes to common CVS modules and is represented by the thickness of edges.

## 3 Network Discovery

The proposed approach [15], [16]. uses VCS to mine commits to source code files that developers make. It then computes similarities between committers and visualizes them in a network using similarities as link strengths. For our purposes, we adopted Cosine similarity measure, which is also used in Collaborative Filtering techniques [26, 28] of Recommender Systems and is similar to an approach for recommending software components [20].

The approach has been previously validated [15] where the structure was known and was able to discover actual developer networks. It is implemented in a software visualization tool Lagrein [14] which shows software metrics together with collaboration networks.

The network is laid out using Force-directed algorithms [8] that iteratively compute vertex positions until the difference between desired and actual distances is minimized. These algorithms are also a common choice for social network visualizations [10, 27].

## 4 Case Study: Company A

Our first industry partner is a Northern European company working in the financial sector. The company has two development teams of 20 people in total and is working on a large distributed system with a size of 500 KLOC. One of the teams is working on server-side components in C++ and the other on client-side components written in Java. While the teams work mostly on separate code, they also collaborate on integration of server and client side components. Thus we expect to see both teams as connected components in the visualization and some links between the teams. The data was extracted from their Subversion (SVN) repository for a period of last 2 years.

Since the developer networks are initially dense, the modularity [9, 23] is low. As the low similarity links are removed, the modularity gradually improves. The resulting network where all light edges are removed shows the two teams connected by few strong connections.

The interesting thing about this network that the nodes between the two teams are the ones that were working on client-server side code integration and thus have collaboration on both teams. Thus the visualization aids the understanding of the organizational structure in an intuitive way.

Another observation that we made was that the server side team writing C++ code (Team 2) has much stronger collaboration ties among its members than the Java team. One explanation for this is that Java IDEs such as Eclipse offer more powerful refactoring tools and thus when files are renamed or split it becomes harder to track changes to same files. Lastly we've noticed in some cases that collaboration was not as strong in the graph as we expected between some people. Upon consulting with the team we realized that developers collaborate using well-defined interfaces. These interfaces are rarely changed and most of the commits go to separate implementation classes.

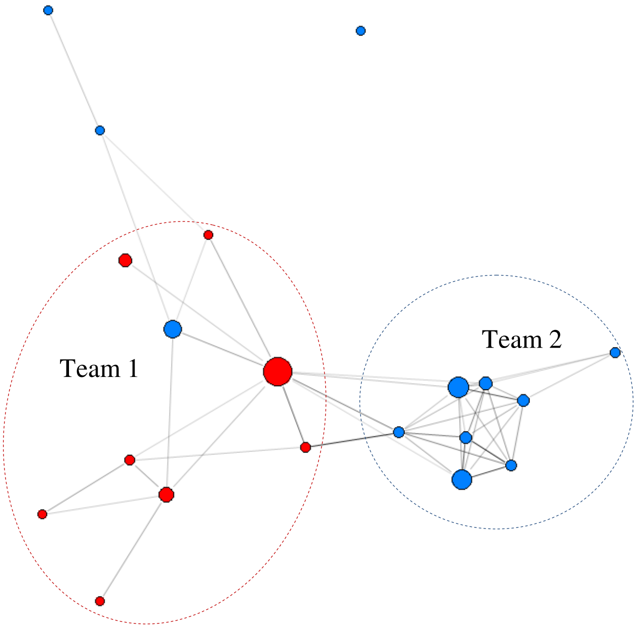



**Fig. 1.** Network using cosine similarity and edges filtering (Company A). The two teams become distinguishable

## 5 Case Study: Company B

Our second industry partner is an IT department of a large Italian company. Their main responsibilities are developing and maintaining custom software that is needed by other departments of the company written in C#, C++ and Visual Basic with a total size approaching 700 KLOC. There is a main development team, second smaller team and one remote team in another country. The main development team is composed of 20 developers, one senior developer acting as the team leader and one manager of the IT department.

Former collaboration with the company is described in separate work [3] and thus the existing team structure was known. Their development style is agile with a customized version of Extreme Programming and pair programming [7]. They also practice collective code ownership therefore we expect to see a lot of code sharing; however more code should be shared inside each team. The developer network was constructed from commit logs of Microsoft Team System VCS over a period of over 2 years and contains 52 committers. Similarly to Company A this network has a large number of very low weight links but after removing them we can see the three teams in the network.

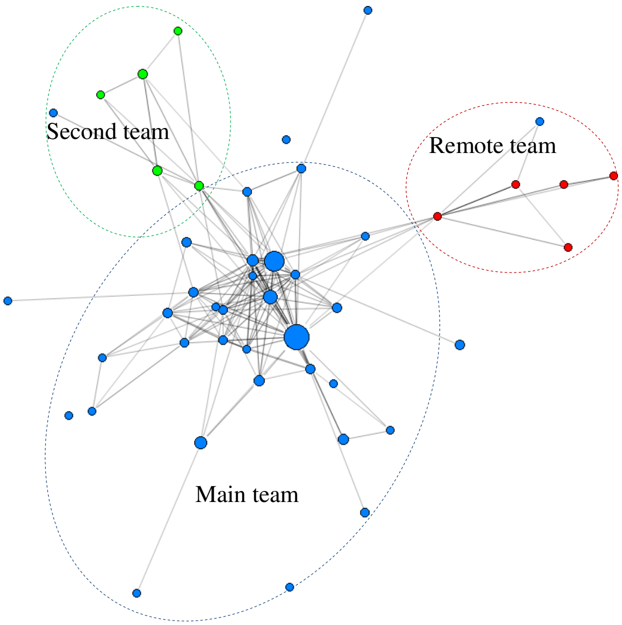



**Fig. 2.** Network using Cosine similarity and edge filtering (Company B). Teams become distinguishable

From the network we can immediately see that the main team is the largest and has strong collaboration ties among its members. The other teams have a lot less people and the links between them are sparser. One interesting observation that can be made is that the remote team collaborates much less with the main team than the Second local team. Although we expected all teams to be separate we initially did not expect such a great difference. These weak ties can be explained by the fact that the remote teams is in a different time zone and have to work on separate code to minimize reliance on the main team. Another surprising fact was the second team and the remote team almost doesn't collaborate on code, which was a fact we previously did not know.

## 6 Case Study: phpMyAdmin Project

Initially, we replicated a previous study [11] of the phpMyAdmin project to compare the resulting developer networks. Similarly to our approach, the developer network was extracted from the version control system, however the links were created when two developers committed to the same directory and all links had the same weight. With a network obtained using such approach the authors notice that it is impossible to determine the importance of each developer and conclude that all developers play the same role. They also mention that the network might be misleading due to link computation at directory level. We confirm this observation and discover a different structure in the project's network using our approach for the same period (until 2004). First we compute the links the same way – at directory level and without assigning weights to them. The resulting networks looks very similar to the one in the previous study and, indeed, no particular structure was evident. Afterwards, we apply the proposed approach and filter out most links (Fig. 3). We can notice that there are two main groups. By looking at the changed files, we noticed that these groups work on different sets of files, however most of these files are located in the root directory of the project. For this reason, link computation at directory level produced dense and compact network and we conclude that the computation is better performed at file level.

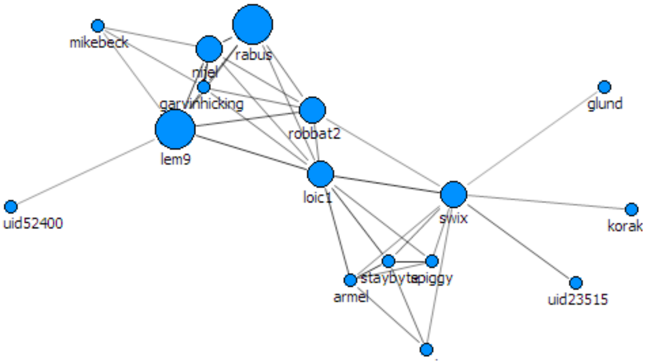



**Fig. 3.** phpMyAdmin collaboration network computed at file level. Some contributors appear having a more central role

While exploring the modified file list we noticed multiple developers which are the only committers to some files. They work on their own subset of files and no one else works on these files. They make many commits (large nodes) and also have many links to other developers indicating significant collaboration activity. Thus we conclude that all developers do not play the same role and there are some with more central and important roles. We later confirmed this by examining sourceforge.net and project's home page where several such developers (lem9, nijel, swix, loic1) are mentioned as project managers and maintainers.

## 7 Case Study: Eclipse DTP Project

Having experimentally selected Cosine similarity and link filtering as effective methods for discovering team structure we proceeded to apply the technique to the Eclipse Data Tools Platform (DTP) project. DTP is a set of tools for database handling and is currently part of Eclipse IDE distribution. The project is large (1.4 MLOC) and is composed of several subprojects: Connectivity, Enablement, Incubator, Model Base and SQL Development Tools. Using its CVS repository we constructed the developer network of 25 people for the period 2005-2010. The Eclipse project provides information on its committers using its Commits Explorer. This application allowed us to learn that contributions to the project have been made by many individual committers and multiple companies including Actuate, IBM, Red Hat and Sybase.

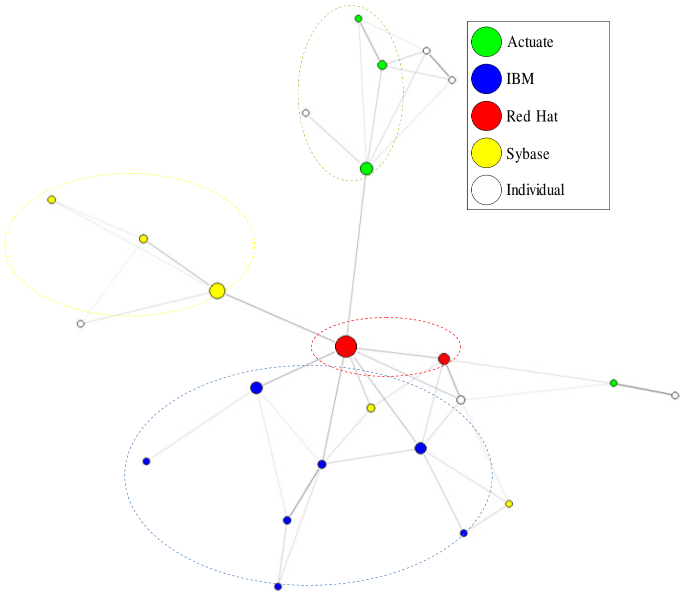



**Fig. 4.** Eclipse DTP project committer network. There is higher collaboration of contributors within each company

Fig. 4 shows the resulting network obtained with filtering and having each company colored in different color. The visualization of the network allows us to gain quick insight into the organizational structure of the project. Namely, contributors do not contribute equally to all parts of the project. They collaborate closely with other contributors from the same company and to a much lesser extent with contributors from other companies. This suggests that the collaboration is performed on specific subprojects.

## 8 Case Study: Gnu Compiler Collection (GCC) Project

The GNU Compiler Collection (GCC) project has a long history and a large number of contributors developing its numerous front-end and back-end projects. We extracted and analyzed their Subversion commit log in the period from 1988-2010 containing commits from 349 committers. This resulted in 100K commits and 8MLOC total codebase size. From the collaborator network (Fig. 5) we can immediately see a large and strongly connected core and a lot of scattered contributors in the periphery around the core.

A particularly interesting aspect is that this network also contains a strongly connected group separate from the core (marked red). By looking at the changes of this group, we can see that they are developing the Fortran front-end because most of their commits were to /gcc/fortran and /gcc/libgfortran directories. Thus we can discover that this community is rather closed because it mostly collaborates among its own members and to a much lesser extent with the rest of the GCC contributors. When we zoom in we can see the ARM architecture community (highlighted in yellow) to the left of Fortran community. One committer (pbrook) stands out in the middle between the ARM and the Fortran communities indicating a lot of involvement in both which we verified using the project's contributions page. Thus by viewing the network we are actually able to identify communities and roles. To summarize, by applying the method on open source projects we conclude that it is able to discover various aspects of the projects that were not evident before. We verified them using additional information from the projects however discovering using visualization involves much less effort.

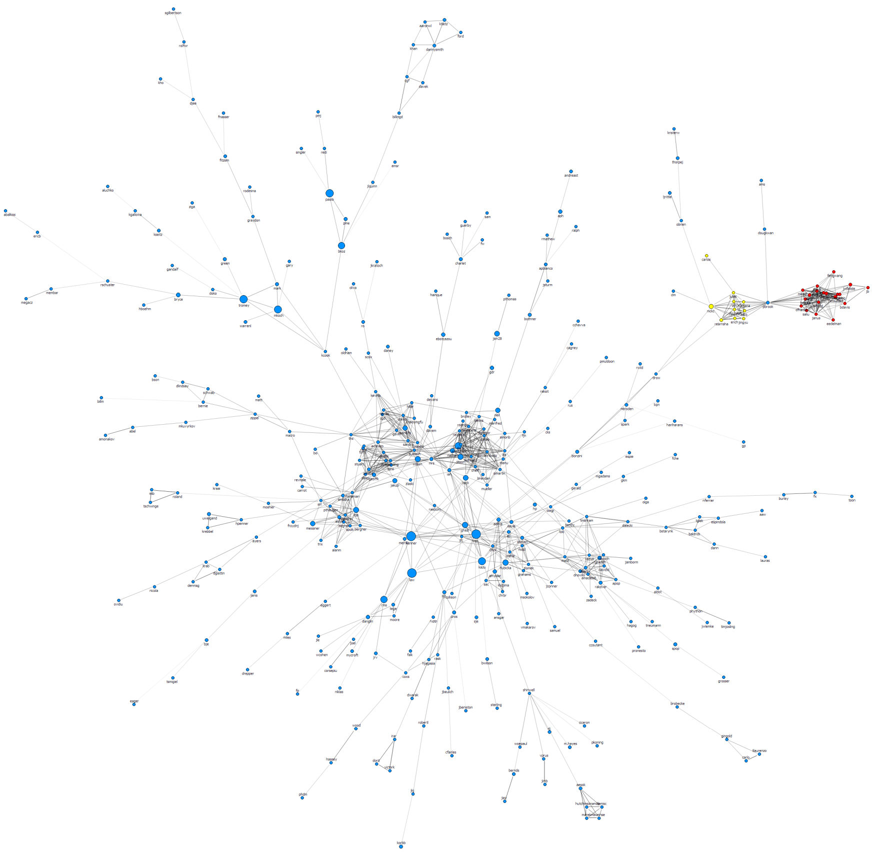


**Fig. 5.** GCC Collaboration Network. Fortran community is marked red

## 9 Conclusions

In this work we studied collaboration networks of two closed-source and three open-source projects using visualizations. The networks were automatically constructed using an approach that analyses software repositories and finds similarities among developers based on commit counts to common files. First we studied collaboration networks of two companies where we could identify teams and also find developers working on integrating work of these teams. In the other company we could see how the main team has less collaboration with a remote team and second team has no collaboration with the remote team. We found in phpMyAdmin project that there are developers with central roles and other contributors with peripheral roles. In Eclipse DTP project we noticed that there are contributions from multiple large companies however more collaboration is happening among developers within each company than between companies. Finally in GCC project we have observed that it consists of multiple sub-communities and that Fortran community is more separated from the other communities. Overall collaboration networks vary greatly in their organization and patterns however these are hard to discover since they are often not documented. Automatic approaches for constructing collaboration networks, as described here, can shed light on the structure of these projects and reveal aspects that were previously not known.

## References

1. Bird, C., Pattison, D., D'Souza, R.: Chapels in the Bazaar? Latent Social Structure in OSS. In: 16th ACM SigSoft International Symposium on the Foundations of Software Engineering, Atlanta, GA (2008)
2. Brandes, U., Gaertler, M., Wagner, D.: Experiments on graph clustering algorithms. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 568–579. Springer, Heidelberg (2003)
3. Coman, I., Sillitti, A.: An Empirical Exploratory Study on Inferring Developers' Activities from Low-Level Data. In: 19th International Conference on Software Engineering and Knowledge Engineering (SEKE 2007), Boston, MA, USA, July 9-11 (2007)
4. Coman, I.D., Sillitti, A., Succi, G.: A case-study on using an Automated In-process Software Engineering Measurement and Analysis system in an industrial environment. In: Proc. of the 2009 IEEE 31st International Conference on Software Engineering, May 16-24, pp. 89–99. IEEE Computer Society, Washington, DC (2009)
5. Crowston, K., Howison, J.: 2005. The Social Structure of Free and Open Source Software. First Monday 10(2) (2005)
6. Di Bella, E., Sillitti, A., Succi, G.: A multivariate classification of open source developers. Information Sciences 221, 72–83 (2013)
7. Fronza, I., Sillitti, A., Succi, G.: An interpretation of the results of the analysis of pair programming during novices integration in a team. In: 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009 (2009)
8. Fruchterman, T.M.G., Reingold, E.: Graph Drawing by Force-Directed Placement. Software-Practice and Experience 21, 1129–1164 (1991)
9. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002)
10. Heer, J., Boyd, D.: Vizster: Visualizing Online Social Networks. In: Proc. of the 2005 IEEE Symposium on Information Visualization, page 5. IEEE Computer Society, Washington, DC (2005)

11. Huang, S.-K., Liu, K.-M.: Mining version histories to verify the learning process of legitimate peripheral participants. In: Proceedings 2nd International Workshop on Mining Software Repositories (MSR 2005), pp. 84–88. ACM Press, New York (2005)
12. Huang, W., Hong, S., Eades, P.: How people read sociograms: a questionnaire study. In: Proc. of the 2006 Asia-Pacific Symposium on Information Visualisation, vol. 60, pp. 199–206. Australian Computer Society (2006)
13. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: KDD 2002: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543 (2002)
14. Jermakovics, A., Scotto, M., Sillitti, A., Succi, G.: Lagrein: Visualizing User Requirements and Development Effort. In: 15th International Conference on Program Comprehension (ICPC 2007), Banff, Canada (2007)
15. Jermakovics, A., Sillitti, A., Succi, G.: Mining and visualizing developer networks from version control systems. In: 4th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2011 (2011)
16. Jermakovics, A., Sillitti, A., Succi, G.: Exploring collaboration networks in open-source projects. In: 9th International Conference on Open Source Systems (OSS 2013), Koper, Slovenia, June 25-28 (2013)
17. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In: Proceedings of the American Mathematical Society, vol. 7(1), pp. 48–50 (1956)
18. Lopez-Fernandez, L., Robles, G., Gonzalez-Barahona, J.M.: Applying social network analysis to the information in CVS repositories. In: Proc. of 1st Intl. Workshop on Mining Software Repositories (MSR 2004), pp. 101–105 (2004)
19. Madey, G., Freeh, V., Tynan, R.: The open source software development phenomenon: An analysis based on social network theory. In: Proceedings of Americas Conference on Information Systems (AMCIS 2002), Dallas, US, pp. 1806–1813 (2002)
20. Mccarey, F., Cinnéide, M.Ó., Kushmerick, N.: Rascal: A Recommender Agent for Agile Reuse. Artif. Intell. Rev. 24(3-4), 253–276 (2005)
21. McGrath, C., Blythe, J., Krackhardt, D.: The effect of spatial arrangement on judgments and errors in interpreting graphs. Social Networks 19, 223–242 (1997)
22. Mockus, A.: Succession: Measuring transfer of code and developer productivity. In: Proc. of the 2009 IEEE 31st International Conference on Software Engineering, May 16-24, pp. 67–77. IEEE Computer Society, Washington, DC (2009)
23. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69(026113) (2004)
24. Ohira, M., Ohsugi, N., Ohoka, T., Matsumoto, K.: Accelerating cross-project knowledge collaboration using collaborative filtering and social networks. SIGSOFT Softw. Eng. Notes 30(4), 1–5 (2005)
25. Pinzger, M., Nagappan, N., Murphy, B.: Can developer-module networks predict failures? In: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, SIGSOFT 2008/FSE-16, Atlanta, Georgia, November 09-14, pp. 2–12. ACM, New York (2008)
26. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of CSCW 1994, Chapel Hill, NC (1994)

27. Sarma, A., Maccherone, L., Wagstrom, P., Herbsleb, J.: Tesseract: Interactive visual exploration of socio-technical relationships in software development. In: Proceedings of the 2009 IEEE 31st International Conference on Software Engineering, May 16-24, pp. 23–33. IEEE Computer Society, Washington, DC (2009)
28. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)
29. Sillitti, A., Succi, G., Vlasenko, J.: Understanding the Impact of Pair Programming on Developers Attention: A Case Study on a Large Industrial Experimentation. In: 34th International Conference on Software Engineering (ICSE 2012), Zurich, Switzerland, June 2-9 (2012)
30. de Souza, C., Froehlich, J., Dourish, P.: Seeking the source: software source code as a social and technical artifact. In: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2005, pp. 197–206. ACM, New York (2005)
31. de Souza, C.R., Quirk, S., Trainer, E., Redmiles, D.F.: Supporting collaborative software development through the visualization of socio-technical dependencies. In: Proceedings of the 2007 international ACM Conference on Supporting Group Work, GROUP 2007, pp. 147–156. ACM, New York (2007)
32. Wolf, T., Schröter, A., Damian, D., Panjer, L.D., Nguyen, T.H.: Mining Task-Based Social Networks to Explore Collaboration in Software Teams. IEEE Softw. 26(1), 58–66 (2009)

# Using Network Analysis to Discover Cooperation Opportunities in Inter-organizational Networks

Laura Ponisio[1], Pascal van Eck[2], Lourens Riemens[3], and Noriyuki Matsuda[4]

[1] BE Software Design, Amsterdam, The Netherlands
ml@ponisio.com
[2] University of Twente, Enschede, The Netherlands
pascal@pascalvaneck.com
[3] Dutch Tax and Customs Administration, Apeldoorn, The Netherlands
lj.riemens@belastingdienst.nl
[4] University of Tsukuba, Ibaraki, Japan
mazda@sk.tsukuba.ac.jp

**Abstract.** In a network of organizations, members are often faced with the problem of choosing partners for closer cooperation within this network. Consequently, network members collect information about potential partners to reach informed decisions about for instance starting new joint development projects or harvesting best practices. The large amounts of information involved in these decision processes obscure possibilities, and choices are made ad hoc. In this paper, we present an approach that uses techniques from network analysis to support organizations in processing and understanding this information. Central in our approach are network visualizations that help in comparing gaps between the aspired and current development levels of the processes of the member organizations. The advantage of our approach, which we validated via expert interviews, is that such visualizations are generated semi-automatically and offer an overall view of the current and aspired situation in the network without losing the ability to pinpoint particular, individual processes of interest.

## 1 Introduction

In today's networked world, organizations seek to improve their IT processes and systems by collaborating with partners in any network they participate in. For example, two organizations may jointly develop a shared online transaction processing system with the objective of reducing costs and harmonizing procedures. However, finding a partner for closer cooperation has several challenges [15]. Firstly, investigating possibilities for closer cooperation in all but the smallest networks results in too much information for decision makers to process in detail; which leads to arbitrary decisions. Secondly, this decision process is very costly and brings long-term consequences.

The purpose of this paper is to explore the potential of network analysis [6, 2] to find partners for closer cooperation with the final objective of improving IT processes and systems of the organizations involved. To this end, this study answers the following question: Is network analysis a useful instrument to find partners with whom to cooperate closer in an inter-organizational network? To answer this question, we put network

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 119–130, 2013.

analysis to test in the domain of eCustoms: a network of 27 European Union (EU) member states that have the same legislation, and have the objective to act as one monolithic customs organization in their relation with traders. Using an action research method in a case at the Dutch Tax and Customs Administration, we devised an approach based on network analysis theory to support choosing partners based on improving understanding of the current development level and improvement aspirations of the organizations.

Our results indicate that our approach has the potential to help eCustoms decision makers to find prospective alliances between customs organizations of EU countries. In fact, our network analysis method has, in the domain of eCustoms, the potential to build a workable solution to the problem of finding a partner. In this study, we show how network analysis techniques made it possible to build automatic visualizations that offer an overall view of the current and aspired situation in the network without loosing the ability to pinpoint particular, individual processes that would not be visible in traditional analysis methods. Having our results confirmed by expert interviews, we conclude that network analysis could be a useful instrument for revealing partners for closer cooperation. Moreover, our results indicate that software process improvement (SPI) may benefit from an improved understanding of organizations in an inter-organizational, alliance formation context.

## 2 SPI in an Inter-organizational Context

The fundamental objective of SPI is to change software development processes in order to achieve improvements in quality and productivity [1]. Several works have attempted to make an integrated body of literature [7, 9], while other research efforts focused on studying factors that influence success of SPI programs [11, 8]. Results assert the importance of adapting metrics programs to their context: to be successful, the programs implemented should be defined according to the focal organization's specific information needs [14]. In fact, research has clearly shown that success in SPI efforts depends on focusing on the needs of the organization in question [1, 13].

In practice we observe a newly emerging context for software process improvement, where organizations seek to jointly develop capabilities in symmetric relations, without a clear distinction between vendor and customer, and at least initially without the clarity provided by a formal contract. IT organizations form (strategic) alliances. Which mechanisms will organizations use to improve systems in the context of such alliances?

### 2.1 Alliance Formation

In finding partner for cooperation, organizations can follow two strategies: exploitation and exploration [16]. According to alliance theory [12], organizations need to be able to use both strategies (which is called 'ambidexterity'). Researchers have developed several measurement instruments to determine the strategic orientation of an organization. For instance, He and Wong [10] developed four Likert-scale items that characterize exploitation and four Likert-scale items that characterize exploration, related to objectives of innovation efforts while Popadiuk [20] uses a different technique to determine the strategic orientation of an organization. His measurement instrument illustrates the breadth of the ambidexterity concept.

Whereas such instruments are useful for empirical research, the increase in complexity of alliances calls for scalable methods that can deal with the many factors involved. In the area of inter-organizational (IO) networks, researchers have recognized the need for an integrative model of cooperation in an IO context and works that supports practice [5]. A survey of network analysis from the management perspective found that the research field is populated with insights derived from micro analysis, but that it was lacking a macro level that would explain processes in the whole network [12].

A case in point is eCustoms, a large distributed system that connects the customs organizations of a number of member states of the EU [19, 18]. Which strategy should member states follow to find partners for closer cooperation? According to He and Wong's objectives, eCustoms could follow an exploitation strategy to find partner for cooperation, for eCustoms seeks to reduce development costs and to learn from each other's best practices. But which concrete, practical techniques can eCustoms members use to determine the inter-organizational ability of potential partners for closer cooperation? To the best of our understanding, such techniques have not yet been explored.

## 3 Research Method

The methodological approach we used in this paper follows the guidelines of Baskerville [3] on methods for carrying out action research. We have chosen action research because we wanted to investigate whether a solution we propose proves to be useful when applied in an actual case in practice. This case is the eCustoms benchmark (see Section 3.1). Experts at the Dutch Customs Administration informed us about the challenges they were facing in finding partners for closer cooperation and shared with us the data of a benchmark performed by them.

The first step in our action research approach comprised problem investigation, which suggested the need for a method to give decision makers more insight about the eCustoms situation in terms of the current and aspired state of processes and supporting IT. As this can be very complex, the method should be semiautomatic and efficient, supporting the decision-making process with visualization and measurements. Consequently, the goal of our research was to define an approach with which a global view of eCustoms data can be built semi-automatically and to evaluate this approach in practice. Because network analysis has proven useful to find the underlying structure behind complex networks in the most diverse research areas [17, 4], we hypothesized that network analysis can be used as the basis for our approach.

The next step in our research was to design a network-analysis based approach with which visualizations of the eCustoms data can be constructed semi-automatically. The approach takes the form of a pipeline of data processing tools. The first stage of this pipeline consists of a program created by us that reads the benchmark data and generates a graph in the GML format, based on the representation introduced in Section 4. The second stage loads that graph in networkX (`networkx.github.com`) and runs some centrality measurements, gathering insight on each customs organization.

The final stage is displaying the visualization, for which we use Gephi (`http://gephi.org`) and YinfanHu's algorithm. We applied our approach to the eCustoms benchmark data to find potential partners for closer cooperation (Section 5) and discussed our insights in expert interviews to see if, in the perception of experts, the global view provided by our visualization is indeed easier to grasp than a long list of data provided by the conventional analysis techniques used in the benchmark project (Section 6).

### 3.1 The eCustoms Case

To study how our approach could be useful to find partners for closer cooperation in an existing inter-organizational network, we applied our method to a concrete instance of such a network: eCustoms. In this network of customs organizations, small groups of members try to improve their supporting IT processes and systems by closely cooperating with one another.

This network was established by the European Commission, aiming to improve ensuring safety of the external borders of the EU, and to facilitate trade. All customs organizations of the 27 member states of the EU belong to this network. eCustoms has been in place for many years, but new demands make it necessary to cooperate closely. Recognizing information is key to find opportunities for collaboration, the Dutch Tax and Customs Administration conducted between January and October 2005 a study that compared business processes, systems and ambitions of ten member states. The benchmarking study provided the data needed to analyze opportunities for cooperation.

### 3.2 The eCustoms Benchmark Data

In our research, we focused on the business process development aspect of the data, consisting of the current and aspired development levels of 53 processes distributed over areas such as declaration handling, inspections, post arrival processing, account management, risk management, collaboration and partnerships and enterprise services. Of these processes, we focused on the 42 that had quantitative answers (processes do not have consecutive numbers in our descriptions and figures because we left out the 11 qualitative items). For each of these processes, a local expert of the customs organization had completed the survey by supplying an assessment of a current and aspired level. The scale of levels was from one to five.

For example, The Netherlands (NL) assigned a five to the current level of process 28, *Calculate duties*, and a five to its aspired level. As both levels are equal to each other and to the highest level possible, NL thus indicated it sees no need to improve this process. Process 53, *Share information within the organization*, tells a different story: it has current level two and aspired level four, indicating interest of NL to improve it. This circumstance opens the door for decision makers to search collaboration with other customs organizations to join forces in improvement projects.

Traditional statistical analysis of the 42 processes with quantitative data revealed that not all processes have the same importance for the Dutch Customs Administration. We performed this traditional analysis in a phase in which we investigated the eCustoms case from many different angles (a phenomenon common to action research) before converging on network analysis. The areas NL are more interested in to develop are *Collaboration and partnerships* and *Shared information within the organization*. This insight is used in Section 5.2.

## 4 Network Approach for Analysis and Visualization

We constructed a directed network for each country using the 42 quantitative items of the survey as the *process nodes* $P_i, (i = 1, \ldots, 42)$. They are linked to the *level nodes* $L_j, (j = 1, \ldots, 5)$, according to the current and aspired ratings of the items. More specifically, edges from $L_j$ to $P_i$ and from $P_i$ to $L_k$ represent the current and aspired relationships, respectively.

In theory, $P_i$ is connected *from* $L_j$ and *to* $L_k$ in one of three ways: 1) $j = k$, which means that $P_i$ is bidirectionally connected to $L_j = L_k$ (in this case we say $P_i$ is a *static node*), 2) $j < k$, which means that the current level of $P_i$ is lower than its aspired level (we say $P_i$ is an *upgrading node)*, or 3) $j > k$, which means that the aspired level of $P_i$ is lower than its current level (we say $P_i$ is a *downgrading node*). This third possibility does not appear in our case study and we think it is very rare in practice.

**Example.** Figure 1 illustrates the application of our method to visualize 42 customs processes of NL. Each process is linked to its current and aspired levels, which are represented by five level nodes, making 47 nodes altogether. To facilitate reading, in these graphs a node is bigger if it is a level node. In our eCustoms examples, the labels of the process nodes range from 1 to 53 (rather than from 1 to 42) in correspondence with the numbers of the questionnaire items. As 11 qualitative items do not appear in the graphs, the numbers are not consecutive.

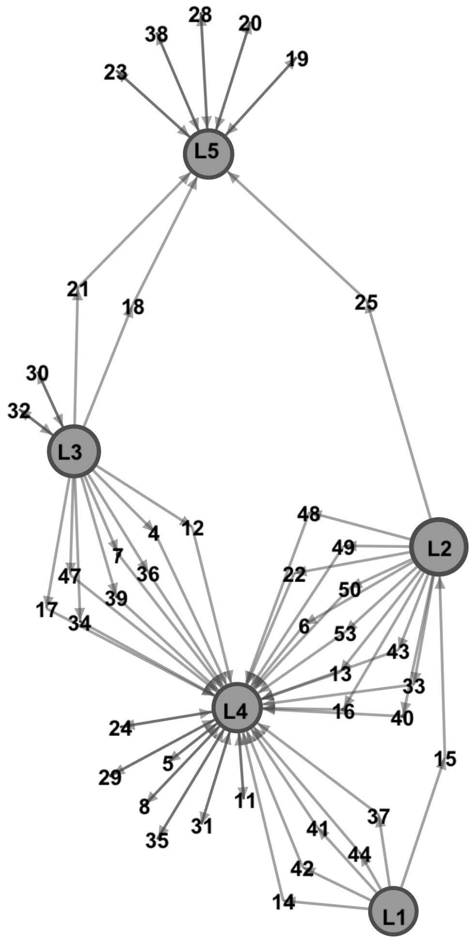



**Fig. 1.** Network representation of the survey results for NL

**Static Nodes.** Processes that are static nodes have reached their maximum level according to what the country finds interesting. At the top of the graph in Figure 1, we see a group around the node with label L5 (processes 19, 20, 23, 28 and, 38). These nodes indicate processes that are already developed (their current level is the same as the aspired level), and there is no need to search for a partner that can teach how to improve on these processes in a technology-transfer relationship.

Further verification revealed that almost all static processes in NL were processes that support the top three IT issues the Dutch Customs Administration was facing at the time of the benchmarking: 1) improve IT processing efficiency and productivity, 2) standardize IT environment, and 3) online government and e-services. For instance, process 28, *Calculate duties*, is essential for customs activities.

Process 30 (*Forecast revenue impact*) and 32 (*Link manifests with declarations*) are static nodes and at a lower level (three). The fact that they are static nodes indicates little interest of improvement, which is in line with the fact that these processes only slightly affect either transaction performance, or the top three IT issues the Dutch Customs Administration is facing.

In Figure 1, NL shows advanced *essential* customs processes. This is in line with what we observed of their customs performance, as NL has one of the biggest entry points in the EU (the Rotterdam harbor). The benchmark study showed that at the time of the benchmark, NL processed the second highest percentage of transit declarations per year in the whole sample, and therefore has high availability requirements. Without advanced essential customs processes, NL would be unable to have the required capacity and robustness.

In terms of opportunities for closer cooperation, the static-nodes-at-a-high-level pattern tells us a good partner for NL would have a high current level for these processes, as well.

**Upgrading Nodes.** Processes that are upgrading nodes have their current development level lower than their aspired level. Improvement is desired there. In Figure 1, these nodes are connected to a current level node that is lower than their aspired level node. Most processes are upgrading nodes: their current level is either L1, L2 or L3, and many have aspired level L4, which tells us most of the processes are desired to improve. A case in point is process 53, *Share information within the organization*. Located between L2 and L4, upgrading node 53 indicates NL is interested in improving this process.

The graph shown in Figure 1 suggests that experts need to focus on the processes linked to level 4 as their aspired level. This result suggests also to look for a symmetric partnership.

Process 25, *Administer the exit of goods*, in Figure 1 is exceptional: it is the only process NL wants to improve from 2 to 5 (the highest level). Its location in the graph draws the attention of decision makers to its uniqueness. This can be made formal using the network analysis concept of betweenness centrality. The graph shows also two other exceptions: processes 18, *Administer the entry of goods*, and 21, *Administer the transit of goods*. Together with process 25, these processes have in common that they belong to an essential area of NL and their aspired level is way higher than their current level. They all belong to the area *Declaration Handling*, an important area in customs that becomes crucial to improve when NL wants to be paper free and more efficient.

## 5 Visualization Findings: Potential Partners for Closer Cooperation

In this section, we show how our network approach for analysis and visualization supports the task of finding a partner for closer cooperation in several ways. Firstly, we show how potential partners or non-partners can be identified by inspecting the shape of their respective graphs (Section 5.1). We then zoom in on closer inspection of the graphs of countries identified as potential partners of NL (Section 5.2).

### 5.1 Comparing Countries

Two countries have good perspectives to succeed in an effort for closer cooperation if the current level and ambitions of their processes match. In order to match, two partners must have important processes (whichever those are for each country) developed to a similar extent. For new development, it is desirable that processes are at a similar level. To harvest best practices, it is desirable to look for a partner with a higher current level. According to eCustoms experts, "The bigger the overlap [in processes], the faster we can work.", and, "If processes don't fit, closer cooperation is going to be very difficult." Both partners must be interested in improving approximately the same set of processes, since closer cooperation requires similar interests to justify the investments needed. In terms of our visualizations, this means that we look for similar patterns around single processes, e.g., the occurrence of upgrading nodes in two graphs for the same process.

Thus, two countries have poor perspectives to succeed in an effort for closer cooperation if the current level and ambitions of their processes do not match. A case in point is shown in Figure 2, where Country A is very different from Country B. For instance, Country B has no process at level 1 (L1 is isolated at the bottom left) and Country B's most processes and systems are level 5 or level 4. In contrast, Country A has no such concentration of processes at high levels.

Conversely, Figure 2d and Figure 2c show an example of two countries with good prospects of closer cooperation: Country C and Country D have the same structure in term of current and aspired level of their customs processes.

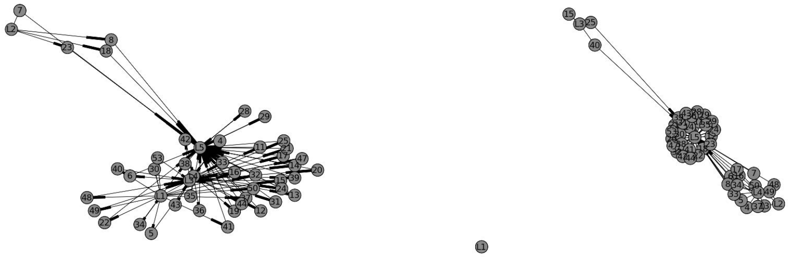
### 5.2 Partners under the Magnifier

When considering a network member as a partner for closer cooperation, it is particularly important to know how the potential partner performs in the processes that are essential to our own organization. A key question is: How advanced are our essential processes on their side? We used this reasoning to better understand which network members are better partners for NL.

Firstly, we need to know which processes are important for our organization. We found the most important processes for NL (Section 3.2) and highlighted them in the network representation of this country. Figure 3a shows a visualization of NL where the aspiration links for important processes are highlighted.

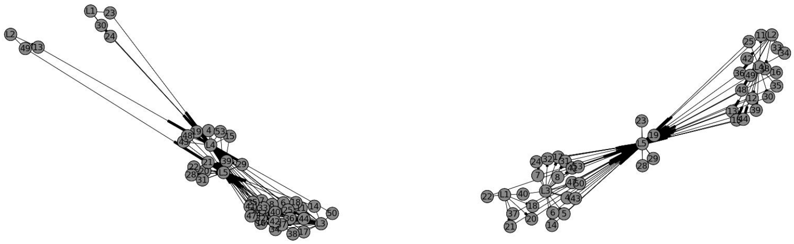
Secondly, we need to know how potential partners perform. To achieve this goal, we built the network representations of other customs organizations in the same way we have seen in Section 4 and highlighted in those graphs the processes that are important for NL. In the graph, an edge is bigger if it belongs to an area that NL is interested to develop. Figure 3 shows the resulting visualization of Country A and Country E.

**Country A.** Figure 3a and Figure 3b show that NL and Country A are interested on developing different things. Drawing Country A from the perspective of NL's needs,



(a) Country A (b) Country B

Two countries with poor prospects of closer cooperation



(c) Country C (d) Country D

Two countries with high prospects of closer cooperation

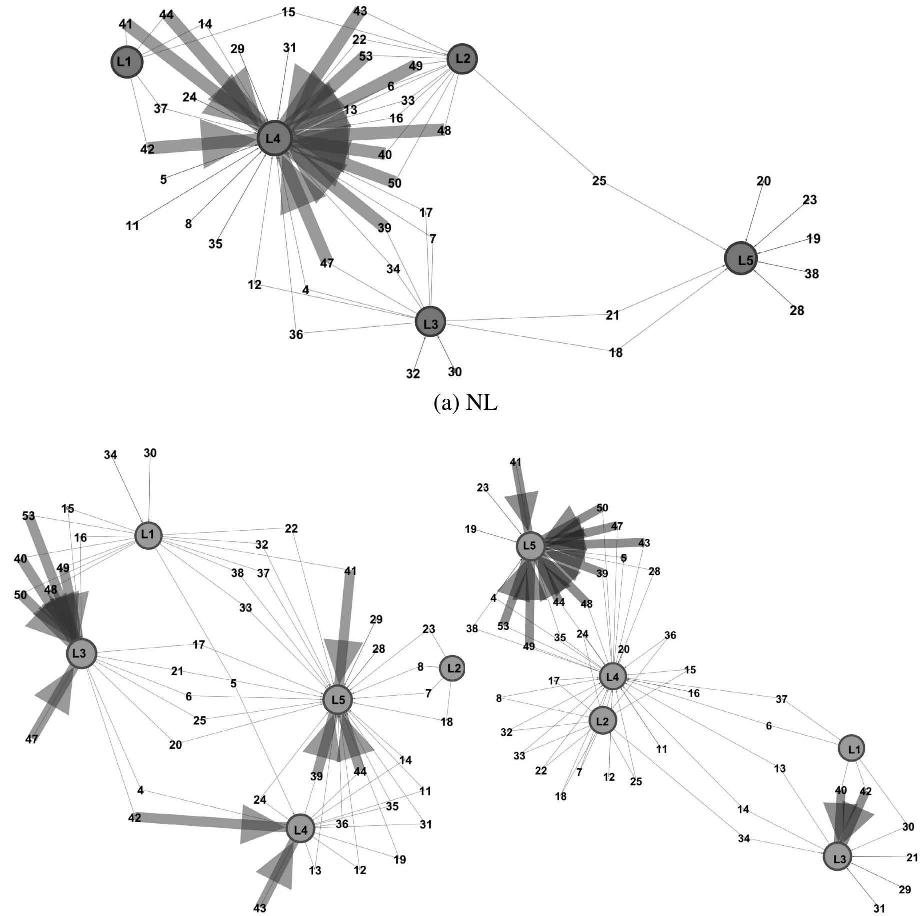
**Fig. 2.** Network interpretation of survey results for several countries

Figure 3b reveals that the processes that are important to NL (thick arrows) are distributed all over the graph of Country A.

In particular, NL is interested in improving 11 processes, only four of which are at a high level in Country A (i.e., 39, 41, 43 and 44). Moreover, five processes important for NL are currently at the lowest possible level of development in Country A, level 1 (i.e., 40, 48, 49, 50 and 53). Since some areas of interest to NL appear in the Country A network representation as static nodes (47 and 43), we learn Country A does not aspire to improve the levels of these areas and therefore it is probably not interested in forming a closer partnership to improve them.

In short, Figure 3 tells decision makers that Country A and NL are too different to make promising close partnership to improve processes in a symmetric bi-directional relationship (at best, they could build an asymmetric technology-transfer partnership).

**Country E.** Figure 3c depicts a network representation of Country E, a better partner for NL. As in Figure 3a and Figure 3b, thick arrows represent the processes that are important to NL. Most of the processes highlighted are pointing to level 5, meaning that Country E is interested in improving almost every single process NL finds important and that the current level of these specific processes are already at a high level (4).




(a) NL

(b) Country A

(c) Country E

**Fig. 3.** The current and aspired values of the essential processes in NL, Country A and Country E

Process 41 is a static node connected to L5, which means Country E already has process 41 at a high level. Thus NL (where process 41 has current level L1 and aspired level L4, as shown in Figure 3a), could profit from the experience of Country E.

Processes 40 and 42 (upgrading nodes linked to L3 in Figure 3c) show that NL and Country E could benefit from a partnership because although at a slightly higher level in NL, both countries aspire to upgrade each of these processes (the corresponding nodes in Figure 3a are upgrading nodes connected to L4).

Altogether, the situation of processes 40, 42, and 41 suggests that a symmetric bi-directional technology-transfer partnership is a possibility, where NL can benefit for process 41 and Country E can benefit for processes 40 and 42. Together with the other processes, Figure 3 shows that NL and Country E have matching current levels and aspirations for essential processes, which might be evidence of a plausible successful closer partnership between NL and Country E.

# 6 Case Study Results: Expert Perception of Visualization Insights

Applying our approach produced two visualizations. Firstly, we saw in Figure 2 a network interpretation for four countries: two countries with poor potential for closer cooperation (Figure 2a and Figure 2b), and two countries with good potential for closer cooperation (Figure 2c and Figure 2d). Secondly, we saw in Figure 3 that our representation indicates that the customs organizations of NL and Country A are very different, hindering possibilities of successful cooperation and that the customs of NL and Country E have interests that fit with one another.

These results have been discussed with customs experts who had participated in the original benchmarking study. The experts found the reasoning leading to the identification of potential cooperation partners surprisingly correct, and proposing Country E as a good potential partner for NL matched their insight.

## 6.1 Implications for Practice

According to Dutch experts, our approach is useful to find potential candidates for closer cooperation because, firstly, it supports decision makers in dealing with the *large amount of information*. The advantage of this method is that it helps decision makers to process a lot of data. "There are too many potential partners and too much information. With three partners you can keep it all in your head, but for more partners you need something like this."

Moreover, manipulation of the graph, for instance highlighting all the processes related to an area, helps decision makers to discuss the common points between two partners. They noted that depending on the area of collaboration, a partial match might be good enough.

Secondly, our method is helpful to find potential partners for closer cooperation because it is explicit, pointing out the *rationale behind the choice* of a potential partner. "A lot we do is intention based. We think it would be good to talk with [country X]. There is much politics. It would be good to base collaboration more on attributes such as the architecture possibilities."

Thirdly, experts noted that our network approach for analysis to find partners for closer cooperation has *good potential for scalability*, since our method could be extended with new attributes such as language and geographical location of the country. *Semi-automatization* makes it possible to scale up, since our approach makes it easy to include more countries and get visualizations instantly. "[The ability to effortlessly include new countries easily is required because] we have now new countries that want to participate."

## 6.2 Discussion of Our Approach

*Novelty.* Like other approaches, our approach supports decision makers in discussing opportunities for cooperation. However, our model goes beyond the theoretical plane by exploiting tools to analyze a great amount of information without loosing neither the big picture, nor the zoom-in lens. This sets our approach apart from traditional methods.

*Generalization to other networks.* Data center consolidation (different issue, different network) and IT services in a network of institutions of higher education (same issue, different network) are examples of areas mentioned in expert interviews where our approach would be useful. Thus, we cannot prove that our research generalizes to other cases, but the results of our case study are encouraging at the very least. Experts consulted were positive about this, considering our method useful to detect opportunities for cooperation in networks where partners could define their ambitions.

*Future work.* While the graphs presented in this paper are generated semi-automatically, currently in our approach the identification of suitable partners for closer cooperation includes manually assessing the (dis)similarity of these graphs. Our approach could be enhanced by applying similarity metrics to discard some graphs, such that only promising candidates are left for manual inspection.

A second line of future work is about different scenarios for collaboration. Collaborating closely is much more than copying. When planning closer cooperation, what should we take from a partner? Should we copy their source code, their design or a functional idea of data-process separation? Our current network graphs such as those presented in this paper are not about source code or systems. However, there is enough evidence to believe our method can potentially be helpful here, too. We expect that we will encounter additional dimensions apart from the two we currently address (current and aspired level and the gap between these) and that we will have to extend our visualizations accordingly, e.g. by using size and/or color of nodes and edges.

## 7 Conclusion

We conclude that a network-analysis based approach for decision support in alliance formation within existing networks is a useful tool for determining potential partners for closer cooperation in such a network. The visualizations that our semi-automatic network-analysis based approach enables reveal similarities and dissimilarities that are the basis for selecting potential partners. This is the starting point for further strategic planning by each of the potential partners. Our approach proved to be able to process large amounts of information.

**Acknowledgment.** We thank the anonymous reviewers for their valuable comments.

## References

[1] Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O.: A conceptual MAP of software process improvement. Scandinavian J. of Inf. Sys. 13, 81–101 (2001)
[2] Barabasi, A.L.: Linked. Perseus Publishing (2002)
[3] Baskerville, R.: Investigating information systems with action research. Communications of the Association for Information Systems 2, 4 (1999)
[4] Borgatti, S.: Identifying sets of key players in a social network. Comp. & Math. Org. Theory 12, 21–34 (2006)
[5] Chi, L., Holsapple, C.: Understanding computer-mediated interorganizational collaboration: a model and framework. J. of Knowl. Management 9, 53–75 (2005)

[6] Freeman, L.: Centrality in social networks conceptual clarification. Social Networks 1(3), 215–239 (1978)
[7] Fuggetta, A.: Software process: a roadmap. In: ICSE 2000: Proc. Conf. on the Future of Software Engineering, pp. 25–34. ACM, New York (2000)
[8] Goldenson, D.R., Herbsleb, J.D.: After the appraisal: a systematic survey of process improvement, its benefits, and factors that influence success. Technical CMU/SEI-95-TR-009, Softw. Eng. Institute, Carnegie Mellon Univ. (1995)
[9] Hansen, B., Rose, J., Tjørnehøj, G.: Prescription, description, reflection: the shape of the software process improvement field. Int. J. Inf. Mngt. 24(6), 457–472 (2004)
[10] He, Z.L., Wong, P.K.: Exploration vs. exploitation: An empirical test of the ambidexterity hypothesis. Organization Science 15(4), 481–494 (2004)
[11] Herbsleb, J., Zubrow, D., Siegel, J., Rozum, J.: Software process improvement: State of the payoff. American Programmer 7, 2–12 (1994)
[12] Hoffmann, W.H.: Strategies for managing a portfolio of alliances. Strategic Management Journal 28(8), 827–856 (2007)
[13] Iversen, J., Ngwenyama, O.: Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at danske data. Int. J. Inf. Mngt. 26(1), 30–43 (2006)
[14] Kautz, K.: Making sense of measurement for small organizations. IEEE Softw. 16(2), 14–20 (1999)
[15] Kumar, K., van Dissel, H.G.: Sustainable collaboration: Managing conflict and cooperation in interorganizational systems. MIS Quarterly 20(3), 279–300 (1996)
[16] Lin, Z.J., Yang, H., Demirkan, I.: The performance consequences of ambidexterity in strategic alliance formations: Empirical investigation and computational theorizing. Management Science 53(10), 1645–1658 (2007)
[17] Matsuda, N., Takeuchi, H.: Networks emerging from shifts of interest in eye-tracking records. eMinds: Int. J. Human-Computer Interaction 2(7), 3–16 (2011)
[18] Ponisio, L., van Eck, P., Riemens, L.: Planning cooperation in inter-organizational systems. In: E-Strategies for Resource Management Systems, pp. 61–85. IGI Global (2010)
[19] Ponisio, M.L., Sikkel, K., Riemens, L., van Eck, P.: Combining visualisation techniques to understand co-operation in inter-organisational systems. Journal of Systems and Information Technology 10(2), 159–179 (2008)
[20] Popadiuk, S.: Scale for classifying organizations as explorers, exploiters or ambidextrous. International Journal of Information Management 32(1), 75–87 (2012)

# The Many Facets of High-Performing Software Teams: A Capability-Based Analysis Approach

Petri Kettunen

University of Helsinki
Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
petri.kettunen@cs.helsinki.fi

**Abstract.** High-performing teams (HPT) have been investigated in many fields ranging from manufacturing to knowledge work. With software teams, however, the concept is still incompletely comprehended. Software teams in practice do not reside in isolation but in specific organizational contexts and, consequently, competitive environments. Their performance is thus relative to the particular context. The performance outcomes of the teams are in turn products of their specific capabilities (including agile), provided by the underlying software competencies. This paper proposes a high-performing software team capability analysis approach supported by provisional instrumentation. The aim of such a capability analyzator is to help software teams and organizations to identify their current capabilities and – in case of gaps – to gauge the development of necessary ones. The case exhibits with respect to agile capability demonstrate that it is able to capture team performance drivers of industrial software organizations under different contextual circumstances for further performance analysis.

**Keywords:** high-performing software organizations, agile software teams, capability development, process improvement, performance management.

## 1 Introduction

Modern high-performing software organizations rely increasingly on capable teams. High-performing teamwork has been investigated in many fields over the years. In particular, the success factors of new product development (NPD) teams are in general relatively well known [1]. However, it is not clearly understood, what high performance means for software development enterprises in total.

This paper approaches those issues by proposing a holistic capability analysis frame for (agile) team-based software enterprises. The attributes of the capabilities are evaluated by our previously developed Monitor instrument [2]. Based on that information, we produce the current capability profile of the team with the Analyzator instrument constructed here. Teams can then be gauged for the required capabilities.

The rest of this paper is organized as follows. The next Section 2 reviews software team performance in general and capability-oriented development views in particular.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 131–142, 2013.

Section 3 then presents the capability-based team performance analysis approach, followed by case examples in Section 4. Finally, Section 5 discusses the proposition with implications and pointers to further work concluding in Section 6.

## 2 Software Team Performance and Capabilities

Industrial-strength software product development is almost always done in teams, even in globally virtual set-ups [3]. Software teams do not exist in isolation in particular in larger product development enterprises [4]. In addition to the context, no two teams are in practice equal inside since teams consist of individual persons with different skills, competencies, and personalities.

In general, there is no one universal measure of software team performance. To begin with, software teams can be seen as general work teams and their performance accordingly [5]. Typically software team performance is associated with productivity [6]. However, software development teams have usually multiple enterprise stakeholders – including the team members themselves – and consequently multiple different dimensions of performance [7], [8], [9]. Prior literature has described many such possible software team performance measures [8], [10], [11], [12], [13], [14], [15], [16], [17].

Agile teams strive for developing the right things (products/services providing optimal value), and getting them released well at the right time (effective and disciplined delivery) [18], [19], [20], [21]. High performance can then be defined in terms of optimal value creation (benefits vs. costs) [19], [22], [23], [24]. In general, the performance in terms of agility can be measured with multiple different scales [25], [26], [27], [28], [29], [30], [31], [32], [33], [34].

Although it is difficult to define general-purpose performance metrics for specific software teams, the measurement systems can be developed based on existing general-purpose frameworks to begin with [35]. It is imperative to know, who judges the success and when [36], [16]. Although financial performance measures are still the most obvious ones in industrial enterprise teamwork, recently additional dimensions have been proposed – such as 'triple-bottom-line' [37].

Considering achieving the performance, the resource-based view (RBV) is a well-known approach for organizational development. Our work presented in this paper builds on those grounds in general. More specifically, we are analyzing high-performing teams in terms of their capabilities. In essence, we are in search for capable teams striving for prowess.

In general, the term capability is used in various ways in extant organizational development and management literature. Here we take the basic stance that they are qualities, abilities, and features that can be used and developed (potential). More systematically, the following definition (by replacing 'person' with 'team') is congruent: "*skills and abilities, aptitudes and attitudes needed by a person to achieve high performance in a specific role*" [38].

A closely related term is competence. Sometimes they are used interchangeably. However, in this work, we consider competencies as components and building blocks of capabilities [39]. This line of thinking meets also for instance the traits of the

Performance Prism framework [40]. Software competences have been categorized in various different ways [41]. In general, just having right competencies is not enough to make a capable team [42].

Agile software teams have certain distinct capabilities with respect to performance [43]. One of the key capabilities of agile teams is consequently the ability to perform fast development cycles with frequent customer feedback. This can be expressed in terms of capabilities [41].

Software organization capabilities are typically seen from the process-oriented viewpoint [44], [45]. In addition to such specific models, there are also many general-purpose organizational development frameworks with supporting assessments instruments [46], [47], [48], [49]. At the software team level, the Team Software Process (TSP) is one of the most well-established performance development approaches [50]. The TSP has subsequently been coupled with the CMMI-DEV model [51].

Agile software teams do by definition self-reflective continuous capability evaluation and improvement [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62]. Although agile software team development has often been seen contradictory to the organizational capability models (chiefly CMM-models) there are certain current attempts to bridge such gaps [63].

## 3 High-Performing Software Team Capability Analysis Method

The overall standpoint of our team analysis approach is as follows. For each particular software team, there is an ideal in its specific organizational context (desired state). The current state of the team may deviate from that for various reasons. The objective is then to understand the current position of the team and the performance traits to be developed and improved in order to reach the desired state (gap analysis).

The approach has been advancing with the following line of thinking: The (high) performance of the team is the result of its capabilities (bundle). The level of the capabilities can be characterized with a set of attributes. The actual realization of the capabilities may be incomplete and possible hindered by impediments.

The attributes of the capabilities are measured by the Monitor (team self-assessment). Based on that information, we produce the current capability profile of the team with the Analyzator. We can then discuss that together with the team in order to see, whether the team have sufficient and fit capabilities for the desired (high) performance, which capabilities should be improved in the future, and what potential obstacles and impediments should be removed in order to get the full benefits of the capabilities.

This work builds on our earlier investigations of sensing high-performing software teams with a self-assessment instrument [2]. In sum, the team monitoring instrument has been constructed as follows:

1. We have reviewed a large body of extant literature on software team performance investigations like exemplified in Sect. 2. A typical research method in such studies is to assess the performance impacts on selected influence factors with correlation analysis.

2. Based on such significant findings, we have incorporated a wide set of influencing performance factors.
3. The factors are then turned into characteristics questions for the survey instrument. A guiding principle for compiling the questions has been to avoid overly theoretical terminology and abstract concepts, which are often interpreted differently in different organizations and not always well known by the practitioners. The general idea is to formulate catalytic questions (probes) triggering further elaborations by the team members rather than being prescriptive.

The current realization of the Monitor instrument is a web-based questionnaire tool. It comprises three main categories of question items: performance, team(work), and organization.

We can now advance with the self-assessment Monitor to more elaborate analysis. The Monitor instrument captures a wide set of team performance attributes. By selecting and combining distinct subsets of them, we can produce capability views of the team. This is the design rationale of the Analyzator instrument proposed in this paper.

The Analyzator aggregates certain subsets of the Monitor questionnaire items and recombines them for the selected team capability indicators. Certain items are coupled to multiple capabilities. Currently the Analyzator covers the following distinct capabilities: Agile, Lean, Business Excellence, Operational Excellence, Growth, and Innovativeness.

Since there are no universally agreed definitions for those capabilities in software teams, we base on their commonly stated characteristic principles to assign the question items. For instance the constituting items (6) of the Agile capability part lean on the focal points of the Agile Manifesto thinking. For example the emphasis on working software is reflected as whether the team is capable of quick round-trip software engineering cycles, c.f., Table 1. Note that the word 'agile' is not used directly.

This work does not propose any particular quantitative formulas for determining the level of the team capabilities (such as capability index). Instead, we rely on the expert judgement of the team itself supported by visual plotting of systematized item combinations as sourced in the team self-assessment (Monitor). The suggested heuristic reasoning is as follows: If the indicator items associated with a particular capability appear to be positive, the current level of the team with respect of that capability may be high. Conversely, if there are some negative signs and/or large variations between the individual team member ratings of the items, the level of capability may be lower.

We have a tool-assisted implementation (MS-Excel) of visual plotting for evaluating the individual capabilities. For each indicative item, the distributions of the selected Monitor questions (currently 6 fixed) are shown with graphs. The tool provides such a view for each individual capability (e.g., Agile).

## 4 Case Exhibits

We have been applying the Monitor-Analyzator instrumentation with several practicing software teams in academic as well as industrial settings. The following presents such results of two different industrial teams with the detailed items of the current Analyzator realization. Following our initial development the Monitor, the Analyzator has been utilized with several investigations with various industrial software development organizations (in Finland). There are two such case teams included here. Considering their key demographic information, the team #1 develops embedded system components in a medium-size global company, whereas the team #2 is a dedicated software product development group in a small, domestic company. Table 1 presents the Monitor data of the four case teams as viewed by the Analyzator. The organization of the table is as follows: The question blocks (6) are the currently

**Table 1.** Case teams Analyzator data (Agile capability)

| **What are the key roles in your team?** | Key | Important | Relative | Some little | Little | I don't know |
|---|---|---|---|---|---|---|
| *Collaborative, representative, authorized, committed and knowledgeable customer (proxy)* | | | | | | |
| Industrial Team #1 | 0 | 2 | 1 | 3 | 1 | 0 |
| Industrial Team #2 | 1 | 0 | 2 | 0 | 0 | 1 |
| **How do you rate the following organizational factors in your context?** | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | I don't know |
| *The organization is flexible and responsive to changing customer needs.* | | | | | | |
| Industrial Team #1 | 1 | 1 | 3 | 2 | 0 | 0 |
| Industrial Team #2 | 1 | 2 | 0 | 0 | 0 | 1 |
| **How do you rate the following concerns?** | Always | Usually | Occasionally | Seldom | Never | I don't know |
| *Our team is capable of quick round-trip software engineering cycles (design-build-test-learn).* | | | | | | |
| Industrial Team #1 | 1 | 5 | 1 | 0 | 0 | 0 |
| Industrial Team #2 | 1 | 2 | 0 | 0 | 0 | 1 |
| *Our decision-making and problem-solving routines and practices are fast and efficient.* | | | | | | |
| Industrial Team #1 | 2 | 3 | 2 | 0 | 0 | 0 |
| Industrial Team #2 | 0 | 3 | 0 | 0 | 0 | 1 |
| **How do you rate the following aspects from your point of view?** | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | I don't know |
| *My team does NOT have the opportunity to use its own initiative or judgement in carrying out its work.* | | | | | | |
| Industrial Team #1 | 0 | 0 | 3 | 3 | 1 | 0 |
| Industrial Team #2 | 0 | 0 | 0 | 1 | 3 | 0 |
| **What are the modes of leadership in your team?** | Always | Usually | Occasionally | Seldom | Never | I don't know |
| *Shared leadership* | | | | | | |
| Industrial Team #1 | 0 | 4 | 3 | 0 | 0 | 0 |
| Industrial Team #2 | 0 | 0 | 3 | 1 | 0 | 0 |

incorporated indicating items of the Agile capability. They have been extracted from the Monitor. The data shows the number of responses of the team self-ratings like described in detail in our initial publication of the Monitor instrument [2].

In essence, the Analyzator views tabulated in Table 1 exhibit, how the case teams perceived certain key aspects contributing to their agile capabilities. However, they do not measure how agile the team was in its performance. That must be measured otherwise. Specific performance measures could then be some of the ones summarized in Sect. 2. As of this writing, we do not have such measurement data readily available. However, we can reflect the overall performance outcomes of the case teams with respect to agility like follows (c.f., Table 1):

- For the industrial teams, the customer collaboration (‘*collaborative, representative, authorized, committed and knowledgeable customer (proxy)*’) is especially important in the smaller product organization (#2) with intimate customer relationships whilst for the systems component team (#1) the customer appears to be more distant. Moreover, the smaller organization (#2) appreciate that they are ‘*flexible and responsive to changing customer needs*’.
- Both teams are usually capable of ‘*fast and efficient*’ decision-making and quick developmental (‘*software engineering*’) cycles. Those are typically important enablers for agility.
- In the industrial teams, the component team in the larger systems organization (#1) perceived to have less ‘*opportunity to use its own initiative or judgement in carrying out its work*’ compared to the smaller organization (#2).
- For both teams, the role of ‘*shared leadership*’ appears to be somewhat unsettled. This may lead to for instance unclear responsibilities, possibly stifling agility.

In addition, in case of the industrial team #2, the R&D manager of the software organization used the Monitor independently of the team (i.e., external view). Table 2 shows his ratings. Comparing to the team internal self-ratings (Industrial Team #2 in Table 1), we can see that they are mutually well in alignment. This suggests that the overall performance outcomes of the case team with respect to agility are as expected.

**Table 2.** Industrial Team #2 Analyzator data by the R&D manager (Agile capability, partial)

| **How do you rate the following organizational factors in your context?** | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | I don't know |
|---|---|---|---|---|---|---|
| *The organization is flexible and responsive to changing customer needs.* | | | | | | |
| Industrial Team #2 EXT | 0 | 1 | 0 | 0 | 0 | 0 |
| **How do you rate the following concerns?** | Always | Usually | Occasionally | Seldom | Never | I don't know |
| *Our team is capable of quick round-trip software engineering cycles (design-build-test-learn).* | | | | | | |
| Industrial Team #2 EXT | 0 | 1 | 0 | 0 | 0 | 0 |

**Table 2.** (*continued*)

| How do you rate the following aspects from your point of view? | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree | I don't know |
|---|---|---|---|---|---|---|
| *My team does NOT have the opportunity to use its own initiative or judgement in carrying out its work.* | | | | | | |
| Industrial Team #2 EXT | 0 | 0 | 0 | 1 | 0 | 0 |

# 5 Discussion

## 5.1 Evaluation

The Monitor-Analyzator approach proposed in this paper does not advocate any normative, analytical team model for a "dream-team" [9]. Instead, it attempts to support particular teams in practice towards their "team-dream". Currently the main limitation of the approach is that, in its current stage of development, the Monitor-Analyzator has not been validated for prediction. For instance, with respect to the Agile capability, the Analyzator view based on the team Monitor instrument self-rating information merely suggests that the team may perform high in terms of agility. However, that is not measured here. It must be measured otherwise (c.f., Sect. 2). Comparing the capability profiles provided by the Monitor-Analyzator against the actual team performances would also provide more validating evidence of longer-term performance development benefits of the proposed approach and the instrumentation.

Conversely, if the Analyzator view indicates that the team itself perceives to have some weaknesses in its current capabilities, there is a risk of lower performance. Such considerations should then be taken into account when anticipating (if not predicting) the teams' future performance. The key is that the team itself recognizes its own level of capability. The Monitor-Analyzator instrumentation approach devised here is exactly for those purposes. The case team examples presented in Sect. 4 justify those considerations. Like illustrated in there, the Analyzator views should be evaluated in conjunction with the actual context and situation of the team.

Like recognized already in our initial works with Monitor, the team self-rating survey has certain inherent limitations and constraints such as lack of common terminology, trust, honesty, self-assessment biases, and survey method limitations [2]. However, most of those limitations and risks can be mitigated by face-to-face discussions with the team members (e.g., clarifying potential misunderstandings). In reality, such reflective dialogue is anyway required to be able to engage the team and stimulate their performance improvements.

## 5.2 Implications

The Monitor-Analyzator is primarily a diagnostic instrument. It does not advocate direct solutions or particular software practices. The general design idea is to illuminate the important capability areas for specific software teams to work on. The aim is to build contextual and situation-aware understanding of the key capabilities for high-performing with the aid of probing and catalytic questions. For example, with respect to the Agile capability, if the team is currently indicating that it is not always '*capable of quick round-trip software engineering cycles (design-build-test-learn)*', they should delve into that in their particular context in order to see, how they could improve it in their practice (e.g., with more efficient software tools). By and large, our Monitor-Analyzator approach strives for addressing the following strategic issues in the organization [64]: What is (business) success for the team / organization? What are the key capabilities needed for success? What are the individual competences needed to support the capabilities? What cultural characteristics are needed to grow the people in and for the teams towards higher performance? More specifically, the following questions can then be set and scrutinized with each particular software team: Are we more interested in the current state or the ideal target state ("would-be")? Are we looking for some specific performance problems or more like an overview? Is there some particular capability that we want to focus on at this stage?

### 5.3 Future Work

We see the following prospective thread for further research and development of the approach and instrumentation:

- By conducting more case studies with different organizations, the actual expressive strength of the various Monitor items could be weighted more systematically. Some less significant ones could then possibly be removed to make it more compact. On the other hand, new important items may still emerge.
- Following that, the current configuration of the Analyzator can be evaluated further with respect to the indicating items (currently 6, but could vary) of each capability descriptor. For the Agile capability, certain frames reviewed in Sect. 2 (e.g., the Agile Positioning System [62]) could possibly be used as reference sources. In addition, potential new capability views could be considered (e.g., Flexibility, Resilience). Moreover, potential linkages between the different elementary items within the capabilities and combining capabilities (bundling) such as Agile & Lean would deserve further investigations. A related higher-level concept is dynamic capability. In essence, they are about processes which combine and modify existing capabilities (resources). However, the present work focuses on individual capabilities, and their potential interplay remains for further investigation.
- Each capability element could be systematized further by defining the related competencies [65], [66].
- The next advancement would be to construct a Navigator instrument based on the current Monitor-Analyzator constellation with the following line of thinking:

1. To begin with, the different organizational stakeholders (internal including the employees and possibly also external) of the specific software team in its particular context shall be identified. Their expectations (needs and wants) for the team performance can then be determined and prioritized.

2. The Analyzator as presented in this paper provides a profile of the software team's current capabilities.
3. The capabilities are then compared against the aligned expectations of the stakeholders. If there are noticeable gaps, the current team capabilities should be improved and possibly also some complementary ones developed in order for the team to be able to perform to the expectations.
4. Both the stakeholder expectations and the associated capability views of the Analyzator should eventually be gauged with appropriate measurements. This would make it possible to realize, how the different capabilities (possibly in combinations) really contribute to the performance.

## 6 Conclusion

In this paper, we have presented a capability-based software team performance development approach focusing on agile capabilities. The approach is supported by provisional Monitor-Analyzator instrumentation, which has been utilized in several practicing software teams both in academic and industrial cases. The key research design principle of our team Monitor-Analyzator approach has been not to limit to any one particular discipline (e.g., computer science). Instead, we take a holistic view of software teams consisting of individuals in their organizational and business contexts. The underlying reasoning is that both the success and potential inhibitors of high performance of the particular team may stem from various different origins, all of which are not necessarily known *a priori*. The intended key users for this instrumentation are practicing software teams themselves, possibly coupled with coaches. The Monitor-Analyzator facilitates forming a self-image of the teams' current performance capabilities, making it consequently possibly to understand them with respect to the ideal state of the team in its particular organizational context. Consequently, with such understanding the software organization can gauge its teams for achieving the overall (business) goals of the organization. Moreover, the organization can direct its development activities according to the capabilities of the software teams with such profound understanding of its team-based strengths in the competitive environment.

**Acknowledgements.** This work was supported by the Finnish National Technology Agency TEKES (SCABO project no. 40498/10).

## References

1. Cooper, R.G., Edgett, S.J.: Lean, Rapid, and Profitable New Product Development. BookSurge Publishing, North Charleston (2005)
2. Kettunen, P., Moilanen, S.: Sensing High-Performing Software Teams: Proposal of an Instrument for Self-monitoring. In: Wohlin, C. (ed.) XP 2012. LNBIP, vol. 111, pp. 77–92. Springer, Heidelberg (2012)

3. Kleinschmidt, E., de Brentani, U., Salomo, S.: Information Processing and Firm-Internal Environment Contingencies. Performance Impact on Global New Product Development. Creativity and Innovation Management 19(3), 200–218 (2010)
4. Kettunen, P.: Agile Software Development in Large-Scale New Product Development Organization: Team-Level Perspective. Dissertation, Helsinki University of Technology, Finland (2009)
5. Hackman, J.R.: Leading Teams: Setting the Stage for Great Performances. Harvard Business School Press, Boston (2002)
6. Petersen, K.: Measuring and predicting software productivity: A systematic map and review. Information and Software Technology 53, 317–343 (2011)
7. Chenhall, R.H., Langfield-Smith, K.: Multiple Perspectives of Performance Measures. European Management Journal 25(4), 266–282 (2007)
8. Stensrud, E., Myrtveit, I.: Identifying High Performance ERP Projects. IEEE Trans. Software Engineering 29(5), 398–416 (2003)
9. Berlin, J.M., Carlström, E.D., Sandberg, H.S.: Models of teamwork: ideal or not? A critical study of theoretical team models. Team Performance Management 18(5/6), 328–340 (2012)
10. Ginac, F.P.: Creating High Performance Software Development Teams. Prentice Hall, Upper Saddle River (2000)
11. Sawyer, S.: Effects of intra-group conflict on packaged software development team performance. Information Systems Journal 11, 155–178 (2001)
12. Narasimhaiah, G., Lam, Y.W.: Who Should Work with Whom? Building Effective Software Project Teams. Communications of the ACM 47(12), 79–82 (2004)
13. Kasunic, M.: A Data Specification for Software Project Performance Measures: Results of a Collaboration on Performance Measurement. Technical report TR-012, CMU/SEI (2008)
14. Symons, C.: Software Industry Performance: What You Measure Is What You Get. IEEE Software 27(6), 66–72 (2010)
15. Lu, Y., Xiang, C., Wang, B., Wang, X.: What affects information systems development team performance? An exploratory study from the perspective of combined socio-technical theory and coordination theory. Computers in Human Behavior 27, 811–822 (2011)
16. Sudhakar, G.P., Farooq, A., Patnaik, S.: Soft factors affecting the performance of software development teams. Team Performance Management 17(3/4), 187–205 (2011)
17. Maheshwari, M., Kumar, U., Kumar, V.: Alignment between social and technical capability in software development teams: An empirical study. Team Performance Management 18(1/2), 7–26 (2012)
18. Bozdogan, K.: A Comparative Review of Lean Thinking, Six Sigma and Related Enterprise Process Improvement Initiatives. Working paper, 060531. MIT, Cambridge (2006)
19. Winter, M., Szczepanek, T.: Projects and programmes as value creation processes: A new perspective and some practical implications. International Journal of Project Management 26, 95–103 (2008)
20. Ancona, D., Bresman, H.: X-Teams: How to Build Teams that Lead, Innovate, and Succeed. Harvard Business School Press, Boston (2007)
21. Allee, V.: Value Network Analysis and value conversion of tangible and intangible assets. Journal of Intellectual Capital 9(1), 5–24 (2008)
22. Buschmann, F.: Value-Focused System Quality. IEEE Software 27(6), 84–86 (2010)
23. Patanakul, P., Shenhar, A.: Exploring the Concept of Value Creation in Program Planning and Systems Engineering Processes. Systems Engineering 13(4), 340–352 (2009)

24. Mossman, A.: Creating value: a sufficient way to eliminate waste in lean design and lean production. Lean Construction Journal, 13–23 (2009)
25. Aoyama, M.: Web-Based Agile Software Development. IEEE Software 15(6), 56–65 (1998)
26. Cockburn, A.: Agile Software Development. Addison-Wesley/Pearson, Boston (2002)
27. Highsmith, J.: Agile Software Development Ecosystems. Addison-Wesley/Pearson, Boston (2002)
28. Anderson, D.J.: Agile Management for Software Engineering. Prentice Hall, Upper Saddle River (2004)
29. Larman, C.: Agile and Iterative Development: A Manager's Guide. Addison-Wesley/Pearson, Boston (2004)
30. Schuh, P.: Integrating Agile Development in the Real World. Charles River Media, Inc., Newton Center (2005)
31. Lyytinen, K., Rose, G.M.: Information system development agility as organizational learning. European Journal of Information Systems 15, 183–199 (2006)
32. Subramaniam, V., Hunt, A.: Practices of an Agile Developer – Working in the Real World. The Pragmatic Bookshelf, USA (2006)
33. Ambler, S.W.: Disciplined Agile Software Development: Definition, `http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm` (accessed December 27, 2012)
34. Draft Recommended Practice for the Customer-Supplier Relationship in Agile Software Projects. P1648/D5. IEEE (2007)
35. Staron, M., Meding, W., Karlsson, G.: Developing measurement systems: an industrial case study. J. Softw. Maint. Evol.: Res. Pract. 23, 89–107 (2010)
36. Agresti, W.W.: Lightweight Software Metrics: The P10 Framework. IT Pro., 12–16 (September-October 2006)
37. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, New York (2010)
38. Professional Staff Core Capability Dictionary. University of Adelaide, Australia (2010)
39. Day, G.S.: The Capabilities of Market-Driven Organizations. Journal of Marketing 58, 37–52 (1994)
40. Neely, A., Adams, C., Crowe, P.: The performance prism in practice. Measuring Business Excellence 5(2), 6–13 (2001)
41. von Hertzen, M., Laine, J., Kangasharju, S., Timonen, J., Santala, M.: Drive for Future Software Leverage: The Role, Importance, and Future Challenges of Software Competences in Finland. Review 262. Tekes, Helsinki (2009)
42. Downey, J.: Designing Job Descriptions for Software Development. In: Barry, C., et al. (eds.) Information Systems Development: Challenges in Practice, Theory, and Education, vol. 1, pp. 447–460. Springer Science+Business Media (2009)
43. Conboy, K., Fitzgerald, B.: Toward a conceptual framework for agile methods: a study of agility in different disciplines. In: Mehandjiev, N., Brereton, P. (eds.) Workshop on Interdisciplinary Software Engineering Research (WISER), pp. 37–44. ACM, New York (2004)
44. CMMI for Development. Technical report, CMU/SEI-2010-TR-033. Software Engineering Institute, Carnegie Mellon University, USA (2010)
45. Guidance on use for process improvement and process capability determination. Information technology, Process assessment, Part 4: 15504-4, ISO/IEC (2009)
46. EFQM Excellence Model. EFQM Foundation, Belgium (2012)

47. Baldrige National Quality Program: Criteria for Performance Excellence. National Institute of Standards and Technology (NIST), Gaithersburg, MD (2012)
48. Drexler, A., Sibbet, D.: Team Performance Model (TPModel). The Grove Consultants International, San Francisco (2004)
49. Miller, L.M.: Lean Team Management: How to Create Lean Management & Lean Organization. L. M. Miller Consulting (2010)
50. Humphrey, W.S.: Introduction to the Team Software Process. Addison Wesley Longman Inc., Reading (2000)
51. Humphrey, W.S., Chick, T.A., Nichols, W.R., Pomeroy-Huff, M.: Team Software Process (TSP) Body of Knowledge (BOK). Technical report, CMU/SEI-2010-TR-020. Software Engineering Institute, Carnegie Mellon University, USA (2010)
52. Boehm, B., Turner, R.: Balancing Agility and Discipline – A Guide for the Perplexed. Addison-Wesley, Boston (2004)
53. Organization Suitability Risk List. White paper, DSDM Consortium, United Kingdom (2004)
54. Lappo, P., Andrew, H.C.T.: Assessing Agility. In: Eckstein, J., Baumeister, H. (eds.) XP 2004. LNCS, vol. 3092, pp. 331–338. Springer, Heidelberg (2004)
55. Hansson, C., Dittrich, Y., Gustafsson, B., Zarnak, S.: How agile are industrial software development practices? Journal of Systems and Software 79(9), 1295–1311 (2006)
56. Pettit, R.: An "Agile Maturity Model"? Agile Journal (2006)
57. Highsmith, J., Wysocki, K.: How Agile Are Organizations Today? Cutter Agile Project Management Executive Report 7(12) (2006)
58. Pikkarainen, M.: Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices. Dissertation, University of Oulu, Finland (2008)
59. Pikkarainen, M., Mäntyniemi, A.: An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies. In: SPICE (2006)
60. Sidky, A.: A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework. Dissertation, Virginia Tech., USA (2007)
61. Kroll, P., Krebs, W.: Introducing IBM Rational Self Check for Software Teams, `http://www.ibm.com/developerworks/rational/library/edge/08/may08/kroll_krebs/`
62. Oza, N., Abrahamsson, P., Conboy, K.: Positioning Agility. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) XP 2009. LNBIP, vol. 31, pp. 206–208. Springer, Heidelberg (2009)
63. Glazer, H.: Love and Marriage: CMMI and Agile Need Each Other. CrossTalk 23(1), 29–34 (2010)
64. Wirtenberg, J., Lipsky, D., Abrams, L., Conway, M., Slepian, J.: The Future of Organization Development: Enabling Sustainable Business Performance Through People. Organization Development Journal 25(2), 11–22 (2007)
65. Turley, R., Bieman, J.: Competencies of exceptional and non-exceptional software engineers. Journal of Systems and Software 28(1), 19–38 (1995)
66. Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F.J., Tovar-Caro, E.: Competence gaps in software personnel. A multi-organizational study. Computers in Human Behavior 29(2), 456–461 (2013)

# Smart Integration of Process Improvement Reference Models Based on an Automated Comparison Approach

Simona Jeners and Horst Lichter

RWTH Aachen University, Research Group Software Construction
{simona.jeners,lichter}@swc.rwth-aachen.de

**Abstract.** A variety of reference models such as CMMI, COBIT or ITIL supports IT organizations to improve their processes. Although these process improvement reference models (IRMs) cover different domains, they also share some similarities. There are organizations that address multiple domains under the guidance of different IRMs. As IRMs overlap in some processes and have inter-dependencies, we developed an approach to integrate multiple IRMs. This enables organizations to efficiently adopt and assess multiple IRMs by automatically identifying IRM similarities and their dependencies. In this paper, we give an overview of this approach and particularly focus on its evaluation.

**Keywords:** reference models, software process improvement, comparison, meta-models, similarity.

## 1 Introduction

Software Engineering Research provides a wide variety of best practices, such as methods, techniques, tools to support IT projects to achieve their goals. However, there are still challenges and constraints that influence the projects and can lead to their failure (they either don't meet the deadlines or don't achieve the requested quality or are canceled) [1]. As the quality of the processes influences the project success, one problem that causes project failure could be that the applied processes do not reflect the mentioned best practices. There are different improvement reference models (IRMs) such as CMMI-DEV (2010), ISO/IEC 15504 (2007) or COBIT (2007) that can be considered and applied to improve the internal processes of an organization. IRMs are collections of best practices (often called procedures) based on experience and knowledge of many organizations.

The adoption and assessment of multiple IRMs bring additional benefits to organizations. The adoption allows organizations to exploit IRM synergy effects. On the one hand organizations can address different and common areas of IRMs in a coordinated way. On the other hand, the weaknesses of a single IRM can be overcome by the strengths of others. Furthermore, the assessment of the organizations internal processes according to multiple IRMs increases their competitive strength on the IT market.

The wide range of existing IRMs and their best practices, their different structure and terminology, their undefined overlappings and dependencies lead to misunderstandings and to unsuccessful application of the IRMs. An integrated view of the IRMs is needed to achieve more transparency in the collection of best practices.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 143–154, 2013.

Our primary goal is to support organizations to efficiently adopt and assess multiple IRMs. One premise for organizations to be able to exploit the synergy effects of multiple IRMs and to efficiently assess them is an integrated view of IRMs, which offers information about the similarities, coverage and dependencies between IRMs' procedures[1]. We aim to achieve our primary goal by implementing the following requirements:

- **(R1)** Identify similar procedures to benefit from synergy effects and to avoid redundancies.
  - **(R1.1)** *Similarity Degree*: For a set of procedures determine how much they have in common.
  - **(R1.2)** *Output Similarity:* Identify which procedures from multiple IRMs are related to a certain output and categorize them according to the lifecycle of the output (its creation, implementation and verification).
- **(R2)** Identify the coverage between procedures to set priorities and to avoid redundancies:
  - **(R2.1)** *Highest Coverage*: For a procedure set determine the procedure which addresses as many elements as possible from this set.
  - **(R2.2)** *Best Coverage:* For a procedure set determine the subset with the minimum number of procedures which must be adopted to cover all procedures from the set while avoiding redundancies.
  - **(R2.3)** *Adoption Degree*: For a procedure which needs to be implemented determine how much of its elements were already addressed and how many elements are left to be considered.
- **(R3)** Identify the dependencies between process areas and procedures, to set priorities in the adoption or assessment (process areas or procedures that have many outgoing dependencies must be considered first) or to manage the interfaces between the different teams in projects or in an organization (different teams in an organization can implement different IRMs).

Thus, organizations can effectively and efficiently adopt and assess multiple IRMs; the efficiency increases through an **automated comparison** approach. According to ISO/IEC 24744, different IRMs vary in format, content and level of prescription [2]. An automated comparison would not be possible without a consistent normalization of the structure and of the terminology. Our approach, MoSAIC (**Mo**del based **S**election, Application and Assessment of **I**mprovement **C**oncepts), enables a fine granular integration of multiple IRMs based on meta-models and on further guidelines for a common structure and terminology.

The remainder of this paper is organized as follows. In Section 2 we give an overview of the MoSAIC approach and of its algorithms to identify procedure similarities, coverage and dependencies. In this paper, we focus on the evaluation of these algorithms and not on their detailed presentation. For this evaluation, we build a tool that implements the requirements mentioned above (R1-R3) and supports the organizations in the adoption and assessment of multiple IRMs. In Section 3, we present this

[1] Procedures can be COBIT control objectives, control practices, CMMI specific-goals, generic-goals, -practices, sub-practices, SPICE practices, and Functional Safety objectives and requirements since we found reasonable similarities between them.

tool. We continue with an evaluation of the MoSAIC approach and close with conclusions and a summary in the last section.

## 2 The MoSAIC Integration Approach

In the following we give an overview on the MoSAIC way to integrate IRMs and compare IRMs' procedures automatically. It defines two meta-models, the *Integrated Structure Meta Model* (IS Meta-Model) and the *Integrated Concept Meta-Model* (IC Meta-Model). Both are used to integrate the structure and the terminology of different IRMs. Figure 1 depicts the purpose of both meta-models and their respective concrete models, IRM-ISMs and ICM. The different structures of IRMs are represented by different geometrical shapes while the different terminology is symbolized by different small geometrical internal shapes. For each IRM a corresponding IRM-ISM can be created (e.g. CMMI-ISM). All ISMs are instances of the IS Meta-Model. Hence, all ISMs use the same set of element types which makes them analyzable and comparable.

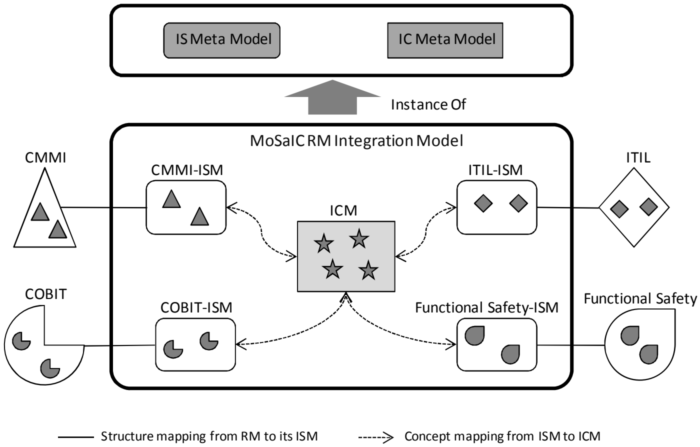



**Fig. 1.** Model-based integration approach of multiple IRMs

The **IS Meta-Model** defines elements such as categories (e.g. CMMI-DEV "*Process Management*" or SPICE "*Process Improvement Process Group*"), **processes** (CMMI "*Project planning*" or COBIT "PO10 *Manage Projects*"), **procedures** but also procedure elements, such as **activities** (CMMI-DEV "*Establish the project's budget and schedule*"), **roles** (Functional Safety "*manufacturers*"), **artifacts** (**outputs** or **inputs**) (CMMI-SVC "*service system design*") and **purposes** (CMMI-DEV "*to estimate the scope of the project*") (see details and related work in [3]). Although the IS-Meta-Model defines guidelines on a fine granular level to model the IRMs, we observed that further rules are necessary to model the IRMs consistently. According to their personality, educational and cultural background the authors of IRMs tend to express the same ideas differently. For example, while on the one hand COBIT and

Functional Safety procedures are written abundant in passive sentences, on the other hand CMMI procedures are written almost completely using the active form. Based on an analysis of the procedure writing styles, we defined further guidelines for modeling the different ISMs according to the IS Meta-Model and we offer a tool support to automatically extract the concepts and their types (activities, outputs, inputs, roles and purposes) from the text of procedures [3].

The **IC Meta-Model** defines elements, such as **procedure concepts** and their similarity relations (**generalizationOf** and **composedOf**). The notion of a procedure concept (concept for short) is essential for MoSAIC. A concept is a word or the smallest combination of words contained in a procedure that has a unique meaning in the context of IRMs (e.g. "*project plan*" or "*work breakdown structure*"). As already mentioned there can be a generalizationOf-relation ("*project plan*" is a generalizationOf *"software project plan"*) and/or a composedOf-relation between ICM concepts ("*project plan"* is composedOf "*activities*", "*roles*" or "*dependencies*"). Thus, ICM is an ontology with concepts and relations between them.

For each output, input, role and purpose in the IRM-ISM, there are one or more concepts in the Integrated Concept Model (ICM) (e.g. "*software key stakeholders*" is connected to the concepts "*software stakeholders*" and "*key stakeholders*"). These connections allow us to automatically compare them. As the output expresses the activity that produces it, we do not connect an activity with a concept in ICM and thus, we do not use the comparison of activities to achieve the requirements mentioned above (R1-R3).

As multiple IRMs contain a great number of unique concepts, modeling guidelines are needed to avoid getting a messy ontology. For this purpose, we performed a profound literature research on the maintenance of ontologies. Based on the ideas of Rector [4], [5] and Guarino et al. [6], we built an ICM that contains generalizationOf-hierarchies trees[2] to avoid complex graphs structures and, thus a "messy" ontology (an ontology that cannot be maintained any more due to the high number of relations between the elements). Furthermore, the ICM concepts in the hierarchy trees are classified to structure them and, thus find them easier (based on the ideas of facet-classification and role modeling of Schmidt et al. [7] resp. of Kensche et al. [8]). Finally, we generally[3] allow only single-point connection between the generalizationOf-hierarcy trees, i.e. their roots can be connected by composedOf-relations. This avoids the definition of a great number of composedOf-relation between the specialization concepts.

The uniqueness of the ICM concepts, their consistent identification, the similarity relations between them and their traceability back to the original concepts of the IRMs allow an automated comparison and identification of dependencies.

---

[2] GeneralizationOf-hierarchies tree contains a root concept (concept with no generalizationOf-parent) and all its specializations concepts (concepts that are connected with the root or with its specializations concepts by the generalizationOf-relation).

[3] There is an exception, i.e. when a root concept is too special to be connected to another root concept (e.g. not every "*plan*" is composedOf "*project lifecycle phases*"; the "*project plan*" is, but the "*review plan*" not).

Our approach can consider the comparison of multiple procedures from different IRMs. We compare the procedures by comparing all their combinations of **activity units**[4]. The comparison of activity units is based on the comparison of ISM concepts (outputs, inputs, roles and purposes) that is based on the comparison of all their related ICM concepts (detailed algorithms can be found in [8], [9]). We improved our algorithm, for example to allow the comparison of multiple procedures. However, the basic ideas are still valid. Summarizing, we compare elements of the same type on different levels and aggregate the results on the next level. While at the level of ICM-, ISM-concepts and activity units a value between 0 and 1 is given, on the procedure level, only some hints about the similarity of procedures are given. As the procedures of different IRMs have different numbers of activity units, this leads to low similarity degrees even though some parts of the procedure are highly similar. Therefore, when we talk about the similarity degree of procedures, we refer to the similarity degree of their activity units.

This comparison algorithm is used to identify the similarity degrees for activity units, ISM- and ICM-concepts and thus, the **procedures' similarity (R1.1)**. Furthermore, the comparison algorithm is also used to determine the **coverage of procedures (R2)**. The coverage of a procedure with respect to a set of procedures is high, if its ISM concepts (outputs, inputs, roles and purposes) have a high similarity degree to all ISM concepts of the considered procedures in the set.

The identification of **similar procedures that are related to the same output (R1.2)** and of **dependencies between procedures (R3)** is based on the similarity relations between ICM concepts (generalizationOf- and composedOf-relation). Procedures with an output which is *related* [5] to a certain ICM concept are such similar procedures (R1.2). Procedures are incoming procedures for a certain procedure *proc* (i.e. *proc* is dependent on these procedures), if they produce an output that is *related* to ICM concepts directly connected to the inputs of *proc*.

## 3 Tool Support

In this section we shortly describe how the MoSAIC web application implements the requirements R1 to R3.

To fulfill the requirements R1.1 and R2, the web application visualizes all IRMs as a tree, where each element can be expanded to see its sub-elements (an IRM with its categories, categories with its processes, and processes with its procedures) (see fig. 2). The user can select procedures or processes (all their procedures will be considered) to compute their similarity degree, i.e. **similarity degree of their activity units, ISM- and ICM-concepts (R1.1)**. He can choose between a bilateral comparison (pairs of two procedures are generated and compared) and multiple comparisons (all procedures are simultaneously compared). Furthermore, a minimum threshold for the similarity degree can also be set to filter the results.

---

[4] Activity unit contains one activity with its outputs, inputs, roles and purposes.

[5] An ISM-concept (output, input, role or purpose) is related to an ICM-concept, if the ISM concept is directly connected to the ICM concept, to all its specializations (generalizationOf-relation) or all its parts (composedOf-relation).

First, the application displays an overview of similarities between the considered procedures (the maximum similarity degree of their activity units is displayed). Finally, a more detailed view is displayed, where the user can see the similarity degree between the activity units, their outputs, inputs, roles and purposes (fig.2).

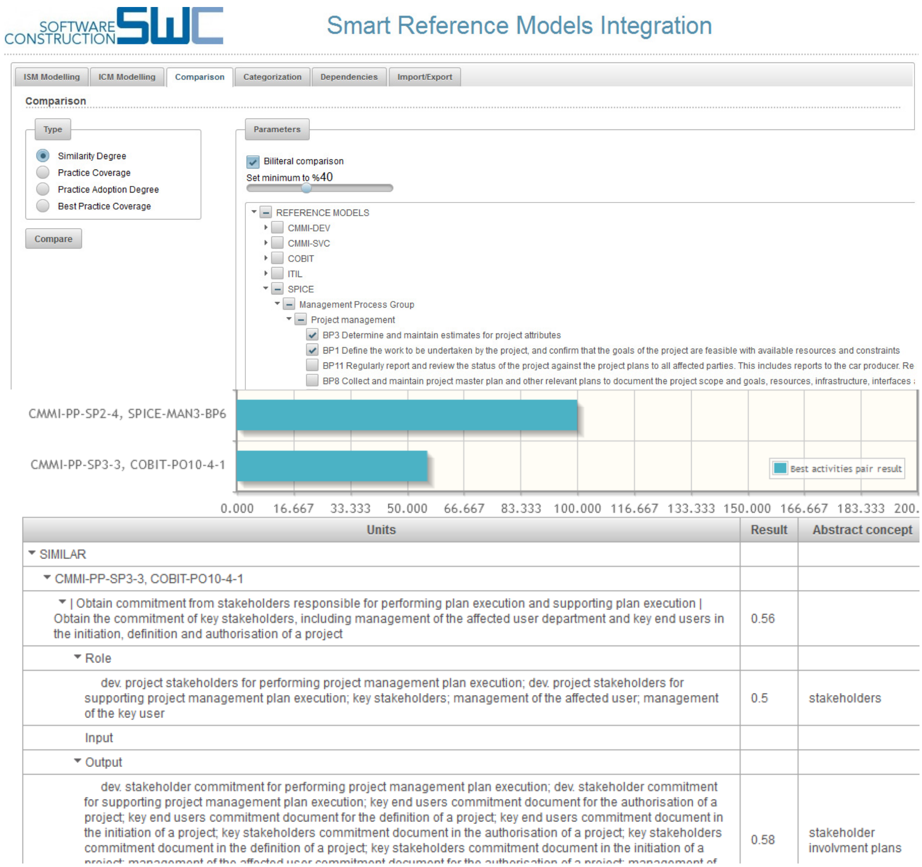



**Fig. 2.** Bilateral comparison with similarity degree results

The chart in the figure shows that the procedures PP-PP-SP-2.4 "*Plan for resources to perform the project*" and SPICE-MAN3-BP6 "*Allocate resources to activities and determine schedule for each activity and for the whole project*" contain activity units that have a 100% similarity degree. The detail table displays the comparison results of COBIT-PO10.4.1: "*Obtain the commitment and participation of key stakeholders, including management of the affected user department and key end users in the initiation, definition and authorization of a project*" and CMMI-PP-SP3.3 "*Obtain commitment from relevant stakeholders responsible for performing and supporting plan execution*". Both procedures have each only one activity unit with a 56% similarity degree. As the **context** of the outputs and roles is different (IT in general in

COBIT, development in CMMI) and COBIT is **gives more details** than CMMI ("*key stakeholder*", "*stakeholder for the initiation, definition and authorization*" in COBIT), we obtained on the ISM concept level a 58% similarity degree for the outputs and 50% for the roles. The table also shows the abstract concepts of all outputs resp. of all roles, i.e. the common ancestor concepts in the generalizationOf-hierarchy tree. This detail helps the organization to identify which are the common high-level concepts that need to be adopted to address all the concepts defined by the procedures.

As already mentioned, the user can choose to compare multiple procedures. For example, the figure 3 displays the results on the activity unit level (four combination sets with each three activity units are possible) for SPICE-MAN3-BP8 "*Collect and maintain project master plan and other relevant plans to document the project scope and goals, resources, infrastructure, interfaces and communication mechanisms*" (two activity units), CMMI-PP-SP2.7 "*Establish and maintain the overall project plan*" (two activity units) and COBIT-PO10-7 "*Establish a formal, approved integrated project plan (covering business and information systems resources) to guide project execution and control throughout the life of the project*" (one activity unit). Although all three procedures refer to the integrated project plan, the similarity degree is not 100%, because the context is different (IT and development) and SPICE and COBIT give more details about the purpose (SPICE "*to document the project scope (..)*" and COBIT "*to guide project execution (..)*").

| Units | Result |
|---|---|
| ▾ SIMILAR | |
| ▾ COBIT-PO10-7, SPICE-MAN3-BP8, CMMI-PP-SP2-7 | |
| ▸ \| Establish a formal, approved integrated project plan \| Establish the overall project plan \| collect project master plan and other relevant plans | 0.56 |
| ▸ \| Establish a formal, approved integrated project plan \| Maintain the overall project plan \| collect project master plan and other relevant plans | 0.36 |
| ▸ \| Establish a formal, approved integrated project plan \| Establish the overall project plan \| maintain project master plan and other relevant plans | 0.36 |
| ▸ \| Establish a formal, approved integrated project plan \| Maintain the overall project plan \| maintain project master plan and other relevant plans | 0.76 |

**Fig. 3.** Multiple Comparison with similarity degree results on activity unit level

Furthermore, the tool calculates the coverage of each procedure considering a procedure set to identify the **highest coverage (R2.1)**, the **best coverage** to determine the minimum subset of procedures to be adopted avoiding redundancies and the **adoption degree of a certain procedure (R2.3)**, i.e. the percentage this procedure has already been covered by the already adopted procedures.

The figure 4 displays the coverage results of the procedures SPICE-MAN3-BP1 "*Define the work to be undertaken by the project, and confirm that the goals of the project are feasible with available resources and constraints*" and CMMI-DEV-PP-SP1.1 "*Establish a top-level work breakdown structure (WBS) to estimate the scope of the project*". As the CMMI procedure does not cover the inputs (fraction result is 0/3, i.e. 0 from 3 inputs), only one output (fraction result ½, i.e. – 1 from 2 outputs) of the two procedures, its coverage is 37%. The tool also displays which are the different and common concepts. As we also considered the similarity of concepts to calculate the coverage, similar concepts can be also displayed.

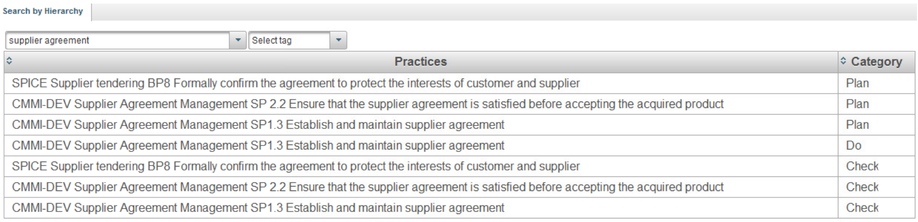
| Units | Result | Fraction Result | Different Concepts | Common Concepts | Similar Concepts |
|---|---|---|---|---|---|
| ▾ CMMI-PP-SP1-1, SPICE-MAN3-BP1 | | | | | |
| ▾ SP1.1 Establish a top-level work breakdown structure (WBS) to estimate the scope of the project | 0.37 | | | | |
| Output | 0.5 | 1.0/2 | ; dev. confirmation report; confirmation report considering feasibility between project goals, ressources and constraints | dev. project top-level WBS; | ; |
| Input | 0.0 | 0.0/3 | dev. ressources; dev. constraints; dev. goals; | | |
| Role | 0.0 | 0.0/0 | | | |
| ▾ BP1 Define the work to be undertaken by the project, and confirm that the goals of the project are feasible with available resources and constraints | 1.0 | | | | |
| Output | 1.0 | 2.0/2 | | dev. confirmation report considering feasibility between project goals, ressources and constraints; dev. project top-level WBS; | |
| Input | 1.0 | 3.0/3 | | dev. ressources; dev. constraints; dev. goals; | |
| Role | 0.0 | 0.0/0 | | | |

**Fig. 4.** Practice coverage

Considering the above procedures, the adoption degree of the SPICE activity units ("*confirm that the goals of the project are feasible with available resources and constraints*" and *"define the work to be undertaken by the project")* when CMMI procedure is already adopted **(R1.2)**, is 0% (none of the SPICE elements were already addressed) resp. 93% (almost all SPICE elements, except inputs were addressed).

The best coverage for the three procedures CMMI-DEV-SP2.5 "*Plan for knowledge and skills needed to perform the project*", SPICE-MAN3-BP5 "*Identify required skills needed for the project and allocate them to individuals and teams*" and COBIT-PO10-8-2 "*Identify required skills and time requirements for all individuals involved in the project phases in relation to defined roles. Staff the roles based on available skills information*" is the set containing the COBIT and SPICE procedure. This means that the adoption of the two procedures covers the adoption of the CMMI procedure.

**Similar procedures regarding their outputs (R1.2)** can be identified by selecting an ICM concept. The tool displays all procedures whose output is related to this ICM concept. Inspired by [10], we automatically categorize the identified procedures in the categories PLAN, DO, CHECK (Demming-Cycle-Phases). This is possible, since we tag each ICM concept by its status: if it is created (PLAN – "Establish a work breakdown structure"), if it is implemented (DO – "Perform the project plan") or if is verified (CHECK – "Analyze the software requirements"). This categorization supports an efficient adoption and assessment, as the PLAN-procedures must be adopted resp. verified before DO- and before CHECK-procedures.



Search by Hierarchy

supplier agreement | Select tag

| Practices | Category |
|---|---|
| SPICE Supplier tendering BP8 Formally confirm the agreement to protect the interests of customer and supplier | Plan |
| CMMI-DEV Supplier Agreement Management SP 2.2 Ensure that the supplier agreement is satisfied before accepting the acquired product | Plan |
| CMMI-DEV Supplier Agreement Management SP1.3 Establish and maintain supplier agreement | Plan |
| CMMI-DEV Supplier Agreement Management SP1.3 Establish and maintain supplier agreement | Do |
| SPICE Supplier tendering BP8 Formally confirm the agreement to protect the interests of customer and supplier | Check |
| CMMI-DEV Supplier Agreement Management SP 2.2 Ensure that the supplier agreement is satisfied before accepting the acquired product | Check |
| CMMI-DEV Supplier Agreement Management SP1.3 Establish and maintain supplier agreement | Check |

**Fig. 5.** Categorization of procedures considering the ICM concept "supplier agreement"

To identify the **dependencies between the procedures (R3)**, we offer a special visualization. The user can identify the dependencies between process areas of desired IRMs, all dependent procedures of one or more process areas considering desired IRMs, or the dependency tree of a single procedure. When displaying all dependent procedures of one or more process areas, the tool differentiates between the incoming and outgoing procedures by the line colors: red (more dark) for incoming resp. green (more gray) for outgoing procedures (fig. 6). For example, procedures such as SPICE-MAN-BP3 or CMMI-RD-SP3.3 define different metrics to monitor the project or to analyze the requirements that are defined as input in CMMI-DEV-MA-SP1.2, where the measurement data for these metrics is obtained.

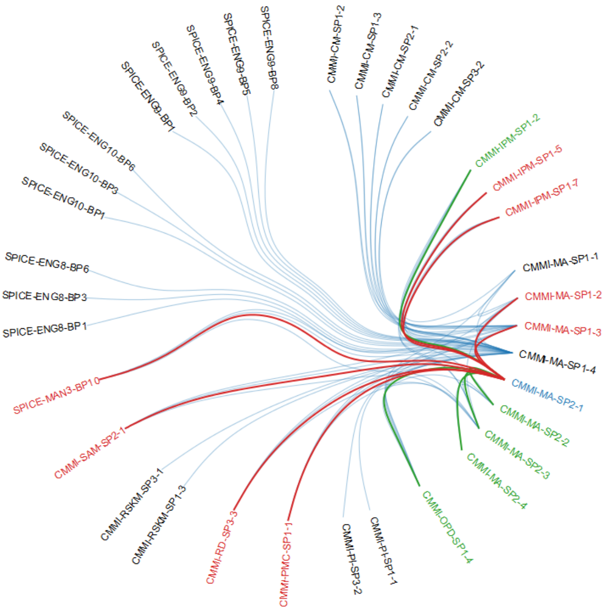



**Fig. 6.** Identification of dependencies for MA procedures considering all CMMI-DEV level 2 and level 3 processes and a subset of SPICE processes

## 4 Evaluation

In the following we present the evaluation results of the MoSAIC approach. First, we present the result of a study with our cooperation partner whether the functionality provided by our tool (requirements R1–R3) supports our primary goal: to efficiently adopt and assess multiple IRMs. Furthermore, we present the evaluation results of our algorithms (e.g. practice comparison algorithm) that implement these requirements. Finally, we give a short overview of our experiences working with the MoSAIC approach and focus on evaluation results of our algorithms that are the basis to fulfill the requirements (R1-R3).

Together with five experts from our cooperation partner (an IT department of an insurance company that started in 2005 with the process improvement and adopted during the years CMMI-DEV (maturity level 3 achieved in 2012) and ITIL; currently they are interested in the adoption of COBIT and certification according to CMMI-SVC) we evaluated the requirements relevance for their process improvement program. Furthermore, we collected their feedback to the tool usability. The experts had to evaluate the relevance of the requirements according to an ordinal scale: *low* – not relevant resp. difficult to use, *medium* – nice to have resp. moderate in use, *high* – relevant resp. easy to use. We calculated the frequencies of high, medium and low and marked the maximum frequency (Table 1). The evaluation results show that organizations have a high interest in the provided functionality, i.e. the requirements (R1 – R3) have a high relevance for process improvement and a tool would support organizations to efficiently implement them. The feedback to the usability was also good. One remark was that the results visualization can be slightly improved to better extract the needed information for the adoption or assessment (e.g. the results should be sorted according to the similarity degree or practice coverage to immediately identify the similarities or to set priorities in the adoption).

**Table 1.** Requirements and Tool Evaluation

| | Functionality | | | | | | Tool Usability | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Experts Evaluation | | | Frequencies | | | Experts Evaluation | | | Frequencies | | |
| | high | medium | low | high | medium | low | high | medium | low | high | medium | low |
| Similarity Degree Bilateral Comparison | 4 | 1 | | 80,0% | 20,0% | 0,0% | 2 | 3 | | 40,0% | 60,0% | 0,0% |
| Similarity Degree Multiple comparison | 3 | 2 | | 60,0% | 40,0% | 0,0% | 3 | 2 | | 60,0% | 40,0% | 0,0% |
| Practice Coverage | 4 | 1 | | 80,0% | 20,0% | 0,0% | 2 | 3 | | 40,0% | 60,0% | 0,0% |
| Practice Adoption Degree | 5 | | | 100,0% | 0,0% | 0,0% | | 5 | | 0,0% | 100,0% | 0,0% |
| Best Coverage | 3 | 2 | | 60,0% | 40,0% | 0,0% | 2 | 3 | | 40,0% | 60,0% | 0,0% |
| Catageorization | 4 | 1 | | 80,0% | 20,0% | 0,0% | 4 | 1 | | 80,0% | 20,0% | 0,0% |
| Dependencies | 4 | 1 | | 80,0% | 20,0% | 0,0% | 3 | 2 | | 60,0% | 40,0% | 0,0% |

The experts mentioned further requirements that can support their process improvement:

- Identify all procedures from desired IRMs (ITIL, CMMI-DEV or COBIT) to achieve a certain maturity level (e.g. level 4 or 5 of CMM-DEV) and address as much as possible from the desired IRMs.
- Restrict the access on certain IRMs, as some IRMs should not be public (internal processes or IRMs, such as Functional Safety, that needs a license).

- Identify the similarity degree of procedures without considering their context (software development, hardware development, whole IT or services). This helps to define generic processes that are available for different departments in an organization. Furthermore, assessments according to these generic requirements can be performed to get first impressions on the implementation. If needed, the detailed can be further assessed. This increase the efficiency of the assessment process.

In the following, we give an overview of our experiences with the MoSAIC approach and the tool. Based on all the guidelines of the MoSAIC approach (meta-models or consistency rules), we modeled big parts of multiple IRMs (all entire 18 processes from CMMI-DEV maturity level 2 and 3, 10 processes from SPICE, 8 processes from COBIT, parts from ITIL and CMMI-SVC). We have extracted over 1000 unique concepts that we have connected to the inputs, outputs, roles and purposes of the IRMs procedures. With the support of the parser tool (c.f. [3]) and the possibility to model IRMs in XML format and then import them, we were able to quickly model the IRMs in our tool. Due to the hierarchy trees with their most abstract concept at the top and to the categorization of the sub-concepts in the ICM, it was easy to insert new concepts, to find and assign them to the procedure elements. Furthermore, the tool supported us in the evaluation of our algorithms.

Based on the experts' results from the first evaluation[6] [8] on the identification of similar procedures and by calculating their similarity degree **(R1.1)**, we improved our algorithms. We also involved a further expert (member of the ISCN (International Software Consulting Group) group) in the evaluation by comparing his subjective rating with our results for practices from CMMI and SPICE. Totally, we evaluated the similarity scores of 151 procedure pairs from CMMI, COBIT and SPICE and compared our results with the experts' results (for this comparison we mapped our results to five categories: [1,1] as *identical;* [0.67, 1) as *high;* [0.3, 0.67) as *medium;* (0, 0.3) as *low;* [0,0] as *different*). We obtained good results by comparing the similarity metric results on the activity unit level with the experts' judgments: **0.25** and **0.26 deviation** for **CMMI-COBIT** resp. for **CMMI-SPICE** (on average less than every second metric result deviates by more than one point from the given category). As the determination of the coverage **(R2)** is based on the similarity algorithm, we did not perform an explicit evaluation.

To evaluate the dependencies **(R3)**, we verified dependencies between procedures within the CMMI process areas REQM, MA, CM, PPQA based on the identified dependencies in [11]. Our approach identified the same dependencies graphs. As our approach to categorize procedures according to their output **(R1.2)** is also based on the relations between ICM concepts, we did not perform an explicit evaluation. The correctness of the ICM is actually a prerequisite for all requirements. In the future, we tend to perform a broader quality review on ICM. However, the guidelines to model the ICM ontology and the connections of the ICM concepts to the different IRMs support us well to achieve a correct ICM.

[6] A consultant with over 20 years experience in COBIT, ITIL and CMMI; one consultant with over 15 years experience in CMMI and 5 years in SPICE; An employer from our cooperation partner.

## 5 Future Work and Conclusions

In this paper we presented an approach to integrate multiple IRMs based on IRM concepts to support an efficient adoption and assessment of IRMs. Based on two meta-models (Integrated Structure Meta-Model and Integrated Concept Meta-Model) and on the IRMs' knowledge, we created an ontology that contains all their concepts and the similarity relations between them. A tool uses this ontology and its connections to the IRMs, automatically identifies similar procedures, computes their coverage and identifies their dependencies. This ontology can be used for further purposes, e.g. to better understand the IRMs or the IT context. As the quality of this ontology is a prerequisite for the approach, we want to review bigger parts of ICM and increase its quality. In our future work, we want to extend our approach and support organizations and projects in selecting IRMs procedures that are best suited for them and bring high benefit.

## References

1. The Standish Group International: CHAOS Summary 2009. Chaos, pp. 1–4 (2009), http://www.statelibrary.state.pa.us/portal/server.pt/document/690719/ (retrieved)
2. ISO/IEC TR 24774: Systems and software engineering – Life Cycle Management – Guidelines for process descriptions (2010)
3. Jeners, S., Lichter, H., Dragomir, A.: Towards an Integration of Multiple Process Improvement Reference Models based on Automated Concept Extraction. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 205–216. Springer, Heidelberg (2012)
4. Rector, A.L., Wroe, C., Rogers, J., Roberts, A.: Untangling taxonomies and relationships: personal and practical problems in loosely coupled development of large ontologies. In: Proceedings of the K-CAP 2001, pp. 139–146. ACM (2001)
5. Guarino, N., Welty, C.: Ontology Analysis of Taxonomic Relationships. Data and Knowledge Engineering Journal 39, 51–74 (2000)
6. Schmidt, M., Polowinski, J., Johannes, J., Fernández, M.A.: An integrated facet-based library for arbitrary software components. In: Kühne, T., Selic, B., Gervais, M.-P., Terrier, F. (eds.) ECMFA 2010. LNCS, vol. 6138, pp. 261–276. Springer, Heidelberg (2010)
7. Kensche, D., Quix, C., Chatti, M., Jarke, M.: GeRoMe: a generic role based metamodel for model management. In: Proceedings of the 4th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), Agia Napa, Cyprus (2005)
8. Jeners, S., Lichter, H., Pyatkova, E.: Metric based Comparison of Reference Models based on Similarity. International Journal of Digital Content Technology and its Applications (JDCTA) 6(21) (November 2012)
9. Jeners, S., Lichter, H.: Automated Comparison of Process Improvement Reference Models based on Similarity Metrics. In: Proceedings of APSEC 2012, Hong Kong (2012)
10. Chen, X., Staples, M.: Using Practice Outcome Areas to Understand Perceived Value of CMMI Specific Practices for SMEs. In: Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.) EuroSPI 2007. LNCS, vol. 4764, pp. 59–70. Springer, Heidelberg (2007)
11. Chen, X., Staples, M., Banermann, P.: Analysis of Dependencies between Specific Practices in CMMI Maturity Level 2. In: O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.) EuroSPI 2008. CCIS, vol. 16, pp. 94–105. Springer, Heidelberg (2008)

# Tailoring the Software Product Management Framework for Use in a Healthcare Organization: Case Study

Samuel A. Fricker, Marie Persson, and Madelene Larsson

Blekinge Institute of Technology, 371 79 Karlskrona, Sweden
{samuel.fricker,marie.persson,madelene.larsson}@bth.se

**Abstract.** Many reference models were developed for software process improvement. Each model, however, is an idealized prescription that is applicable in a limited set of situation only. This paper has investigated how an existing reference model can be tailored to a domain it has not been designed for initially. The tailoring approach is based on translating the reference model to the new domain and on inductive interviews for evaluating the translated model. The approach has been applied for assessing and improving strategic requirements engineering practice in a healthcare organization with a framework for software product management.

**Keywords:** reference model tailoring, inductive process improvement.

## 1 Introduction

A plethora of reference models have been developed for improving processes and capabilities. Models such as the CMMI [1] and ITIL [2] prescribe broad sets of best practices and have been successfully used for all-over-the board process improvement in software organizations. Lightweight models such as the software product management (SPM) framework [3] and the improvement framework for lightweight assessment and improvement planning iFLAP [4] focus on specialized processes and roles and were successfully used in practice for focused and cost-efficient process improvements.

Any model, however, abstracts and represents only a fraction of the phenomena that can be observed in reality, usually those that were perceived relevant for the creation of the specific model [5]. As a consequence, reference models and the processes and capabilities they encourage represent idealized guidelines for selected domains. Many situations are inconsistent with the assumptions behind these ideals, however. For example, the focus of CMMI on software development, rather than software use for service provision gave rise to the creation ITIL. Similarly, the strategic requirements engineering activities needed by product managers for planning software products gave rise to the SPM framework because CMMI was not specific enough to support these concerns.

Process improvement situations that are insufficiently supported by existing frameworks encourage researchers and experts to create new reference models. This

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 155–166, 2013.

increases the number of standards that need to be known, selected, and applied in practice. Reference model tailoring represents an alternative to the creation of new models. Tailoring involves the interpretation and translation of an existing framework to a new context, which has been insufficiently supported previously. The resulting increased applicability of the existing frameworks limits the growth of the number of frameworks and improves the understanding of their validity.

This paper describes a case of tailoring an existing reference model to a new domain. It describes a simple two-step approach for translating the reference model and for evaluating the translation. As a result, existing process improvement knowledge can be transferred instead of being reinvented. The results contribute to a consolidation of software process improvement frameworks and enable the use of new domains to validate process knowledge.

The remainder of the paper is structured as follows. Section 2 describes related work. Section 3 describes the research methodology. Section 4 describes the translation results and section 5 the evaluation results of the reference model tailoring approach. Section 6 discusses the results. Section 7 concludes.

## 2 Related Work

Process knowledge is applied to so many different situations that no single model is able to capture all variability. Software process tailoring has been coined as a term to describe the adaptation of "off-the-shelf" software processes to meet the needs of a specific organization [6]. To enable such tailoring, situational factors have found their way into process improvement frameworks to account for an organization's specific process improvement ambitions and for domain specialties [7, 8]. Companies can choose their desired level of maturity and omit practices and capabilities that they perceive excessive [1].

Variability exists also within an organization. Projects and organizational units are required to tailor idealized processes to make them practicable, efficient, and effective [9, 10]. Tailoring strategies include dropping, downsizing, adding, expanding, and refinement actions applied on resources, communication, decision-making, documentation, knowledge, and technology. Analysis of the gap between the planned process model and the process enactment allows steering and managing process tailoring and improvement [11]. Enactment of tailored processes results in real-world experimentation with results that enable learning in the organization [12].

For situations, where no process knowledge is available, inductive process improvement approaches have been proposed [13]. In a bottom-up fashion, critical issues are identified and solutions sought for addressing these issues [14]. When based on appropriate sampling of projects, roles, and practitioners the organization's knowledge can be externalized and effective improvement results obtained [4]. The results of inductive improvements capture process knowledge that can be made available to the software industry, for example by building new or updated frameworks.

Many situations with no process knowledge available are still so similar to domains with existing reference models so that inductive creation of a new reference model is ineffective. The organizational learning process would require too much effort and the results would be applicable to the concerned organization only. In these situations, more effective is the tailoring of an existing reference model and transfer the process knowledge it captures. Such tailoring provides the additional opportunity of understanding how domains relate to each other and of extending the validation of existing process knowledge.

## 3 Research Method

Our work aimed at understanding how to transfer an existing reference model from a known assessment domain to a new such domain while being confident that the weak points of the assessed processes are found and the most valuable changes identified. The here presented case study [15] was part of an improvement initiative in a Swedish health-care organization that uses IT solutions and embedded systems such as medical devices that it procured in a regulated market [16]. The effort aimed at improving strategic requirements engineering in the organization.

Due to the similarities of software product management [17] with the strategic requirements engineering needs in the healthcare organization, we selected the SPM framework as a basis for process assessment and improvement. In a two-step process, we tailored the reference model to the healthcare domain and evaluated the tailored version with inductive questions that we integrated into the practitioner interviews. The interviews were analyzed with content analysis [18] to identify correspondences and misalignments of the assessment framework. The results are two-fold. On a method engineering level, they show how to translate and evaluate an existing assessment framework into a new, initially unforeseen domain. On a method application level, they show how to assess strategic requirements engineering of a healthcare organization with the software product management framework.

To evaluate the fitness of the tailored SPM framework for strategic requirements engineering in a healthcare organization (SRE@HC) we posed the following initial research question. RQ1: *What are the correspondences between the SPM framework and SRE@HC?* The identified correspondences were used to build the SRE@HC framework that we evaluated with the following research question. RQ2: *What is the congruence of the SRE@HC framework with the SRE@HC domain?*

The research was performed in collaboration with one of the county councils in Sweden. It served a population of 150’000 people with one hospital and multiple primary care centers. The hospital was divided according to medical specialties and services, including orthopedics, pediatrics, radiology, and operating room departments. The county council was supported by an organization that included IT, procurement, and estate departments. The support organization ensures compliance with regulations such as WTO GPA. On top of the administration, a political organization took overall responsibility for healthcare delivery. Fig. 2 (right-hand side) gives an overview of the county council and its constituents. The county council is representative for other public-sector healthcare organizations, except that it does not include medical research departments that can be found at university hospitals.

The research was performed as a two-step process. Step 1 answered RQ1 by tailoring the SPM framework into the SRE@HC framework. Step 2 answered RQ2 by evaluating the application of the SRE@HC framework in the healthcare organization. Fig. 1 gives an overview.

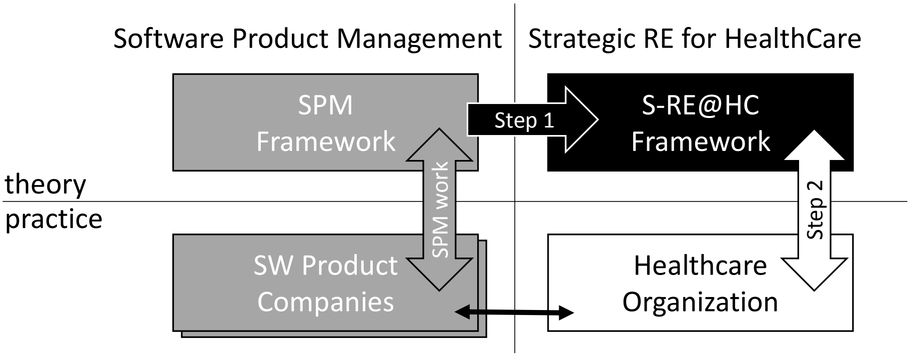



**Fig. 1.** Research process (grey: previous work, black: step 1, white: step 2)

Step 1: RQ1 was answered by first identifying correspondences between the SPM framework and strategic requirements engineering in the healthcare organization and then validating the resulting assessment instrument with an expert responsible for strategic requirements engineering in our partner organization. The requirements engineering and healthcare experience of the authors enabled the first step. Correspondences were identified for organizational roles, activities, and artifacts. As a result of this mapping, a tailored questionnaire for SRE@HC assessment was created, which was reviewed internally in the research team as an offline evaluation that did not involve any outside experts [19]. The expert from the county council performed a practitioner evaluation by reviewing the questionnaire.

Step 2: RQ2 was answered by evaluating the SRE@HC instrument in real process improvement. 14 interviews were performed that lasted approximately one hour each. Questions about compliance with SRE@HC practices were used to identify the maturity of the organization from the perspectives of the interviewees. Questions about the rationales for compliance and non-compliance with SRE@HC practices and open-ended questions about total improvement potential were used to collect evidence about congruence of the SRE@HC framework with the SRE@HC domain. This evidence was analyzed by directed content analysis to identify agreements, disagreements, and omissions of the SRE@HC framework with respect to the SRE@HC domain.

Flexible research is confronted with the following threats to validity: reactivity, respondent bias, researcher bias, reliability, and generalizability [20].

*Reactivity* refers to the way in which the researcher's presence alters the behavior of the subjects involved in the research. One of the researchers had established trusted relationships with many of the interviewed practitioners in the healthcare organization during multiple years that preceded this research. The trusting relationship and the inside-out knowledge of the organization reduced the likelihood of receiving biased information. The researchers without personal relationship assured neutrality of the research.

*Respondent bias* refers to the risks of obtaining answers that respondents judge are those the researchers want and of having information withheld that can be used against the respondents. This threat was the most critical threat in the presented research as the respondent's organizational units and activities were assessed. Risk-limiting was that all respondents perceived strategic requirements engineering to be a key area to improve and that they benefitted from the improvements launched on the basis of the obtained results. In addition, the results were triangulated between the individually interviewed subjects and member checking used with the responsible for strategic requirements engineering at the organization.

*Researcher bias* refers to the preconceptions and assumptions the researchers into a study. We used observer triangulation and an audit trail to address this threat to validity. Each interview was performed by two researchers. Also the analysis of the interview results was performed jointly. A record of the data collection and data analysis activities was kept during the study.

*Reliability* refers to how carefully the research was performed and how honestly the results were presented. Reliability was achieved by following the above-described research design, and by verifying the results with member checking. In addition, a chain of evidence was maintained.

*Generalizability* refers to how far the obtained results are applicable and valid both within the studied setting and beyond. For internal generalizability, we used a combination of purposive and stratified sampling. Interview partners were selected to cover the organizational units and roles needed for the assessment as well as possible. The responsible for strategic requirements engineering at the partner organization acted as a gate-keeper in the interview partner selection. All selected interview partners participated in the study. Concerning external generalizability, we used convenience sampling in the selection of the partnering county council. This decision reduced the reactivity threats of the study, but implied that the results are only generalizable to those parts of a healthcare organization that exclude research hospitals.

## 4 Step 1: Translation of the SPM Framework

The scope and structure of the SPM reference model are well described by its design decisions and creation process [3, 21, 22]. Product management was thought to interact with external stakeholders such as customers and supplying partners and with internal stakeholders such as a company board and various company functions. Professional software product management was interpreted as a matter of well-organized information collection, analysis, and decision-making about to the development and release of a portfolio, products, releases, and requirements for the market.

Strategic requirements engineering in the studied healthcare organization was strikingly similar. The hospital and primary care centers interacted with patients as customers and the support organization as a supplying partner. The support organization did the same by considering the hospital and primary care centers as customers and by facilitating procurements from external suppliers. Decisions were made about the organization's portfolio, about services to be developed and assets to be procured, about projects to be performed for evolving services and assets, and about needs for investments.

Despite the similarities, many differences between software product management and strategic requirements engineering in the healthcare organization existed. They concerned the organization of supplies and services, the assignment of activities to roles, the approach to decision-making, and the concepts and terms used to refer to the entities that are managed. In comparison to a software product company, the healthcare organization was not only embedded in an external supply chain, but had in addition an internal supply between the healthcare core units (a hospital and primary care centers) and the support units (IT, medical technologies, and procurement among others). Fig. 2 gives an overview of the two types of organizations. To serve patients, each service delivering unit managed its own portfolio of services. The service units delivered equipment to the healthcare units. The units had shared decision-making for investments that were needed for maintaining and enhancing the service and equipment portfolios.

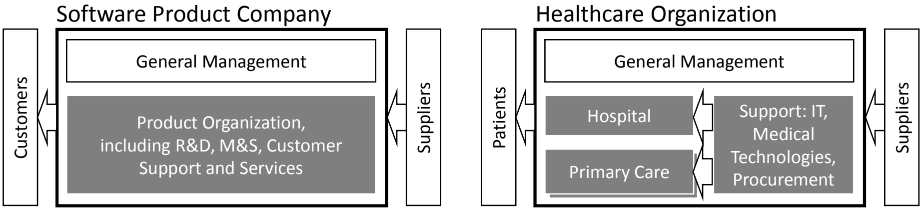



**Fig. 2.** Structures of a software product organization (left) and of a healthcare organization (right). The hospital and primary care centers represent the healthcare core business.

In the healthcare organization the SPM product management role was split and distributed over multiple roles. The medical director was responsible for the portfolio of services offered to patients. He delegated this responsibility to heads of department and heads of ward for each specialty in the hospital, such as orthopedics, and to the heads of the primary care centers. Each such head was then also responsible for managing the lifecycle of the equipment required to deliver healthcare services. Project leadership, for example for business process improvement projects, did not exist on the healthcare side and were delegated to the support units. Needs for investment were collected and specified by deputy managers.

The support organization was responsible for the assets needed to perform the healthcare services. The head of IT was responsible for the software solutions, and the head of medical technologies for equipment such as operations robots and radiology labs. For managing the portfolio and the investments they collaborated closely with the chief financial officers. Responsible for specific assets and investments for improving or replacing them were the IT architect, the head of support services for healthcare equipment, and the head of procurements. Project managers performed procurement and system integration projects and managed the requirements.

The organizational differences implied that the SPM roles needed to be translated into the healthcare context. The translations affected the wording used in the SRE@HC assessment questionnaires and the roles that were interviewed. Table 1 gives an overview of these translations that accounted for the differences observed above. 4 of total 13 roles could be transferred without adaptation.

**Table 1.** Translation of SPM roles to SRE@HC healthcare and support roles

| SPM | Healthcare | Support |
|---|---|---|
| Market | Population | HC core units |
| Customer | Patient | HC core unit |
| Prospect | - | - |
| Partner | Other healthcare organizations | Supplier |
| Partner network | Association of county councils | Domain specific groups of interests |
| Competitor | Alternative | - |
| Company board | Investment council | Investment council |
| Development | Healthcare services and support units | Suppliers |
| Market research party | Regulator | Regulator |

The differences between the software product, healthcare, and support domains implied also that SPM concepts needed to be translated. The translations again affected the wording necessary to make the SRE@HC assessment questionnaires understood by the interviewees. Table 2 gives an overview. 12 of total 23 concepts could be transferred without adaptation.

**Table 2.** Translation of SPM concepts to SRE@HC healthcare and support concepts

| SPM | Healthcare | Support |
|---|---|---|
| Product line | Assets of same supplier | Assets of same supplier |
| Product | Service (type of operation, treatment, etc.) | Asset, Equipment, Solution, Investment |
| Component | Asset, Equipment, Solution, Investment | Parts |
| Roadmap | Plan | Plan |
| Release | Project results (new service) | Project results |
| Release definition | Specification | Specification |
| Requirements | Needs (pre-project), requirements | Needs (pre-project), requirements |
| Partnering & contracting | Provision of access to services | Provision of access to assets |
| SLA | Quality guarantees | SLA |
| Pricing model | Pricing model (only dentistry) | - |
| Revenue | Revenue (only dentistry), Reinvestment | Investment |
| Engineering capacity | Budget | Budget |

The translated SRE@HC framework for assessing a healthcare organization was structured, packaged, and used alike the software product management framework.

## 5 Step 2: Evaluation of the SRE@HC Framework

Fig. 3 shows the maturity profile of the county council that was assessed with the translated framework. Each block represents a focus area with capabilities that are ordered from left to right according to increasing maturity. The portfolio focus areas were *population/HC core unit analysis*, *lifecycle management*, and *access provision* from left to right. The service and asset focus areas were *intelligence*, *service/asset planning*, and *equipment/component planning*. The project focus areas were *requirements prioritization*, *specification*, *specification validation*, *change management*, *project result validation*, and *launch*. The need focus areas were *need gathering*, *requirements identification*, and *requirements organizing*.

| | Healthcare | Support |
|---|---|---|
| Portfolio | Medical Director<br>Head of Hospital Dept.<br>Head of Prim. Care Center | Head Medical Technology<br>Head IT<br>CFO<br>CFO Dentistry |
| Service / Asset | Head of Ward | Head Investments<br>Head Med. Tech. Services<br>IT Architect<br>Head Procurement |
| Project | - | Project Mgr. |
| Requirements | Deputy Head of Ward | Project Mgr. |

**Fig. 3.** SRE@HC maturity profile of the county council (green/light colored: capability implemented, red/dark colored: capability not implemented)

The assessment of the capabilities was based on the translated roles and concepts. The capabilities suggested by the translated reference model were understandable for the interviewees, with the following exceptions.

Portfolio: The interviewees stated that they are in a controlled market that does not allow competition to the county council within the county. They acknowledged, however, that private organizations started to provide primary care and that county councils compete across the counties. All units except parts of dentistry could not specify pricing for their services. Their compensation was determined by a re-investment formula. The healthcare organization tried to achieve synergies across services and assets, but not with product lines. Instead they were interested of using product families from the same supplier. The first two exceptions were due to regulations of the healthcare sector. The third exception was due to the use instead of development of assets.

Service / Asset: The respondents stated that the make or buy decision was always a buy decision. The healthcare service units obtained assets from the support units, the support units procured them from external suppliers. The delegation to the service units was the key rationale for the organizational split of service and service units. The external procurement was due to political regulations.

Project: The healthcare service units performed only operations and delegated the projects to the support units. Revenue consideration was again not as important because of the re-investment funding approach. Instead, cost savings were important. This exception was due to the culture of the organization.

Requirements: All capabilities were understandable.

The answers to the inductive questions about improvement potential partially overlapped with the reference model. Fig. 4 gives an overview of the elicited challenges together with causes and consequences.

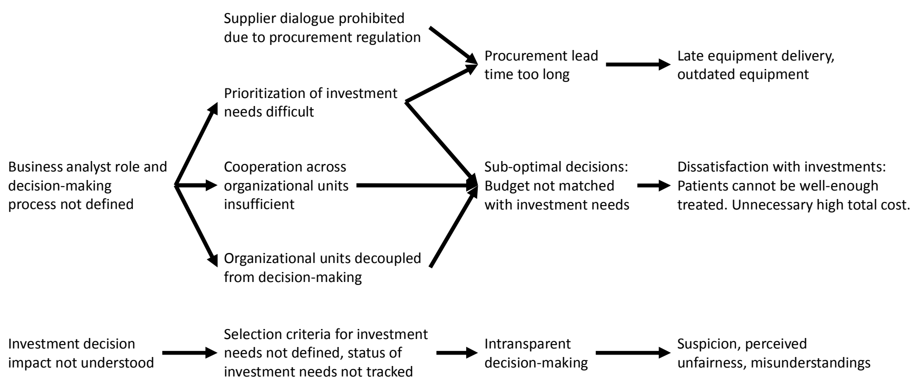



**Fig. 4.** Improvement potential based on induction: challenges, causes, and consequences

Congruent were the following findings. The organization had not defined any business analyst role. The observed lack of this role is congruent with the reference model that expects stakeholder needs to be collected and transformed into well-communicated and managed requirements. Organizational units were decoupled from decision-making. Such observed lack of integration integration is congruent with the reference model that expects stakeholder consultation. The organization requested transparency of decision-making. Again such transparency was also foreseen in the reference model, as part of requirement lifecycle management, prioritization methodology, and communication of plans.

The reference model was missing several improvements that were perceived critical by the interviewees. The interviewees requested a decision-making process that defines how the many parties should collaborate. The SPM activities that were assumed to be coordinated by a single product manager had to be translated into a concerted collaboration of managers for strategic requirements engineering and investment decision-making. Also, the interviewees requested impact evaluation of the investment decisions and the consequent project results. The reference model only requested functional validation and certification. Finally, the organization looked for ways of improving prioritization of investment needs. The reference model foresees such prioritization only for services and assets, but not in sufficient depth for the whole portfolio. Such portfolio decisions would have been important for matching budget with investment needs.

A problem area that was completely excluded by the reference model was the difficulty of regulated procurements. The reference model did not suggest any practice for how requirements should be specified for such procurement and for how to reduce lead time.

Some of the expected capabilities of the reference model were not adequate. The reference model had too high expectations on the handling of intellectual property. None of the interviewees perceived such a practice to be critical. The reference model recommended collaboration with supplying partners. Such collaboration is prohibited by procurement regulations for fairness reasons.

The assessment based on the SRE@HC framework and the inductive improvement potential questions was effective. The healthcare organization perceived the assessment results to be credible and initiated improvement actions. Positions for business analysts were created and a first position already publicly announced. Organization-wide process definition was launched to improve collaboration across organizational units for investment decision-making. A tooling project was launched for tracking needs and requirements, for increasing transparency of decision-making, and for identifying bottlenecks that lead to long procurement lead times.

## 6 Discussion

The presented case has shown that it is possible to transfer existing reference models to a domain that was not foreseen by the authors of the original model initially. The tailoring is a kind of situational adaptation [8] for transferring knowledge from one domain to another: here from software product management to strategic requirements engineering in healthcare. Understanding of how to transfer reference models extends the ability of process improvement professionals to take advantage of existing knowledge. It discourages the reinvention of yet another reference model each time a new process improvement problem is encountered and encourages consolidation of existing models instead.

The case showed that the tailoring can be performed with two simple steps: translation and evaluation of the reference model. Translation was needed to adapt the reference model to the changed organizational structure, and the roles and concepts of the new domain. Expertise in the target domains, here requirements engineering and healthcare, enabled such translation. Comparison of framework-based assessment results with inductive questions about improvement potential enabled the evaluation of the translated framework. Capability requirements that were congruent with capability needs confirmed adequacy of the reference model. Concepts that are difficult to understand, unnecessary capability requirements, and missed improvement needs indicated needs for further tailoring of the translated reference model.

The case represents a single transfer of a reference model into a new domain. Replication of such work is needed to better understand how existing results can be effectively diffused, how evaluation results can be integrated into original models, and when the benefits outweigh the cost in comparison to invention of a new reference model.

## 7 Summary and Conclusions

Many process improvement domains benefit from knowledge embedded in reference models that were designed for other domains. The paper has presented a two-step process for tailoring an existing reference model to such a new domain. The process is based on translating organizational structure, roles, and concepts and on inductive validation of the prescriptions that the reference model contains.

The process has been applied for transferring best practice from software product management to strategic requirements engineering in a healthcare organization. Application of the process in the case study showed feasibility and effectiveness of the tailoring. The evaluation results showed that important process assessment needs were congruent with the structure and scope of the initial model and that the missed improvement issues could be captured with lightweight inductive questioning.

The results are a rich, empirically grounded basis for improving strategic requirements engineering also in other organizations and for transferring the knowledge captured in the software product management reference model to even other domains. Such work should be the focus of future research.

## References

1. CMMI Product Team, CMMI for Development, Version 1.3. Carnegie Mellon University (2010)
2. Taylor, S.: ITIL Service design. The Stationery Office, Norwich (2007)
3. Bekkers, W., van de Weerd, I., Spruit, M., Brinkkemper, S.: A Framework for Process Improvement in Software Product Management. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 1–12. Springer, Heidelberg (2010)
4. Pettersson, F., et al.: A practitioner's guide to light weight software process assessment and improvement planning. Journal of Systems and Software 81(6), 972–995 (2007)
5. Stachowiak, H.: Allgemeine Modelltheorie. Springer, New York (1973)
6. Pedreira, O., et al.: A Systematic Review of Software Process Tailoring. ACM SIGSOFT Software Engineering Notes 32(3), 1–6 (2007)
7. Clarke, P., O'Connor, R.V.: The Situational Factors that Affect the Software Development Process: Towards a Comprehensive Reference Framework. Information and Software Technology 54(5), 433–447 (2011)
8. Bekkers, W., et al.: A Situational Assessment Method for Software Product Management. In: 18th European Conference on Information Systems (ECIS 2010), Pretoria, South Africa (2010)

9. Xu, P., Ramesh, B.: Using Process Tailoring to Manage Software Development Challenges. IT Professional 10(4), 39–45 (2008)
10. Basili, V.R., Rombach, H.D.: Tailoring the Software Process to Project Goals and Environments. In: 9th International Conference on Software Engineering (ICSE 1987), Monterey, CA, USA (1987)
11. Huo, M., Zhang, H., Jeffery, R.: A Systematic Approach to Process Enactment Analysis as Input to Software Process Improvement or Tailoring. In: 13th Asia Pacific Software Engineering Conference (APSEC 2006), Bangalore, India (2006)
12. Garvin, D.: Building a Learning Organization. Harvard Business Review 71(4), 78–91 (2000)
13. Briand, L., El Emam, K., Melo, W.: An inductive method for software process improvement: concrete steps and guidelines. In: El Emam, K., Madhavji, N. (eds.) Elements of Software Process Assessment & Improvement. Wiley-IEEE Computer Society (2001)
14. Basili, V.R.: Quantiative Evaluation of Software Methodology. Computer Science Technical Report Series, University of Maryland: College Park, Maryland, USA (1985)
15. Yin, R.K.: Case study research: Design and methods. SAGE Publications (2008)
16. World Trade Organization, Understanding the WTO, World Trade Organization, Geneva, Switzerland (2011)
17. Fricker, S.: Software Product Management. In: Maedche, A., Botzenhardt, A., Neer, L. (eds.) Software for People: Fundamentals, Trends and Best Practices, pp. 53–81. Springer (2012)
18. Hsieh, H.F., Shannon, S.E.: Three approaches to qualitative content analysis. Qualitative Health Research 15(9), 1277–1288 (2005)
19. Helgesson, Y.Y.L., Höst, M., Weyns, K.: A review of methods for evaluation of maturity models for process improvements. Journal of Software Maintenance and Evolution: Research and Practice 24, 436–454 (2012)
20. Robson, C.: Real world research: A resource for social scientists and practitioners-researchers. Blackwell Publishers Ltd., Oxford (1993)
21. van de Weerd, I., et al.: On the Creation of a Reference Framework for Software Product Management: Validation and Tool Support. In: International Workshop on Software Product Management, Minneapolis (2006)
22. van de Weerd, I., et al.: Towards a Reference Framework for Software Product Management. In: 14th IEEE International Requirements Engineering Conference (RE 2006), Minneapolis, MN, USA (2006)

# Harmonizing Software Development Processes with Software Development Settings – A Systematic Approach

Simona Jeners[1], Paul Clarke[2], Rory V. O'Connor[3], Luigi Buglione[4], and Marion Lepmets[2]

[1] RWTH Aachen University, Research Group Software Construction, Germany
simona.jeners@swc.rwth-aachen.de
[2] Regulated Software Research Group, Dundalk Institute of Technology, Ireland
{Paul.Clarke,Marion.Lepmets}@dkit.ie
[3] Dublin City University, Ireland
roconnor@computing.dcu.ie
[4] ETS Montréal / Engineering.IT SpA, Italy
luigi.buglione@eng.it

**Abstract.** The software process landscape is rich in complexity and many alternative software development approaches have emerged over the past 40 years. However, no single software development approach is universally implemented and it seems likely that no single approach can be universally useful. One of the primary reasons that no single approach is universally useful is that no two software development settings are identical. We have assembled a team of recognized academics, who together with industrial collaborators, plan to map the complex world of software processes with the context of software development projects. The results of our initial mapping efforts, reported in this paper, demonstrate that although there are challenges in an undertaking such as this, the outcomes are potentially of considerable value to both software researchers and practitioners.

**Keywords:** Software Process, Situational Factors, Process Improvement, Mappings, Systematic Approach.

## 1 Introduction

When compared with some of the more established engineering disciplines, it has been claimed that the profession of software engineering can be considered to be in its youth [1]. However, arguments to the contrary also exist: that the practice of software development may already be quite mature [2], and that software engineering may not be a true engineering discipline at all [3]. Whether software development is or is not a true engineering discipline may for many practitioners represent an academic debate. In practice, software development is beset with many challenges and constraints. The variety of problems to which software is proposed as a solution is very broad, and the tooling and materials employed in software development are constantly evolving.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 167–178, 2013.

Nonetheless, many general models and frameworks for software development have been published, and some of these approaches have proven to be beneficial.

Owing to the rich variety of software development settings (for example: the nature of the application being developed, team size, requirements volatility), the implementation of a set of practices for software development may be quite different from one setting to another. Process capability and maturity frameworks (CMFs), such as CMMI-DEV [4] and ISO/IEC 15504 [5], recognize that different implementations of software processes are possible and provide mechanisms for assessing any given implementation. Furthermore, CMFs also provide a roadmap for process improvement. However, evidence of the benefits of CMFs is predominately restricted to larger organisations [6], [7]. Limited evidence of the benefits of CMFs for smaller software development settings also exists [8-10]. However, it has been suggested that such approaches may not be suited to the needs of small software development organizations – and it would appear that in practice, smaller organizations tend not to adopt CMFs.

Together with other so-called *traditional* approaches, such as Quality Management Standards (e.g. ISO-9001), CMFs have been criticized for being overly restrictive (or *heavy)* in terms of their ability to support the innovative and speculative nature of software development [11]. As a result, the Agile Manifesto [12] was devised as an alternative philosophy to developing software, addressing some of the limitations of traditional approaches. In particular, the agile manifesto emphasizes the need for working software over extensive documentation, while also promoting the frequent delivery of smaller usable features rather than waiting a long time to deliver a single large system. A number of agile software development approaches, generally termed agile methodologies, have been developed [13], [14]. Furthermore, published studies have demonstrated the benefits of adopting an agile software development approach, including increased productivity, improved time to market [15] and reduced code defect densities [16]. While the advent of agile methodologies has delivered benefits to software development initiatives, it has also been noted that the general philosophy may suffer from a number of limitations. For example, it has been argued that agile development methodologies may require a very skilled software developer, a *premium* developer [17], and that some approaches place an impractical demand on customer collaboration [18].

The preceding paragraphs describe just a small subset of the approaches to software development (herein termed *Improvement Reference Models* (IRMs)) that have been proposed over the past few decades. And despite the benefits of each individual approach, no single approach has been universally adopted. Rather, software development projects and organizations appear to choose a base model that works for them, thereafter adapting and changing their specific processes to address their own specific needs [19]. Therefore, the basic requirement of a software development process is that it "*should fit the needs of the project*" [20]. Although it is relatively straightforward to understand that a software development process should ideally be harmonized with the context within which the software must be developed and delivered, no earlier published research has focused on identifying the relationship between aspects of software development settings (which we term the *situational context*) and the broad dimensions of software development processes. Therefore, this research is motivated to address this gap, and in order to do so, the authors have secured the participation of both industrial and academic collaborators. Together, and over an extended period of time, these collaborators will develop a systematic approach to identify the relationships between factors of situational context and various aspects of the software development process. Our approach could also support the IT projects or IT departments that use frameworks such as ITIL [21] and CMMI-SVC [22]. However, we have chosen to focus first on the software development area.

Our primary goal is to support projects to efficiently achieve their objectives by a systematic improvement of their internal processes. A high Return on Investment (ROI) is a prerequisite for this improvement, i.e. perform improvement initiatives that bring the most benefit and can be managed by the project without risking the project goals and constraints (time, cost and quality).

To support projects, we aim to develop a systematic approach that identifies best practices from different improvement reference models (IRMs) that are best suited for an IT project. Our approach considers the following aspects:

- **Value/Benefit:** The context of the project must be considered to identify the best practices that bring the most benefit.
- **Cost:** The adoption cost of the best practice should not jeopardize the achievement of the project goals.

The remainder of this paper is structured as follows: Section 2 introduces a systematic approach wherein different contributors iteratively map factors of situational context to software development processes. In Section 3 we report on the initial application of this approach and in Section 4, we reflect on the challenges and efficacy of the approach. Section 5 presents a conclusion as well as outlining future work plans.

## 2 Approach

This section outlines a systematic approach adopted in order to map situational factors to IRMs practices. The approach has two main phases (fig. 1): **(1) Trial Approach** – experts perform a subjective mapping between a subset of situational factors and IRM practices; **(2) Broader Mapping Program** – more experts and IT project members evaluate the mappings between a larger set of situational factors and IRMs that will support the improvement of the systematic mapping approach. The **Trial Approach** consists of the following steps:

**T1. Secure the participation of experts for trial.** Our goal is to involve many experts to perform or evaluate mappings between situational factors and various IRM practices.

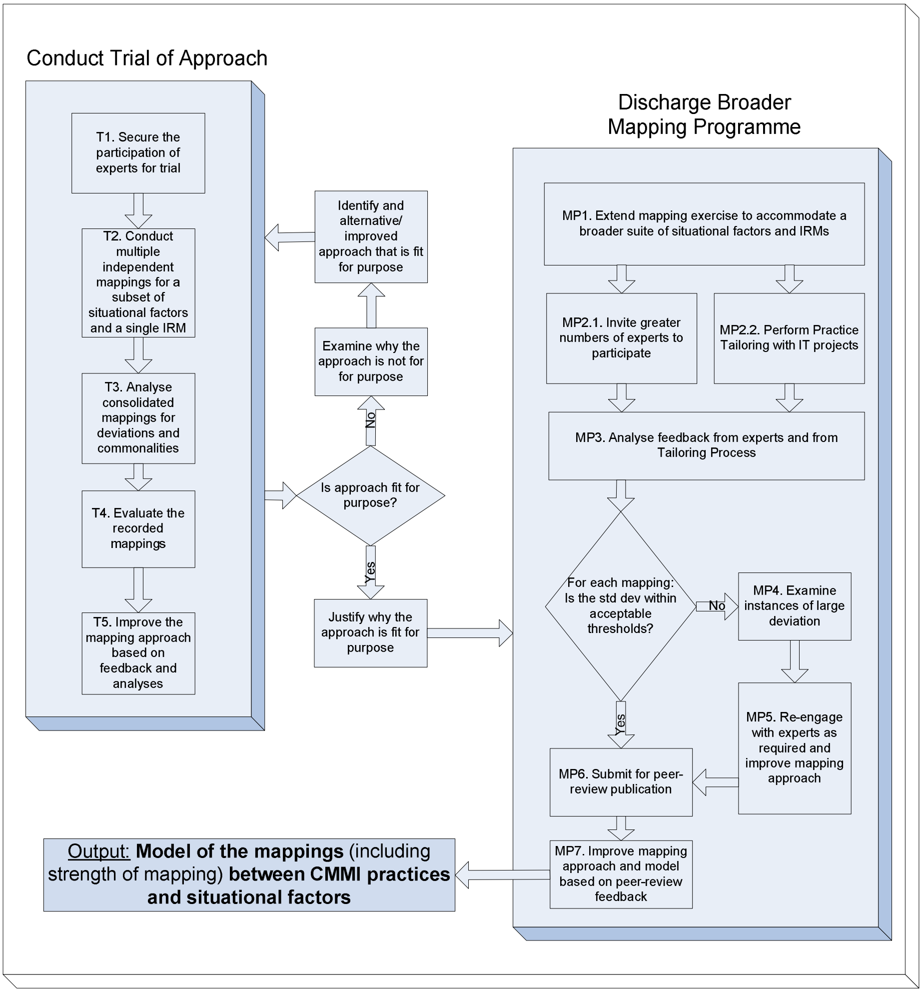



**Fig. 1.** Overview of the proposed mapping approach

**T2. Conduct multiple independent mappings for a subset of situational factors and a single IRM.** Our goal is to perform a series of independent mappings on a subset of the factors and practices. A *performer* conducts subjective mappings of the perceived strength of the relationship between a practice and a situational factor, according to the following four-point ordinal scale:

- **3** – the practice highly supports the project in managing the situation described by the situational factor
- **2** – the practice supports the project in managing the situation described by the situational factor
- **1** – the practice weakly supports the project in managing the situation described by the situational factor

- **0** – the practice does NOT support the project in managing the situation described by the situational factor

The independent results of each performer are consolidated to obtain an overview of the mapping strength for each factor and IRM practice.

**T3. Analyse consolidated mappings for deviations and commonalities.**
Examine the contributions from the various performers and seek to confirm common understandings for the different factors and practices. Note commonalities and address instances of deviation as deemed appropriate. This may result in a revised set of consolidated mappings.

**T4. Evaluate the recorded mappings.**
Conduct an independent evaluation of the consolidated mappings through the use of an *evaluator*. The evaluator performs a review considering not just their subjective opinion but also the previously consolidated performers input - leading to better results.

**T5. Improve the mapping approach based on feedback and analyses.** Mark-up the previous mappings based on the combined feedback. (Note that at once the research advances to this stage, it is envisaged that a number of practitioners will be engaged in the further improvement of the mapping approach prior to discharging the broader mapping programme).

The **Broader Mapping Programme** comprises of the following steps:

**MP1. Extend mapping exercise to accommodate a broader suite of situational factors and IRMs.** Based on our systematic approach, identify the mapping strength for more situational factors and more IRMs (Note that our approach can be extended to address IT projects from other domains (Services, Functional Safety)).

**MP2.1. Invite greater number of experts to participate.** Involve further experts to participate in the evaluation and improvement of the mapping approach. An online survey may help to get feedback on the method and on the results for additional domains (e.g. Services, Functional Safety).

**MP2.2. Perform Practice Tailoring with IT projects.** Evaluate the mapping results by using these results in practice. An industrial partner will choose development projects with different characteristics aiming to identify IRM practices that are best suited to the given project situation(s). We aim to conduct a Tailoring-Workshop with the members of these projects: first, we identify the situational factors that are most relevant for the project; secondly, we provide the project with our mappings, as a recommendation for practice adoption. Based on the benefit and on the cost for the adoption, the project makes a decision which practices should be adopted. During the practice adoption, we aim to collect feedback from the project: did the adoption of the practices bring the desired benefit, i.e. helped managing a certain critical situation in the project?

**MP3. Analyse feedback from experts and from project members.** Consolidate feedback from practitioners and the impact on the mapping framework.

**MP4. Examine instances of large deviation.** Identify the mappings where there is a large deviation between our results and the feedback from the experts and project members.

**MP5. Re-engage with experts as required and improve mapping approach.** In a Retrospective-Workshop with selected experts and project members, examine the large deviations and identify improvements to our systematic mapping approach.

**MP6. Submit for peer-review publication.** The mapping approach along with implementation outcomes is submitted for academic peer-review.

**MP7. Improve mapping approach and model based on peer-review feedback.** The findings collected during the peer-review are used to make final improvements to the mapping approach.

The outputs from the two phases outlined above are (1) a systematic approach to objectively map situational factors and IRM practices; (2) a matrix with the relationships between software development settings and software development processes.

## 3 Application and Results

This section outlines the steps performed so far (T1 to T4) and the results achieved:

**T1. Secure the participation of experts for trial.** Inviting and motivating different experts to participate in the trial. The authors of this paper were all involved in the trial.

**T2. Conduct multiple independent mappings for a subset of situational factors and a single IRM.** First, a set of situational factors and IRMs was defined. To identify the relationships between situational contexts and software development processes, it is important that comprehensive and reliable reference frameworks are employed. For the software development processes, any software process model could potentially be employed. However, of all the process models published to date, the two most comprehensive are ISO/IEC 12207 [23] and the CMMI-DEV. Both of these two resources are comprehensive and have been widely applied in practice; therefore, either was suited to our mapping task. Since our industrial collaborators expressed a strong preference for CMMI-DEV (this was their area of expertise), it was decided that the CMMI-DEV would be employed as the process reference for the mapping exercise.

Regarding the situational context for software development, again a number of possible reference frameworks existed. The work of Xu and Ramesh [24] identifies twenty distinct situational factors, while later works include even greater numbers of factors – for example, Petersen and Wohlin [25] identify twenty-one factors, and Bekkers et al. [26] list thirty distinct factors. However, it is the situational factors reference framework developed by Clarke and O'Connor [27] that is both the most recent and the most comprehensive contribution to date regarding situational context. Clarke and O'Connor [27] have systematically included the earlier identified works in the development of their framework. Furthermore, their situational factors reference framework also incorporates important seminal contributions from a range of related domains, including risk factors for software development (e.g. [28]), software cost estimation (e.g. [29]), and software process tailoring (e.g. [30]). For the initial mapping exercise, we randomly selected two different situational factors: "*performance of application(s)/product(s)*", and "*changeability of requirements*".

As per the process outlined in Section 2, four performers attempted an initial mapping, with a fifth academic performing the evaluation of the mappings. A template was created to document the subjective mappings of each performer (Fig. 2 provides a snapshot). This template contains all the practices of CMMI-DEV ML2 and ML3 (since these two processes are widely used by organizations [31]) categorized by their process areas and maturity levels. For each practice, the performer could specify the mapping strength 0-3 by marking the corresponding cell with "x". The number of "x" indicates the number of performers that agreed to a certain mapping strength. As the mappings are subjective, we introduced a justification column to document the reasoning of the experts for the chosen strength. After the performers finished specifying their mappings independently, their respective inputs were consolidated.

| IRM practices / Situational factor | | Required performance of application(s)/product(s) | | | | |
|---|---|---|---|---|---|---|
| | | Concerned with the performance demands that are placed product(s)/application(s) under development. For example, product(s)/application(s) may be required to process a high number of transaction peRSecond. | | | | |
| | | Consolidated Independent Mappings | | | | Justification |
| CMMI Process Area | | 3 | 2 | 1 | 0 | |
| **Measurement and Analysis** | **MA** | | | | | |
| Establish and maintain measurement objectives that are derived from identified information needs and objectives. | SP 1.1 | x | xxxx | | | If there are specific performance requirements, then it may be necessary to set objectives and measures in relation to the performance of application(s)/product(s). |
| Specify measures to address the measurement objectives. | SP 1.2 | | xxxxx | | | |
| Specify how measurement data will be obtained and stored. | SP 1.3 | | | xxxx | x | Although the collection of measurement data may be importnant where performance is an important consideration, this does not imply that it is necessary to specifiy how the measurements will be obtained or analysed. |
| Specify how measurement data will be analyzed and reported. | SP 1.4 | | | xxx | xx | |
| Obtain specified measurement data. | SP 2.1 | | xxxx | x | | If ther are specific performance criteria to satisfy, then the collection and analysis of the measurement data is going to be necessary. |
| Analyze and interpret measurement data. | SP 2.2 | | xxxxx | | | |
| Manage and store measurement data, measurement specifications, and analysis results. | SP 2.3 | | x | xxxx | | If there are specific performance criteria to satisfy, then the measurement data/results may need to be stored and communicated to stakeholders. |
| Report results of measurement and analysis activities to all relevant stakeholders. | SP 2.4 | | x | xxx | x | |

**Fig. 2.** A Fragment of the mapping

**T3. Analyse consolidated mappings for deviations and commonalities.** Based on the mapping consolidation, we conducted a discussion based on three principles.

**Principle 1:** Instances of significant disparity would be prioritized for discussion. For example, if each of the 4 participants had a different mapping strength for a situational factor to a CMMI-DEV practice, then clearly there was considerable disagreement on the strength of the relationship and hence, a discussion was warranted to establish if there was a lack of common understanding.

**Principle 2:** Instances where one (or more than one) of the participants had considered that there was no relationship between a situational factor and a CMMI-DEV practice (and others disagreed) were also prioritized for discussion. This was considered important as the decision to rule out any relationship between a factor and a practice could have important implications for the overall work.

**Principle 3:** As a general rule, if the reported mapping strengths were clustered in just two or three adjacent cells, such instances could be de-prioritized (with the exception of rule number 2 above – i.e. one of the cells was a 0 [or no relationship] mapping).

In the discussion, we use the idea of the "poker planning"-method for cost estimation [32], asking the contributors with the minimum and maximum strength to justify their selection. This often led to an adjustment to the initial inputs.

**T4. Evaluate the recorded mappings.** The consolidated and analysed mappings were independently evaluated by an experienced academic evaluator (who was not involved in the mapping process up to this point). The evaluator identified the frequencies of provided mapping strengths as a mechanism for taking all views into account and for assisting in calculating the overall mapping between a situational factor and the CMMI-DEV procedure. This led to a series of evidences that as follows:

**Situational Factor 1 – Required Performance of Application(s)/Product(s)**: Process Areas (PA) with the strongest mapping to the performance factor were MA (Measurement & Analysis), PMC (Project Monitoring & Control), SAM (Supplier & Agreement Management), RD (Req. Development) and REQM (Requirement Management), and VAL (Validation). Adding process categories of these PAs, we see that the Support and the Project Management categories (two PAs for each category) were related to ML2; and the Engineering process category (with two PAs) to ML3. When we look at the staged representation of the CMMI-DEV, those four ML2 process areas are effectively requested to have good performance as REQM defines guidelines to manage the project requirements, SAM leads to a good relationship with (sub)providers and assures the fulfillment of requirements for the supplier deliveries, PMC requests monitoring the project results to fulfill its requirements and MA is the basis for this monitoring using and analyzing different metrics. On ML3, RD is the main input for any software lifecycle (SLC) activity.

**Situational Factor 2 – Requirements Changeability:** The PAs with the strongest mapping to requirements changeability were CM (Configuration Management), PMC (Project Monitoring & Control), PP (Project Planning), REQM (Requirement Management), IPM (Integrated Project Management), RD (Requirement Development) and RSKM (Risk Management). In other words, the process categories on ML2 were Support (one PA) and Project Management (three PAs); and on ML3 Project Management (two PAs) and Engineering (three PAs) categories.

In summary, both RD and REQM are grouping practices that aim to collect, define, analyze and manage the requirements (incl. their changes), while PP and PMC are their counter-side in terms of planning and controlling that variability, often expressed in the so-called '*scope creep*' phenomenon, as well described in the IFPUG Function Point Analysis CPM (Counting Practice Manual) [33]. At ML3, IPM defines practices to track and resolve critical dependencies caused by requirements changes with the different stakeholders, while RSKM helps identifying and analyzing the risks that can be caused by the changes.

# 4 Retrospective

In this section we will briefly outline some of the key challenges encountered while executing this study, the actions taken to address them and open challenges for the continued evolution of the research.

An important early task to address was the selection of suitable expert participants and the associated administrative and coordination issues for project execution. The lead researcher used a network of personal contacts, which were initially established at European and international software process conferences. From a starting point of 2 experts, a further 3 were recruited. All correspondence was conducted via email and teleconference facilities (Skype), which was hindered by scheduling/availability of experts, time differences, etc. However, the geographical co-location (Ireland) of 3 of the experts alleviated some of these difficulties.

When conducting the initial mapping exercise, it became apparent that the various experts had applied subjective interpretations regarding certain situational factors and CMMI-DEV practices, which led to inconsistent initial mappings. For example, the situational factor *Commitment of Personnel* was interpreted differently requiring discussions during teleconferences. This led to a description of each situational factor being added to the template to ensure a more consistent interpretation of the factors (refer to Fig. 2, rightmost column, second row). Despite this addition, the situational factor regarding human-centric activities still proved extremely difficult to reconcile among experts regarding different interpretations, resulting in the decision to not include such factors in the initial phase and to more carefully consider these issues at a later stage.

In addition, during this initial exercise there was substantial discussion on the usage of a four-point ordinal scale, with suggestions that a 5 or even 8-point scale could be more appropriate as it could lead to a richer understanding of the relationships. However, to date the decision is to maintain a 4-point scale.

A final point worthy of comment relates to the time and logistical issues surrounding the consolidation of results. This required between 1.5 and 2 hours of intensive discussion per situational factor for 3 experts to analyze and agree. Potential logistical issues would arise here if a larger number of experts were used. In addition the usage of a relatively simple Excel-based spreadsheet made progress with altering and consolidating mappings slow. This could be aided by the creation of an enhanced spreadsheet harnessing macros or possibly a database system. A final remaining challenge to be addressed relates to the selection of an appropriate form of evaluation for both the research approach and outputs. As this work progresses, this will become a more critical consideration. However, at this early stage in the research, this remains an open challenge.

## 5 Conclusion and Future Work

The software process landscape is rich in complexity and many alternative software development approaches have been developed over recent decades. However, no single software development approach is universally implemented or useful. One of the primary reasons for this is the significant variation that is witnessed in software development endeavors. Just one software developer completes some software projects, while other projects require a large team. There is a broad range in the value of software projects, and a wide spectrum to be satisfied in terms of the criticality of operational domain. Some software development efforts are highly innovative with emerging requirements, while other efforts may offer greater requirements certainty earlier in the implementation cycle.

Given such variation in software development settings, it is not surprising to discover a wide variety of approaches to software development. However, although a variety of approaches exist, the authors of this paper contend that insufficient guidance is offered on the activity of tailoring software processes and process improvement efforts to individual settings. Therefore, it is important that further research be dedicated to examining the relationship between software development settings and software development processes. In this respect, we have assembled a team of recognized academics, who together with industrial collaborators, plan to map the complex world of software processes with the context of software development projects.

In this paper, we have outlined an approach to identify mappings between processes and project settings. We have reported on our initial experiences from the application of the process. These initial findings highlight some of the significant challenges that our mapping project has to overcome. For example, we have had to expand the previously available descriptions of situational factors with concise definitions that permit a more consistent interpretation of the role of individual factors. We have also discovered that the role of human-centric factors, such as the commitment of employees, is difficult to agree upon. Hence, the mapping of human-centric factors has been postponed to a later phase. Since the broader mapping program represents a very large undertaking, we plan to complete the work in an iterative fashion over a broad period of time. Therefore, the essential purpose of this paper is to highlight the need for this research, identify an approach to ground the mapping exercise, and to report on the initial mapping of two situational factors to all of the practices of CMMI-DEV. In the future, we aim to decrease the subjectivity of such mappings by proposing an approach to systematically map situational factors to processes and by the involvement of more experts from research and industry. Therefore, we envisage that later reports of this research activity will contain mapping tables that will serve as valuable new resources for both practitioners and researchers. Such mapping tables will identify, for the first time, the combined view of researchers and practitioners on the relationship between aspects of situational contexts and software development processes.

## References


1. Jacobson, I., Ng, P., McMahon, P., Spence, I., Lidman, S.: The Essence of Software Engineering: The SEMAT Kernel. Queue 10(10), 40–51 (2012)
2. Schaefer, R.: Software Maturity: Design as Dark Art. SIGSOFT Software Engineering Notes 34(1), 1–36 (2009)
3. Denning, P.J., Riehle, R.D.: The Profession of IT. is Software Engineering Engineering? Communications of the ACM 52(3), 24–26 (2009)
4. SEI: CMMI for development, version 1.3. Software Engineering Institute, CMU/SEI-2006-TR-008, Pittsburgh, PA, USA (2010)
5. ISO/IEC: IS0/IEC 15504: Information technology - process assessment, part 1 to part 5. International Organisation for Standardization, Geneva, Switzerland (2005)
6. Herbsleb, J., Goldenson, D.: A systematic survey of CMM experience and results. In: Proceedings of the 18th International Conference on Software Engineering (ICSE 1996), pp. 323–330. IEEE Computer Society, Los Alamitos (1996)
7. Gibson, D., Goldenson, D., Kost, K.: Performance results of CMMI-Based Process Improvement. Software Engineering Institute, Carnegie Mellon University, CMU/SEI-2006-TR-004, Pittsburgh, Pennsylvania, USA (2006)
8. Cepeda, S., Garcia, S.: Is CMMI Useful and Usable in Small Settings? CrossTalk. The Journal of Defense Software Engineering 21(2), 14–18 (2008)
9. Cater-Steel, A., Rout, T.: SPI long-term benefits: Case studies of five small firms. In: Oktaba, H. (ed.) Software Process Improvement for Small and Medium Enterprises - Techniques and Case Studies. IGI Global, Hershey (2008)
10. Laporte, C.Y., Desharnais, J.M., Abouelfattah, M., Bamba, J.C., Renault, A., Habra, N.: Initiating Software Process Improvement in Small Enterprises: Experiments with Micro-Evaluation Framework. In: Proceedings of the International Conference on Software Development, pp. 153–163 (2005)
11. Dyba, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology 50(9-10), 833–859 (2008)
12. Fowler, M., Highsmith, J.: The Agile Manifesto. Software Development, 28–32 (2001)
13. Beck, K.: Extreme programming explained: Embrace change. Addison-Wesley, Reading (1999)
14. Schwaber, K., Beedle, M.: Agile software development with SCRUM. Prentice Hall, Upper Saddle River (2002)
15. Reifer, D.J.: How Good are Agile Methods? IEEE Software 19(4), 16–18 (2002)

16. Fitzgerald, B., Hartnett, G., Conboy, K.: Customising Agile Methods to Software Practices at Intel Shannon. European Journal of Information Systems 15(2), 200–213 (2006)
17. Constantine, L.: Methodological Agility, http://www.ddj.com/architect/184414743
18. Greer, D., Conradi, R.: Software Project Initiation and Planning - an Empirical Study. IET Software 3(5), 356–368 (2009)
19. Coleman, G., O'Connor, R.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software 81(5), 772–784 (2008)
20. Feiler, P., Humphrey, W.: Software process development and enactment: Concepts and definitions. SEI, Carnegie Mellon University, CMU/SEI-92-TR-004, Pittsburgh, Pennsylvania, USA (1992)
21. Taylor, S., Cannon, D., Wheeldon, D.: ITIL The Cabinet Office (2011)
22. SEI: CMMI for Services, Version 1.3, CMU/SEI-2012-TR-034. Software Engineering Institute, Pittsburgh, PA, USA (2010)
23. ISO/IEC: ISO/IEC 12207-2008 - systems and software engineering – software life cycle processes. ISO, Geneva, Switzerland (2008)
24. Xu, P., Ramesh, B.: Software Process Tailoring: An Empirical Investigation. Journal of Management Information Systems 24(2), 293–328 (2007)
25. Petersen, K., Wohlin, C.: Context in industrial software engineering research. In: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 401–404. IEEE Computer Society, Washington (2009)
26. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: Proceedings of the Second International Workshop on Software Product Management (IWSPM 2008), pp. 41–48. IEEE Computer Society, Los Alamitos (2008)
27. Clarke, P., O'Connor, R.V.: The Situational Factors that Affect the Software Development Process: Towards a Comprehensive Reference Framework. Journal of Information and Software Technology 54(5), 433–447 (2012)
28. Benaroch, M., Appari, A.: Financial Pricing of Software Development Risk Factors. IEEE Software 27(5), 65–73 (2010)
29. Boehm, B., Clark, B., Horowitz, E., et al.: Software cost estimation with cocomo II. Prentice Hall PTR, Upper Saddle River (2000)
30. Cameron, J.: Configurable Development Processes. Communications of the ACM 45(3), 72–77 (2002)
31. SEI: CMMI for SCAMPI SM Class A Appraisal Results 2012 Mid-Year Update, SEI, CMU (2012)
32. Grenning, J.: Planning Poker. Renaissance Consulting – April 2012 (2002)
33. IFPUG: Counting Practices Manual (Version 4.3). International Function Points User Group (October 2009), http://www.ifpug.org/?p=83

# Review of Critical Success Factors Related to People in Software Process Improvement

Sussy Bayona, Jose A. Calvo-Manzano, and Tomás San Feliu

Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software,
Universidad Politécnica de Madrid, Facultad de Informática
Campus Montegancedo, 28660 Boadilla del Monte Madrid, Spain
sbayonao@hotmail.com, {jacalvo,tsanfe}@fi.upm.es

**Abstract.** Information systems are designed, constructed, and used by people, using a set of defined processes previously deployed, as a result of software process improvement initiative. Software process improvement is not a purely technical task, but a complex psycho-socio-technical process. This paper presents a systematic review to identify the categories most used in literature to classify software process improvement success factors and the critical success factors related to people. We found several studies related to critical success factors, but only 10 of them proposed a critical success factors classification. The quantitative data from a systematic review were analyzed using descriptive statistics. As a result the following common categories are identified: organization, people and process. The critical success factors related to people are presented in this paper.

**Keywords:** Critical success factors, Taxonomy, Process improvement, Systematic review.

## 1 Introduction

Today, organizations are in a continuous process of transformation. Software Process Improvement is an approach to increase the efficiency and effectiveness of a software development organization and to enhance software products [1]. Achieving a successful process improvement initiative is a challenge for organizations despite the widespread use of sound project management practices and process improvements models and standards [2]. Information systems are designed, constructed, and used by people [3] and the design of software processes used is not a purely technical task, but a complex psycho-socio-technical process. The literal meaning of “psycho-socio-technical" is the dynamic relationship of psychological, social and technical effects that continuously interact with each other. Psychological, because deploying new processes is a change introduced by people and affects people. People have emotions and much of their behavior is based on emotions. Sometimes the decisions and attitude of management usually clash with the emotions of employees. The social context of communication between employees and senior executives and managers is crucial

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 179–189, 2013.

when the processes are defined and deployed. Differences between the description and how the processes have been applied can cause problems in software projects. Technical is related tools, techniques, methods, procedures and technology.

A systematic literature review related to critical success factors in software process development identified 25 factors [2]. The following factors were identified: (1) *appropriate development processes and method (process)* and (2) *effective project management and methods (project manager).*

The first factor is related to the process defined by the organization (Process Engineering) to be used in the software process development. *Process Engineering* provides the *process* perspective and is concerned with how to define and build processes and understand their performance. It is related to (1) how to specify processes with adequate empirical evidence of their performance, and (2) engineer, assemble, combine, and reuse process components to meet required performance targets.

The processes must be adapted to the organization's needs and aligned to the business goals [4, 5]. The processes must be clear, unambiguous and measured. However, organizations have problems when implementing them [6, 7]. Once the processes are defined by Process Engineering, they are deployed throughout the organization. *Process deployment* is about getting the processes into actual practice. Process deployment provides the *process* perspective and the success depends strongly on people at all levels: individual, group or organizational. If they are not managed properly, the organization does not achieve institutionalization of their processes. Issues in process institutionalization arise due to the fact that most of these efforts focus on technical issues and the issues related to people are ignored [6]. There is a set of factors that influence the successful software process improvement initiatives [4, 5, 6, 7, 8, 9]. McDermid and others [10] agree that human factors have been ignored in process improvement and this has affected the efficiency of improvement initiatives. Hall and Wilson [11, 12], suggest that experiences, opinions and perceptions of the practitioners have an indirect effect on software quality.

So, it is necessary to identify what *the critical success factors related to people* are and to know their impact on process improvement initiatives. Classifying the success factors into appropriate categories will contribute to the state-of-the-art knowledge about software process improvement, and help practitioners to focus their improvement programs on them. In this paper, we identify through a systematic review the critical success factors related to people to be taken into account when a process improvement initiative is designed. For this purpose, the systematic review technique [13] was used. A systematic review is a formal and verifiable process that researchers carry out to document the state of knowledge on a specific topic. We believe the results of this study will benefit practitioners and researchers.

The rest of this paper is structured as follows: Section 2 describes the methodology of the systematic review. Section 3 presents a report of the systematic review results, while Section 4 present the critical success factors related to people. Finally, Section 5 draws some conclusions based on the review carried out.

# 2 Research Methodology

In this section we describe the design and the execution of the systematic literature review. In order to establish the critical success factors related to people, a systematic review protocol was performed. A systematic review is a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [13]. The systematic review allows: (1) to review the relevant works in the study area, (2) to control, evaluate and confirm the results, and (3) to identify research gaps that will lead to new topics for research. In order to develop the systematic review protocol of this research the protocol proposed in [14], and the guide proposed by Kitchenham et al. [13] were taken into account.

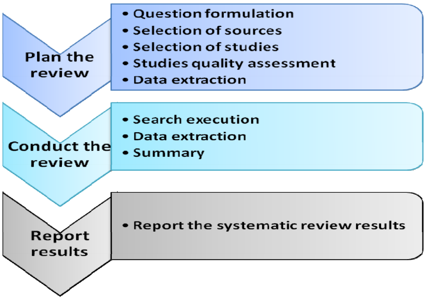



**Fig. 1.** Systematic review steps

Fig. 1 outlines the research process we have used. A description of the research process related to *Plan the review* and *Conduct the review* follows. The *Report results* are described in Sect. 3.

## 2.1 Plan the Review

In this step, a review protocol that specifies the methods that will be used to undertake a specific systematic review is developed.

### 2.1.1 Question Formulation

- Q1: What are the categories and subcategories that have been used to classify a critical success factor for process improvement initiatives?
- Q2: What are the critical success factors related to people?

### 2.1.2 Selection of Sources

The sources include specialized digital sources of software engineering literature, such as *Science @ Direct, IEEE Explore, ACM Digital library, SpringerLink, Institute for Scientific Information (ISI) Web Knowledge, Wiley InterScience*; articles and specialized conference presentations such as Software Engineering Process Group, European Systems & Software Process and Innovation (EUROSPI), SPICE as well as reports, articles and presentations by the Software Engineering Institute (SEI), Cross-Talk, and IT Governance.

The terms used in the search string were identified from previous experience in the subject area. The search strings include the words: "process deployment", "software process improvement", "SPI", "critical success factors", "key factors", "human factors", "social factors", "taxonomy", "catalog", "barriers", "motivators" and "demotivators" connected with the words "CMM", "CMMI", "SPICE", "ISO9000" "MPS" and "IDEAL".

The search mechanisms of the available search engines are different. It was necessary to design and use different search strings for each database, maintaining the equivalence. The list of sources includes relevant journals such as: *Information and Software Technology, Software Process Improvement and Practice, Journal of Systems and Software, IEEE Transactions on Software Engineering, IEEE Software, Software Quality Journal, and Journal of Defense in Software Engineering.*

#### 2.1.3 Selection of Studies

Once the sources are defined it is necessary to describe the process and the criteria for studies selection and evaluation. The inclusion criteria are:

- Include empirical studies of process improvement and process deployment. Papers must (1) discuss the critical success factors classification, and (2) identify the categories and subcategories.
- Papers containing keywords that match those defined in the search string.
- Papers whose title, summary or content are related to the topic.

The exclusion criteria are:

- Papers that are based only on a particular opinion.
- Short papers.

Papers that are not relevant for the research questions, or are not related specifically to the study, are excluded. To select the studies, the criteria of inclusion and exclusion perform are applied in four steps:

- Perform a preliminary selection of the studies, applying the search string on each selected source.
- Review the list of studies to verify if they comply with the inclusion and exclusion criteria. To select the primary studies the following steps are performed:
  - Step 1. Read the title, if the information is insufficient, perform Step 2.
  - Step 2. Read the article abstract. If the information is insufficient, perform Step 3. Read the full text. If they provide sufficient information, the study is selected and saved (sufficient information is related to the papers when they discuss the critical success factors and their classification).
- Evaluate the list of studies that are included and excluded. In case of disagreements in evaluation, the researcher had to reach an agreement on the selection of studies.
- Evaluate the list of selected papers.

#### 2.1.4 Studies Quality Assessment

To evaluate the quality of the studies, a quality checklist was used as a quality assessment instrument. The quality assessment checklist contains the following questions:

- SQA1: Is primary study relevant to the research that is being done by the researcher? To evaluate the quality of the papers with respect to their ability and suitability to answer our research questions, it is assumed that they are reliable and have sufficient quality to contribute to this systematic review.
- SQA2: Do studies provide enough information to identify the categories used in order to group the critical success factors? The main purpose of quality assessment studies is to assess the impact of the quality of the primary studies on the conclusions.

The responses were: Yes=1, Partially=0,5 and No=0 for both quality assessment checklist.

#### 2.1.5 Data Extraction

To extract the relevant data from each paper (papers selected following the steps described in Sect. 2.1.3) and to standardize the way information has been represented, a data extraction form was designed. The data extraction form has to be filled out with the relevant information: author(s), publication year, study type, category, definition category, subcategory, definition subcategory, critical success factors by category and their description. Keeping the data will allow a more detailed analysis later.

### 2.2 Conduct the Review

After the planning phase the review protocol can be applied. The review protocol was reviewed and approved by the systematic review team members. To validate the review protocol a pilot of data extraction process was carried out. The studies found in the pilot were validated with the studies found in a previous literature review performed by one of the authors.

#### 2.2.1 Search Execution

Following the protocol (Sect. 2.1.3), and using the search engines that count each of the identified digital sources and submitting the search string, a total of 1412 studies were found in the databases after eliminating the duplicate studies. Only 10 primary studies that discussed the critical success factors classification and identified the categories were selected. The selected papers were assessed taking into account the quality assessment criteria described in Sect. 2.1.4. The studies contain case studies, experiments, surveys, experiments etc. The studies found are the basis for following the systematic review process. The list of selected papers was evaluated by two authors. Also, the list of selected papers was compared with a list of papers identified by previous traditional review of the literature.

#### 2.2.2 Data Extraction

From each paper we extracted the data using the form designed (Sect. 2.1.5). Software SPSS was used to store the data extracted in the forms. As a result of the data extraction process two databases were elaborated. The first related to critical success factors (critical success factors, critical success factors definition, category, subcategory, and author) and the second to critical success factors classification (author, study type, year, category, definition category, subcategory, definition subcategory).

#### 2.2.3 Summary

This section presents the data resulting from the selected studies. **Statistical calculus results:** with the purpose of understanding the relation between results obtained, a statistical analysis was conducted using the information extracted from the primary studies (see Sect.3 and Sect. 4). **Results presentation in tables:** the results obtained from the systematic review were summarized in tables and displayed in graphics. **Numbers of studies:** a total of 10 primary studies.

## 3 Report of the Systematic Review Result

After the systematic review execution, the results were summarized and analyzed. In the analysis, statistical tools were used to determine: (1) overview of selected studies, (2) the categories used to classify the critical success factors, (3) the list of critical success factors by people category and (4) the list of subcategories by categories. We found several studies related to critical success factors in process improvement initiatives but only 10 of them presented the critical success factors classified. The papers were published between 1995 and 2010. The most used techniques were surveys and case studies. In some studies, the authors used a combination of techniques. The studies were carried out in large and small organizations.

### 3.1 Categories Identified

Table 1 summarizes the categories used to classify the critical success factors by author. The maximum categories number is four (30% of the studies), 30% of the studies defined three categories and the minimum is two categories (40% of the studies). An important issue to take into account in a taxonomy design is the number of categories. Not more than 7 categories are recommended at the first level. Then it is necessary to know the number of categories proposed by authors. In Table 1, the column labeled Number (N) shows the number of categories defined by the authors to classify the critical success factors. Savcenko et al. [15] performed a systematic review (a secondary study) and combined success factors into groups. They propose 9 categories and one of them is *People related factors* category. This category included: job satisfaction and motivation of the organization´s members. The terms used by the author to name the category in some cases are different. The *People* category is named *Human factors* by Hall et al [21]. Others categories used to classify the critical success factors are: Awareness [23], Support [23], and Context [24].

**Table 1.** Categories used to classify the critical success factors by author

| Category | Author | Category | Author |
|---|---|---|---|
| Organizational | [17][18][21][23][25] | Resource management | [20] |
| Process | [17][19][24] | Information sharing | [20] |
| *People* | *[18][19][21]* | Implementation factors | [21] |
| Technology | [18][19] | Environmental-dependent | [22] |
| Knowledge | [18] | Environmental-independent | [22] |
| Project | [25] | Products | [24] |
| Commitment/Resources | [16] | Role of management | [20] |
| Management style | [16] | Project organization | [20] |

### 3.2 Categories Identified and Critical Success Factors

From the selected studies, we identified the categories used to classify the critical success factors for process improvement. The frequency analysis method helps to organize the qualitative data, once the data collection process is completed. Frequency analysis is a descriptive statistical method that shows the number of occurrences for each category identified by author. The statistical information such as the number of occurrences and percentages of each variable (category) can be represented in the shape of frequency distribution. In this case, the frequency distribution has two elements: (1) categories identified, and (2) number of times the authors make reference to each category. A criterion for determining the importance of the category used to classify the critical success factors was established based on frequency analysis (frequency greater than 2).

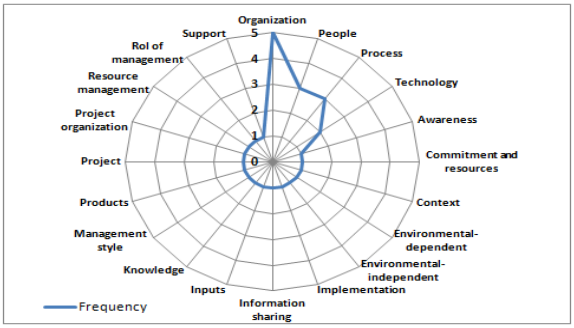



**Fig. 2.** Categories identified

The categories most commonly used are: Organization, People and Process (see Fig. 2). The Organization category according El Eman et al [17] are those variables that characterize the organization undergoing software process improvement, and the characteristics of the organizational software process improvement effort itself. The Organization category was also identified by Kaltio et al [18], Hall et al [21], Niazi et al [23], and Mohd et al [25]. The Process category according Wilson et al [24] is related to the method used to develop, implement and maintain the program and includes: let the objectives determine the improvements, have an independent software process improvement team, and do not use the program to assess individuals. The Process category also was identified by Hantos et al [19], and El Eman [17].

**Table 2.** Critical success factors by People category

| N | Description | Author | Critical success factors |
|---|---|---|---|
| 1 | People, representing the human resources and their personal skills and capabilities that are used to execute the process instance. | [18] | Time allocation, Knowledge, Standing motivation, Motivation, Abilities |
| 2 | The people, process, and technology categorization is a fair translation of Moore's whole product concept. | [19] | Staff involvement* * |
| 3 | This slightly unconventional angle emphasizes the inextricable relationship between software process improvement and people. Indeed many of the case studies reported in the literature consider human factors critical to software process improvement success. | [21] | Software process improvement leaders (change agents, opinion leaders), Management commitment, Staff involvement |

* The terms people, human factors were harmonized to people, ** Not imposed

Once the categories were identified, the different critical success factors to be grouped under each category were identified. A matrix was performed in order to collect the information related to: category, author, category description, critical success factors, and their description. This paper focuses on critical success factors related to people. Table 2 shows a list of critical success factors by People category defined by the authors of the primary studies.

## 4 Critical Success Factors Related To People

A description of critical success factors identified in the People category follows.

- *Software process improvement leaders: software process improvement managers contribute significantly to software process improvement success. The im*portance of the change agents and opinion leaders in managing software process improvement and the need to appoint highly respected people to software process improvement. Software process improvement is inherently linked with the change [26].
- *Management commitment*: the criticality of gaining and maintaining management support for software process improvement. A manager with high commitment was replaced by one with less commitment; previous process improvements were lost.
- *Staff involvement*: is related to involving development staff in software process improvement. The need to generate a culture of process ownership is emphasized, as is the need to value software process improvement as real.
- *Time allocation*: This is related to the time allocated and available to carry out the activities. Sufficient resources are to be allocated. Based on experience, the organization can gain an understanding of suitable resource allocation for the different roles.

- *Knowledge*: the knowledge required to perform Software Process Improvement program activities.
- *Standing motivation*: formal title and rank, as well as the more informal respect that the person has from the other members of the organization has an impact on the ability to achieve results in Software Process Engineering.
- *Motivation*: influences the likelihood and the extent to which a person will actually use the time allocated to carry out the tasks and may affect the quality of the results as well. (Motivation includes: achievement of visible results, existence and nature of feedback, clarity of set targets, management commitment).
- *Abilities:* skills and knowledge that can be taught and acquired, and characteristics that people have. Examples include: use of a tool, process knowledge, understanding of how the organization works or how to deal with people, openness and natural networking skills, charisma or leadership abilities.

### 4.1 Limitations

The paper proposes three relevant categories of software process improvement success factors that can be useful to reach a consensus about classifications of factors. A possible validity threat about the results of the systematic literature review is that the proposed categories are extracted from papers that explicitly provide some classification of software process improvement factors. But other relevant papers, which can even report success factors with more empirical evidences than selected ones, can be excluded because they do not classify the factors.

## 5 Conclusions

In this research study we use the technique of systematic review to identify the categories used to classify the critical success factors in software process improvement. Through the systematic review a large number of factors affecting Software Process Improvement initiatives discussed in the literature were identified, but only 10 primary studies are related to establishing a critical success factors classification. There is divergence in the existing studies, the classification of the critical factors in literature is varied and there is no consensus of authors. The maximum categories number defined by the diverse authors is four and the minimum is two. To identify the categories, the research data from the literature were categorized and coded in order to perform frequency analysis. There is no evidence that a standard classification system of critical success factors has been developed. The categories are: organization, process and people. This paper is focused on the People category. The critical factors related to people were: software process improvement leaders, change agents, and opinion leaders to manage software process improvement initiatives; management commitment; staff involvement, time allocation, motivation, abilities, skills and knowledge. The results show that a process improvement initiative is impacted by social aspects since the processes deployed are defined and used by people. A process improvement requires leaders and change agents in order to put the processes into practice manage change and maintain staff motivation. The staff resist initiatives that they perceive as imposed (on them). Staff involvement in process improvement activities can minimize the resistance and keep staff motivated. Also, a process improvement initiative requires resources to perform the activities, and people with skills, abilities and knowledge. Having identified the critical factors of Software Process improvement related to organization and processes we will be able to develop a/the taxonomy for the critical success factors of process deployment in order to standardize the terms, definitions, and identify the subcategories and elements.

**Acknowledgments.** This work was sponsored by everis Foundation and Universidad Politécnica de Madrid through the Research Chair in Software Process Improvement for Spain and Latin American Region.

## References

[1] Unterkalmsteiner, M., Gorschek, T., Moinul, A., Kian Cheng, C., Bayu, R., Feldt, R.: Evaluation and Measurement of Software Process Improvement-A Systematic Literature Review. IEEE Transactions on Software Engineering 38(2), 398–424 (2012)
[2] Damaševičius, R.: On The Human, Organizational, and Technical Aspects of Software Development and Analysis. Information Systems Development, pp. 11–19 (2010)
[3] Mohd, N., Shamsul, S.: Addressing a critical success factor for software projects: A multi-round Delphi study of TSP. International Journal of the Physical Sciences 6(5), 1213–1232 (2011)
[4] Clarke, P., O'Connor, R.V.: The influence of SPI on business success in software SMEs: An empirical study. Journal of Systems and Software 85(10), 2356–2367 (2012)
[5] Niazi, M., Babar, M.: Motivators of Software Process Improvement: An Analysis of Vietnamese Practitioners' Views. In: 11th International Conference on Evaluation and Assessment in Software Engineering (EASE), Keele University, UK, April 2-3 (2007)
[6] Hall, T., Rainer, A., Baddoo, N.: Implementing Software Process Improvement: An Empirical Study. Software Process Improvement and Practice 7, 3–15 (2002)
[7] Pitterman, B.: Telcordia technologies: the journey to high maturity. IEEE Software 17(4), 89–96 (2000)
[8] Nasir Mohd, H., Ahmad, R., Hassan, N.: Resistance factors in the implementation of software process improvement project. Journal of Computer Science 4(3), 211–219 (2008)
[9] Montoni, M., et al.: Taba Workstation: Supporting Software Process Deployment Based on CMMI and MR-MPS.BR. In: Münch, J., Vierimaa, M. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 249–262. Springer, Heidelberg (2006)
[10] McDermid, J., Bennet, K.: Software Engineering research: A critical appraisal. IEEE Proceedings on Software Engineering 146(4), 179–186 (1999)
[11] Hall, T., Wilson, D.: Views of software quality: a field report. IEEE Proceedings on Software Engineering 144(2), 111–118 (1997)
[12] Hall, T., Wilson, D.: Perceptions of software quality: a pilot study. Software Quality Journal 7(1), 67–75 (1998)
[13] Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in software engineering. EBSE Technical Report EBSE-2007-01 (2007)

[14] Biolchini, J., Mian, P., Natali, A., Travassos, G.: Systematic review in software engineering. Technical report, Systems Engineering and Computer Science Department UFRJ Brasil (2005)
[15] Savcenko, A., Tanveer, B.: Software Process Improvement Success Factors - through systematic review and industrial survey. Master thesis, Blekinge Institute of Technology (2009)
[16] Goldenson, D., Herbsleb, J.: After the Appraisal: A systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Software Engineering Institute (SEI), TR CMU/SEI-95-TR-009 ESC-TR-95-009 (1995), `http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr009.95.pdf`
[17] El Emam, K., Goldenson, D., McCurley, J., Herbsleb, J.: Modeling the likelihood of software process improvement. Empirical Software Engineering 6(3), 207–229 (2001)
[18] Kaltio, T., Kinnula, A.: Deploying the defined software process. Software Process: Improvement and Practice 5(1), 65–83 (2000)
[19] Hantos, P., Gisbert, M.: Identifying software productivity improvement approaches and risks: construction industry case study. IEEE Software 17(1), 48–56 (2000)
[20] Sassenburg, H.: Assessing readiness for (software) process Improvement, Methods & Tools, pp. 18–24 (Spring 2002), `http://www.methodsandtools.com/archive/archive.php?id=35`
[21] Hall, T., Rainer, A., Baddoo, N.: Implementing Software Process Improvement: An Empirical Study. Software Process: Improvement and Practice 7(3), 3–15 (2002)
[22] Guerrero, F., Eterovic, Y.: Adopting the SW-CMM in a Small IT Organization. IEEE Software 21(4), 29–35 (2004)
[23] Niazi, M., Willson, D., Zowghi, D.: A maturity model for the implementation of software process improvement: an empirical study. Journal of Systems and Software 74(2), 155–172 (2005)
[24] Wilson, D., Hall, T., Baddoo, N.: A framework for evaluation and prediction of software process improvement success. Journal of Systems and Software 59(2), 135–142 (2007)
[25] Mohd, H., Ahmad, R., Hassan, N.: Resistance Factors in the Implementation of Software Process Improvement Project in Malaysia. Journal of Computer Science 4(3), 211–219 (2008)
[26] Koining, S., Sicilia, M., Messnarz, R., Barriocanal, E., Garre, M., Siakas, K., Clarke, A.: Understanding the relation of SPI and SR: A proposed mapping of the SPI Manifesto to ISO 26000:2010. In: Proceedings of the EUROSPI 2011, pp. 7.27–7.37 (2011)

# A Proposal for the Improvement Predictability of Cost Using Earned Value Management and Quality Data

Adler Diniz de Souza and Ana Regina Cavalcanti Rocha

COPPE/UFRJ - Universidade Federal do Rio de Janeiro
Programa de Engenharia de Sistemas e Computação
Av. Horácio Macedo, 2030, Prédio do Centro de Tec., Bloco H, Sala 319,
Caixa Postal 68511 – CEP 21941-914 – Rio de Janeiro, RJ
adlerunifei@gmail.com, darocha@centroin.com.br

**Abstract.** Although the Earned Value Management – EVM technique is utilized by several companies in different sectors for over 35 years, in order to predict cost results, many studies detected vulnerabilities in the technique, among them: (i) there is instability in the cost and time performance indicators during the Project; (ii) there is a trend of deterioration in the cost and time indicators when the projects are near their end, and others. The present study proposes an extension of this technique, through the integration of the history of quality performance data as means of improving the technique's cost predictability. The proposed technique is evaluated and compared to the traditional technique through different hypothesis tests, utilizing data from the simulation projects. The technique was more accurate and more precise than the traditional EVM for the calculation of the Cost Performance Index – CPI and the Estimate At Completion – EAC.

**Keywords:** Earned Value Management, Cost Performance Index – CPI, Estimate At Completion – EAC, Software Quality, Measurement and Analysis.

## 1 Introduction

The Project Management Institute – PMI [1] currently estimates that approximately 25% of global Gross National Product – GNP is spent on projects and that close to 16.5 million professionals are directly involved in project management throughout the world. This volume of projects and changes in an increasingly competitive global scenario generates the need of faster results, with higher quality, lower costs and shorter deadlines. To assess whether or not a project will reach its goals of time, cost and quality, several measures are collected during its execution, and various performance indicators are produced and periodically analyzed. When there are deviations larger than the tolerance in some performance indicators, corrective actions are undertaken in order to improve them. Among the main available techniques for the analysis of cost and time - EVM, is considered the most reliable [2].

The EVM is a technique that integrates scope, time and cost data to measure project performance and predicts its cost and schedule, based on the current performance of the team. However, it does not integrate data quality project to predict the

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 190–201, 2013.

cost and time. The technique gained great importance when, in 1967, the United States Department of Defense - DoD, began requiring its use as a means to control the costs of contracted projects [2]. Several formulas derived from EVM measurements are available and have been studied in the last 15 years [2]. However, studies intended to improve the predictability of the results of time and cost have remained stagnant over the last decade and still require further studies [2]. Particularly in Software Engineering some model references like CMMI-Dev [3] requires the gathering of measures and the development of indicators of the most important processes, responsible for achieving the business goals of organization.

This paper propose an improvement in the EVM, with an integration of quantitative information of the subprocesses related with quality, which are relevant for the business goals related to cost. The main objective is to use the proposed technique like a performance model to predict the final cost of software projects.

## 2 Earned Value Management

The method of EVM allows the calculation of variances and performance indices of cost and time, which generate forecasts for the project, given its performance so far, allowing the implementation of actions aimed at correcting any deviations [2]. This allows the project's manager and your team to adjust their strategies, make trade-offs based on the goals, on the project's the current performance, on trends, and on the environment in which the project is being conducted [4].

According to [1], EVM has an essential role in the success of projects, responding to managerial issues that are considered critical, such as: i) how efficiently are we using our time? ii) when is the project likely to be finalized? iii) how efficiently are we using our resources ? iv) how much above or below the budget will we be at the end of the project, given the current productivity of the team? The method of EVM is based on three basic measures, which are derived to generate other measures and performance indicators. These basic measures are: i) Planned Value - $PV_{Acum}$: represents planned costs accrued up to a given date; ii) Earned Value - $EV_{Acum}$: represents the budgeted cost of the work performed up to a given date; and iii) Actual Cost - $AC_{Acum}$: represents the actual cost of work performed [1].

The basic measures discussed do not allow making predictions of cost to complete the project, and answer the questions posed above. For this purpose it is necessary to generate performance indicators, among which the most widely used is the Cost Performance Index - $CPI_{Acum}$ (Accumulated until the actual date). The CPI is a measure of the value of work performed compared to the actual cost or progress made in the project. It shows how efficiently the project team is using its resources, calculated as:

$$CPI_{Acum} = \frac{EVAcum}{ACAcum}\ , \tag{1}$$

The CPI is considered the EVM's most critical indicator, because it measures the cost efficiency of work performed [1].

As the project progresses, the project team can develop a forecast for the Estimate At Completion - EAC, which may differ from the Budget At Completion – BAC

(BAC is the project`s cost baseline), based on project performance [1]. The EAC provides the final estimate of cost and is given by the equation below (assuming the cost performance remains the same):

$$\text{EAC} = \frac{BAC}{CPIAcum}, \tag{2}$$

## 3 Problem Description

The EVM technique makes use of the CPI to make cost projections at the end of the project. This index is the subject of several discussions on its applicability and reliability to make projections, as reported in works carried out by [4], [5], [6], [7] and [8].

The major focus of the discussion is the CPI Accumulated - $\text{CPI}_{\text{Acum}}$ stability.

According to [9], stability can be defined as a state of statistical control that provides with a high degree of confidence, the performance prediction of some variable in the immediate future.

Florac [9] states that the stability of a process is considered by many as the core of the management of processes, and it is essential for companies to produce products according to what has been planned and to improve processes in order to produce better and more competitive products.

A study reported in [5] evaluated the $\text{CPI}_{\text{Acum}}$ stability of several projects of the Department of Defense (DoD), and found that the index was stable after 20% of project execution. This study generalized the result, concluding that any project could use the technique reliably, after 20% of project execution. This information was used as a criterion for retaining or cancelling projects in the U.S. government, which showed $\text{CPI}_{\text{Acum}}$ below 0.9 after 20% of project execution, because according to the study, the stability of the index was evidence that a project with poor CPI was unrecoverable.

However, several other studies have questioned the generalization of these results in different contexts (projects developed outside the scope of DoD), and showed different results, i.e., they showed instability in cost performance indexes for most of the project [5], [6], [7] and [8].

Claiming that the $\text{CPI}_{\text{Acum}}$ is unstable and varies widely during the execution of a project avoids making accurate projections of cost estimate at the end of the project (EAC), unless one knows or has any expectation that this variation is due to factors already known.

The proposed evolution of the EVM technique presented in the next section suggests that the lack of quality data in the traditional EVM technique may be one of the causes for the wide variation in CPI and significant drop of performance near the end of the execution of projects observed by [2], [4] and [6].

Thus, one of the justifications for the $\text{CPI}_{\text{Acum}}$ instability is the occurrence of quality non-compliances that have not been fixed and therefore were not considered in the calculation of performance indicators. Thus, considering the $\text{CPI}_{\text{Acum}}$ for a given project, its deviation from the baseline should not be evaluated in isolation. This indicator should be evaluated together with another quality indicator that shows the impact of identified and expected non-compliances in relation to project cost measures, thus reflecting the quality cost in the cost indicator.

# 4 Proposal of Quality EVM

If it is obvious that in the Budget At Completion - BAC is no longer viable, the project manager must prepare an Estimate at Completion - EAC. Developing an EAC forecast involves finding estimates or forecasts of future events and conditions for the project based on information and knowledge available at the time of prediction. Information on work performance include past performance of the project and any information that could impact it in the future [1].

Quality data of process are information that may impact future performance of the cost performance index and are not used to make these projections.

All costs incurred in the management of the project quality activities are called quality costs, and refer to the total cost of all efforts related to quality during the product life cycle [13].

According to [13], quality costs are usually divided in two categories:

- Compliance costs: costs that are allocated throughout the project for activities to prevent failures such as: i) training, ii) documentation of processes, iii) tests and iv) inspections.
- Noncompliance costs: costs that are allocated in the course and after the project execution, attributed to failures. They can be divided into two categories, internal failure costs, e.g.: i) re-work, ii) wastes and external failure costs, such as: i) loss of reliability, ii) product warranty and iii) loss of market.

The proposed technique is based on two basic measures, which are used to generate other measures and performance indicators.

Total Expected Noncompliance – TENC represents total NC expected for a given process of a project, namely, the TENC is the process quality baseline. This measure is generated taking into account the project size. Thus, given the size of a project, the expected number of noncompliance can be calculated using the following equation:

$$\text{TENC} = \text{Size} * \text{INCH}, \tag{4}$$

Where:

- The software size can be given in function points, use case points or other size measure, as long as the technique used in all projects is the same.
- INCH is the noncompliance rate that indicates the number of expected noncompliance for a given process, given by the following equation:

$$\text{INCH} = \frac{\sum N.C.\ of\ all\ executed\ projects}{\sum Size\ of\ all\ executed\ projects}, \tag{5}$$

This indicator is generated for each process that is used by the technique, and will be an organizational indicator generated based on historical data of various projects.

The TENC has an effort and a cost associated to it. They are denominated TENC (h) and TENC ($). The TENC (h) measures the effort to fix the noncompliance presented by the TENC and it is calculated by the equation below:

$$\text{TENC (h)} = \text{TENC} * \text{AEE}, \tag{6}$$

Where:

- The AEE is the Average Estimated Effort to fix the noncompliance and it is calculated by the following equation:

$$\text{AEE} = \frac{\sum Effort\ to\ fix\ N.C.of\ all\ Executed\ Projects}{\sum Number\ of\ N.C.of\ all\ Executed\ Projects}, \tag{7}$$

The TENC ($) measure the cost to fix the noncompliance presented by the TENC and it is calculated by the equation below:

$$\text{TENC (\$)} = \text{TENC (h)} * \text{Man Hour Cost}, \tag{8}$$

Where:

- The Man Hour Cost is the average cost of the project hours used by the company using the technique.

The Identified Noncompliance INC represents detected NC, corresponding to a particular process in a particular project. INCs are given by the following equation:

$$\text{INC} = \sum N.C., \tag{9}$$

The measures described above can be used to analyze the current performance of projects; however, it is not possible to make future projections of their behavior using these measures in an isolated way. Quality forecast can be obtained by calculating performance indicators using the measures presented. The quality performance indicators are shown below:

The QPI Quality Performance Index is an indicator that shows how efficient the quality of a particular process is. Given a certain date, the indicator show if the number of noncompliance is higher or lower than expected, allowing making projections about the future quality performance through NC Estimate to Complete (NCEC), as shown in Fig. 1 – NCP projections using QPI. This index is given by the following equation:

$$\text{QPI} = \frac{\text{ENC}(d)}{\text{INC}(d)}, \tag{10}$$

Where:

- ENC (d): represents the total NC expected for a given date. At the end of the project execution, ENC (d) must be equal to TENC;
- INC (d) represents the total NC identified for a given date.

Values below 1 for the indicator mean that a higher number of noncompliance than expected are being found. Values above 1 indicate that a lower number of noncompliance than expected are being found.

The purpose of the quality performance indicator is to predict the amount of future noncompliance, given the current performance, and assess the impact of quality performance for project costs.

However, it is important that the process used is known, stable and that its auditors have the necessary skills to perform their activities. This will ensure that QPI > 1, for instance, reflects an improvement in the process quality and not the auditor's lack of skill in finding problems in this process.

As the project progresses, the project team can develop a New forecast for the NC Estimate to Complete (NCEC), which may be different from Total Expected noncompliance - TENC based on quality performance. This new estimate should only be calculated if the trend is that the Quality Performance Index - QPI remains the same, i.e., in case the TENC is no longer feasible.

To evaluate the TENC feasibility, project progress and extent of changes that have occurred in relation to quality in a given time interval will be used at the quality baseline, or the TENC measure, and its projection called NC Estimate to Complete (NCEC), as illustrated in fig.1, which can be calculated by the equation below:

$$NCEC = \frac{TENC}{QPI} \quad , \tag{11}$$

The positive or negative variations in the number of noncompliance can bring impacts on project costs. The effort and cost related to prevention, detection and resolution of bugs and noncompliance typically belong to their respective baselines, in any project. However, companies rarely take into account quality information during the monitoring and control of cost and schedule. Thus, when quality produces results different from those expected, traditional cost performance indicator (CPI) may present wrong information. An example of this statement occurs when CPI indicates that the project is within expected budget; however, there are more identified noncompliance than expected for the project.

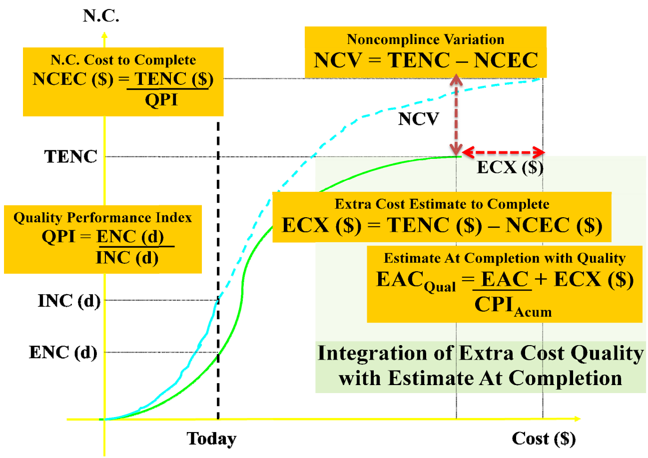



**Fig. 1.** TENC Projections Using QPI

Thus, the proposed technique suggests that the NC Estimate to Complete (NCEC) should be used to obtain the Noncompliance Variations (NCV) on the TENC. This measure informs how different the quality is from expected, as shown in the following equation:

$$NCV = TENC - NCEC, \tag{12}$$

Based on NCV, the cost variation should be calculated and incorporated into project Estimated At Completion. It is done, through the addition of the noncompliance cost variation to the tradicional EAC. Thus, the equation is as follows:

$$EAC_{Qual} = \frac{BAC}{CPIAcum} + ECX\ (\$), \tag{13}$$

Where:

- ECX ($): Noncompliance cost variation, corresponding to the cost (positive or negative) of noncompliance variation.

The noncompliance variation (NCV), will be measured in the X-axis of fig. 1, and will be represented by Extra Cost Estimate to Complete (ECX ($)):

$$\text{ECX (\$)} = \text{TENC (\$)} - \text{NCEC (\$)}, \tag{14}$$

# 5 Planning of the Study

The study's objective was to answer the following question: "Is the EVM traditional technique more accurate and precise than the EVM technique with quality?". Thus, the following hypotheses were set up to evaluate the accuracy of the techniques:

- **$H0_{Accuracy}$**: the traditional EVM technique is as accurate as the EVM technique with quality. $H0_{Accuracy} =$ (Error $EAC_{EVM}$ – Error $EAC_{Quality} = 0$).
- **$H1_{Accuracy}$**: the traditional EVM technique is less accurate than the EVM technique with quality. $H0_{Accuracy} =$ (Error $EAC_{EVM}$ – Error $EAC_{Quality} > 0$).

A similar hypothesis was identified and tested to assess the precision of the techniques.

Three more questions and secondary hypotheses, similar to the first one, were defined, but they were intended to answer if the proposed technique represented more accurately and precisely the traditional technique at the beginning (25% executed), at the middle (50% executed), or near the end (75% executed) of a project.

The techniques presented in section IV were evaluated through a feasibility study, in which the objective was to measure the precision and accuracy of both techniques and compare them.

The accuracy of the proposed technique was calculated using the equation below:

$$EAC_{Quality}\text{Accuracy} = \left|1- \frac{\text{AC Correction Process}}{\text{TENC (\$)}}\right|, \tag{15}$$

The accuracy of the traditional technique was calculated by the equation below:

$$EAC_{Tradicional}\text{Accuracy} = \left|1- \frac{\text{AC Correction Process}}{\text{PV Correction Process}}\right|, \tag{16}$$

The results of the Eq. 15 and Eq. 16 may be a positive or negative value. When the average is calculated a positive value may compensate a negative value, and the average accuracy may masked (i.e. the average of two errors of -20% and +20% is 0%). In this study it's a problem because it doesn't reflect the real error.

To avoid this problem the EAC Accuracy of the techniques was calculated using the absolute value in both equations (Eq 15 and Eq 16). Both average EAC Accuracy was calculated by the equation below:

$$\text{Average}_{\text{ EAC Accuracy}} = \frac{\sum_1^N \text{EAC Accuracy}}{\text{N}}, \tag{17}$$

To measure the techniques' precision, the variability of the CPI techniques was compared to the last estimate, i.e., how much an estimate varied in relation to the previous one, at each of the moments when it was measured. The $CPI_{Acum}$ variation was calculated from the techniques, in relation to the last $CPI_{Acum}$ estimation. Meaning how much the $CPI_{Acum}$ estimation varied in relation to the previous one. It was done through the equation below:

$$\text{Variation}_{\text{CPIActivity(N)}} = \left|1 - \left(\frac{\text{CPIActivity(N+1)}}{\text{CPIActivity(N)}}\right)\right|, \tag{18}$$

Once again the absolute value was used to measure the $CPI_{Acum}$ variation.

The variation was measured by project activities. To test the hypotheses the average variation of projects was calculated using the equation below:

$$\text{Average}_{\text{Variation}} = \frac{\sum_{1}^{N} \text{Variation CPIActivity(N)}}{(N-1)}, \tag{19}$$

One of the main difficulties presented by [2], [7], [10], and [11] in studies related to this one, was the lack of project performance data, available for studies. Thus, it was decided to validate the proposed technique through the performance of project simulations, similarly to the studies conducted by [7], [10] and [11]. Microsoft Excel was used to generate and store effort and non-compliances data and consequently cost (random() function).

## 6 Feasibility Study

The largest cost component in a software project are the man-hours necessary for product development, all the necessary simulations required for the calculation of the base measures and indicators of traditional EVM were based on the planned effort and actual effort for a set of activities of possible processes of any given project, whereby these activities were calculated using the random() function of the MSExcel tool.

**Table 1.** Values Passed to the Random Function for the Initial Simulation

| Process | Variation $H_{Est}$ | Variation $H_{Real}$ | No.of activities |
|---|---|---|---|
| Process 01 | 8 – 30 | 3 – 40 | 12 |
| Process 02 | 3 – 10 | 3 – 10 | 26 |
| Process 03 | 3 – 12 | 1 – 12 | 26 |
| Process 04 | 3 – 17 | 1 – 25 | 26 |

The initial simulations have 4 processes with the variation in the random function shown by Table 1. It was assumed, as premises for the generation of the CPI of the proposed technique that: the quality history data (i.e. number of non conformities, estimated effort to fix the non conformities and real effort to fix the non conformities) of all previously executed projects utilizing a specific process were available. To simulate the existence of non-compliances, columns "$HO_{Est}$", "Occur?", “Fixed”,

"$AC_{Qual}$", "$AC_{Trad}$" of the table 2 were used. Column "$HO_{Est}$" represents the estimated effort to fix the non-compliance and contains the "random" function, which can generate a number between 4 and 12 hours for lines from 1 to 10.

The column "Occur?" contains the "random" function, which can generate only numbers 0 or 1, determining the existence or absence of non-compliance.

The $AC_{Qual}$ column corresponds to the estimated actual cost of the non-compliance correction activity and was calculated using the estimated effort, multiplied by column "Occur?" and by the average cost of the activity, i.e., if the "Occur" column was equal to zero, the $AC_{Qual}$ column would also be equal to zero. Thus, a project with CPI equal to 1, and with a large number of uncorrected non-compliances suggests that there are cost performance problems. This may be one of the justifications for the known problem of loss of cost and schedule performance at the end of the project observed by [2], [4] and [12].

The $AC_{Trad}$ column represents the actual cost for the non-compliance correction and is calculated by multiplying "$HO_{Est}$", "Occur?", "Fixed?" columns, and the "Fixed?" column" also contains the "random" function, with values ranging from 0 to 1, determining whether the non-compliance was fixed or not. If the "Occur?", "Fixed" columns are equal to zero, $AC_{Trad}$ is also equal to zero. This cost is usually added to the AC in the traditional EVM methodology.

**Table 2.** Quality Simulation Data

| NC | $HO_{Est}$ | Occur? | Fixed? | $AC_{Qual}$ | $AC_{Trad}$ |
|---|---|---|---|---|---|
| 1 | 4 | 0 | 0 | 0 | 0 |
| 2 | 7 | 0 | 0 | 0 | 0 |
| 3 | 5 | 1 | 1 | 60 | 60 |
| 4 | 10 | 1 | 1 | 120 | 120 |

The execution of 10 projects was then simulated, with collection of their measurements and calculation of the proposed indicators (equations of 1 to 10). The fig. 2 shows on the X axis each one of the compared projects, and on the Y axis, the sum of the errors of $EAC_{25\%}$, $EAC_{50\%}$ and $EAC_{75\%}$ for each one of the projects. The gain in accuracy using the proposed technique was at times 10 times better than the traditional technique, in the project 4, which suggests that the proposed technique can be 10 times more accurate than the traditional technique, in the simulation.

Statistical tests were performed based on Table 3 and Table 4 to confirm whether the differences in accuracy and precision found in the application of the techniques were significant, thereby approving or rejecting the previously presented hypotheses. The Action tool was used to test the hypotheses of T paired samples, at the 95% significance level. The analysis of the data in Table 3 and Table 5, allows to state, at the outset, that the proposed technique provides more accuracy in the cost estimates, when the project is at 25%, 50% and 75% of execution, and in general taking the three reported moments, at a 95% confidence level.

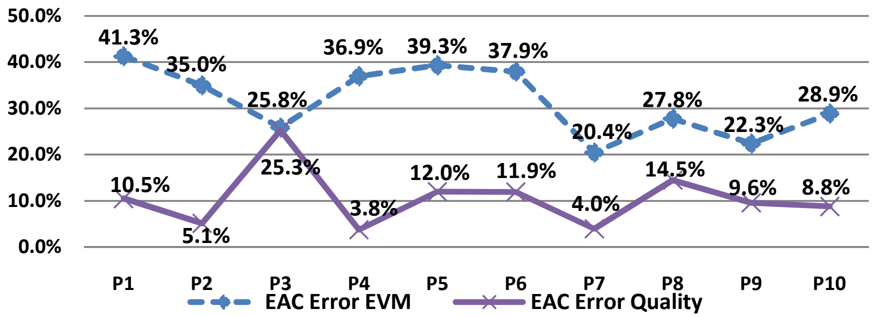



**Fig. 2.** Accuracy – Total EAC Error between the Techniques

**Table 3.** Accuracy (Error Between Thechniques Estimates) of the EAC in Percentage

| %Executed | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | Ave. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25% Trad. | 15,4 | 13,0 | 1,5 | 17,3 | 16,4 | 13,7 | 7,0 | 3,6 | 5,6 | 12,1 | 10,6 |
| 25% Qual. | 1,8 | 0,5 | 17,4 | 1,1 | 6,2 | 3,8 | 0,1 | 9,9 | 0,5 | 4,1 | 4,5 |
| 50% Trad. | 10,3 | 12,0 | 9,2 | 10,0 | 8,9 | 11,0 | 5,4 | 10,4 | 4,7 | 6,7 | 8,8 |
| 50% Qual. | 3,5 | 2,9 | 2,5 | 2,4 | 4,1 | 6,4 | 2,7 | 2,9 | 4,9 | 1,4 | 3,4 |
| 75% Trad. | 15,6 | 10,0 | 15,1 | 9,6 | 14,0 | 13,2 | 8,0 | 13,8 | 12,0 | 10,1 | 12,2 |
| 75% Qual. | 5,2 | 1,7 | 5,4 | 0,2 | 1,7 | 1,7 | 1,1 | 1,8 | 4,1 | 3,2 | 2,6 |

**Table 4.** Precision (Variability Between Thechniques Estimates) of the CPI in Percentage

| %Executed | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | Ave. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25% Trad. | 15,4 | 13,0 | 1,5 | 17,3 | 16,4 | 13,7 | 7,0 | 3,6 | 5,6 | 12,1 | 10,6 |
| 25% Qual. | 1,8 | 0,5 | 17,4 | 1,1 | 6,2 | 3,8 | 0,1 | 9,9 | 0,5 | 4,1 | 4,5 |
| 50% Trad. | 2,3 | 9,2 | 5,4 | 3,3 | 4,7 | 1,8 | 1,8 | 2,6 | 4,4 | 1,0 | 3,67 |
| 50% Qual. | 0,3 | 8,9 | 2,9 | 0,6 | 3,4 | 0,0 | 1,5 | 0,4 | 6,6 | 1,6 | 2,62 |
| 75% Trad. | 15,6 | 10,0 | 15,1 | 9,6 | 14,0 | 13,2 | 8,0 | 13,8 | 12,0 | 10,1 | 12,2 |
| 75% Qual. | 5,2 | 1,7 | 5,4 | 0,2 | 1,7 | 1,7 | 1,1 | 1,8 | 4,1 | 3,2 | 2,6 |

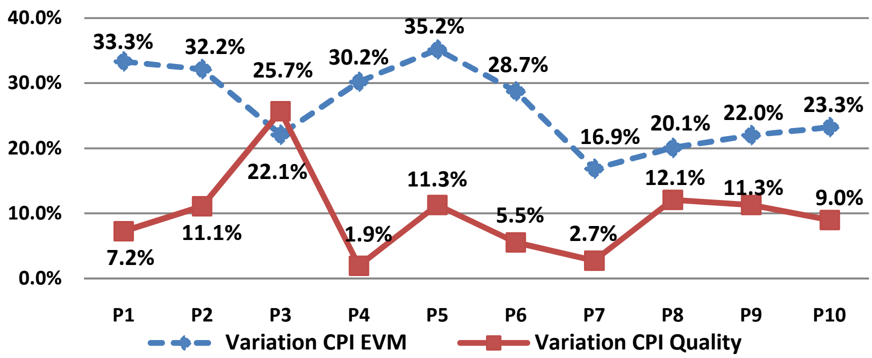



**Fig. 3.** Total Variability of the CPI Between the Techniques

The gains in precision shown in fig. 3 were up to 15 times superior to those of the traditional technique, which suggests that the proposed technique can be 15 times more precise than the traditional one. The analysis of the data in Table 4 and 5 also allows to state, that the proposed technique provides more precision in the cost estimates, when the project is at 25%, 50% and 75% of execution, and in general, taking the three reported moments, at a 95% confidence level.

**Table 5.** Tests of Hypothesis of Accuracy (Error) and Precision (Variability)

| Hypothesis | Test | T | *P* | Conclusion |
|---|---|---|---|---|
| $H0_{Accuracy}$ | $Error_{EAC.EVM} - Error_{EAC.Quality} > 0$ | 6,46 | $5{,}9 \times 10^{-5}$ | Refute H0 |
| $H0_{25\%\ Accuracy}$ | $Error_{EAC.EVM25\%} - Error_{EAC.\ Quality\ 25\%} > 0$ | 1,93 | 0,0424 | Refute H0 |
| $H0_{25\%\ Accuracy}$ | $Error_{EAC.EVM50\%} - Error_{EAC.\ Quality\ 50\%} > 0$ | 6,39 | $6{,}3 \times 10^{-5}$ | Refute H0 |
| $H0_{75\%\ Accuracy}$ | $Error_{EAC.EVM75\%} - Error_{EAC.\ Quality\ 75\%} > 0$ | 14,8 | $6{,}1 \times 10^{-8}$ | Refute H0 |
| $H0_{Precision}$ | $Var_{CPI} - Var_{CPI.Quality.} > 0$ | 4,03 | 0,0015 | Refute H0 |
| $H0_{25\%\ Precision}$ | $Var_{CPI25\%} - Var_{CPI.Quality.25\%} > 0$ | 1,93 | 0,042 | Refute H0 |
| $H0_{50\%\ Precision}$ | $Var_{CPI50\%} - Var_{CPI.Quality.50\%} > 0$ | 2,30 | 0,0197 | Refute H0 |
| $H0_{75\%\ Precision}$ | $Var_{CPI75\%} - Var_{CPI.Quality.75\%} > 0$ | 14,8 | $6{,}2\times10^{-8}$ | Refute H0 |

Any study can present threats that can affect the validity of the results. Therefore the threats of this study will be presented in two categories: i) threats to internal validity and ii) threats to external validity. According to [14] the internal validity observes if the treatments really cause the expected results. In this study, the expected results are: i) decrease the $CPI_{Acum}$ variability and consequently the EAC variability and ii) Decrease the error in the EAC estimate. Both expected results were achieved with the application of the proposed technique. According to [14], the external validity verifies if it's possible to generalize the results. It's necessary to consider that the technique was validated through a simple simulated project. However, the traditional EVM and the proposed technique use a small information set to make a cost projection. And it's important to consider that any variable or project feature omitted from project simulation will affect both techniques (it will cause the same error in both techniques). Nevertheless, it isn't possible to generalize the result.

## 7 Conclusion

This study described the proposal of a technique of EVM, which integrates quality data as a way to improve the predictability of project costs. The study consisted of a feasibility study based on simulated project data with the purpose of determining whether the technique was more accurate and precise compared to the traditional technique, at the beginning (25% of execution), at the middle (50% of execution) and at the end (75% of execution) of the project. In order to evaluate the feasibility of the proposed techniques, several tests of hypotheses were performed about the different research questions posed during the validation of the techniques. The simulation assumed as premise for the utilization of the $EVM_{Quality}$ technique, that the project had a data quality history of all processes of the utilized life cycle. The study aim was achieved, and the proposed technique showed more precision and more accuracy than the traditional technique at the beginning, middle and end of project execution. All the tests of hypotheses showed that the results were significant at the 95% significance level. The next step to this work is use data of real projects to validate the proposed technique like in [8].

## References

1. PMI, Project Management Body of Knowledge-PMBOK Newton Square, PMI (2013)
2. Zwikael, O., et al.: Evaluation of Models for Forecasting the Final Cost of a Project. Project Management Journal 31(1), 53–57 (2000)
3. SEI, S.E.I., CMMI® for Development, V1.3, CMU/SEI-2010-TR-033, SEI (2010)
4. Henderson, K., Zwikael, O.: Does Project Performance Stability Exist A Re-examination of CPI and Evaluation of SPI(t) Stability. Cross Talk 21 (2008)
5. Christensen, D., Heise, S.R.: Cost Performance Index Stability. National Contract Management Journal 25, 7–15 (1993)
6. Vandevoorde, S., Vanhoucke, M.: A comparison of different project duration forecasting methods using earned value metrics. Project Management Journal 24, 289–302 (2006)
7. Souza, A.D., Rocha, A.R.C.: A proposal for the improvement of the technique of Earned Value Management utilizing the history of performance data. In: Proceedings of the Twenty-Fourth International Conference on Software Engineering & Knowledge Engineering - SEKE, pp. 753–759 (2012a)
8. Souza, A.D., Rocha, A.R.C.: A proposal for the improvement the predictability of project cost using EVM and Historical Data of Cost. In: 35th International Conference of Software Engineering-ICSE. ACM SRC, San Francisco (accepted February 2013b)
9. Florac, W.A., Carleton, A.D.: Measuring the Software Process: Statistical Process Control for Software Process Improvement. Addison-Wesley (1999)
10. Iranmanesh, H., Mojir, N., Kimiagari, S.: A new formula to “Estimate At Completion” of a Project’s time to improve “Earned value management system”. International Journal of Project Management (2007)
11. Souza, A.D., Rocha, A.R.C.: A proposal for the improvement of the technique of EVM utilizing the history of performance data. In: Proceedings of the Twenty-Fourth SEKE, pp. A3–A4 (2012b)
12. Lipke, W.: Independent Estimates at Completion – Another Method. Cross Talk The Journal of Defense Software Engineering 17(10), 32 (2004)
13. Putnam, L.H.: Five Core Metrics: The Intelligence Behind Successful Software Management. Dorset House (2003)
14. Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wessl, A.: Experimentation in software engineering: an introduction. Springer (2012)

# Involvement of Stakeholders in Software Processes Improvement to Reduce Change Resistance

Mirna Muñoz[1], Jezreel Mejia[1], Jose A. Calvo-Manzano[2], and Tomás San Feliu[2]

[1] Centro de Investigación en Matemáticas
Av. Universidad no 222, 98068 Zacatecas, México
{mirna.munoz,jmejia}@cimat.mx
[2] Facultad de Informática, Universidad Politécnica de Madrid Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain
{joseantonio.calvomanzano,tomas.sanfeliu}@upm.es

**Abstract.** Organizational software process improvement offers a key opportunity for organizations to become more efficient. However, most of time implements software process improvements initiatives in organizations become a path full of obstacles mainly because stakeholders feel it as an imposition of anybody processes and its implementation as a threat of their jobs. As a result, most of the time the effort in the implementation of software process improvement fails, stakeholders feel frustrated and organizations are more convinced than ever that they must continue doing their work as before even when they do not have the expected results in their job performance. This paper presents an overview of how can be involved stakeholder throughout the implementation of software process improvements so that they feel key elements in order to have a successful software process improvement initiative. Therefore the new processes are perceived as own and their adoption as an evolution of their job that helps them to be more efficient and to have a better job performance.

**Keywords:** software process improvement, knowledge management, software supporting tools, multi-model environment.

## 1 Introduction

Organizational process improvement offers a key opportunity for organizations to become more efficient, therefore, more competitive [1]. As consequence, software process improvement initiatives is logical way to be competitive in the software industry [2][3][4].

However, although many organizations are motivated to improve their software processes, very few know how to do so in a proper way. One of the problems of introducing software process improvement in organizations is the difficulty that an organization faces when the new processes are implemented. In this context, the resistance that stakeholders have in the adoption of the new processes is a key element, since the new processes are perceived as someone else processes. As a

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 202–213, 2013.

result, two scenarios can arise in organizations: the resistance to the implementation of software process improvement increases and the process improvement does not have the expected results [5].

In this context, authors such as O'Connor, Basri, Janh and Nielsen [6][7] have identified the involvement of stakeholders in the implementation of software process improvement as a key aspect in order to achieve a successful software process improvement, so the implication or involvement of stakeholders as a dynamic teams in a process improvement project allows to get better results [6].

The goal of this paper is to present an overview of how the stakeholders can be involved since the beginning of the software process improvement, so the resistance to change is reduced.

This paper is structured as follows: section two introduces to methodology; section three shows the stakeholders identified for the methodology; section four shows a set of activities proposed by the methodology in order to reinforce the prevention of resistance to change; section five presents the case study analysis focused on stakeholders' involvement and finally, section six present the conclusions.

## 2 MIGME-RRC Methodology

MIGME-RRC is a methodology for a gradual and continuous software process improvement focusing on minimizing change resistance called MIGME-RRC (by its Spanish acronym) [8].

This research work mentions MIGME-RRC methodology because this methodology allows to implement software process improvements with a completely involvement of stakeholders since the first improvement phases.

MIGME-RRC is a methodology that has been developed taking knowledge from different areas such as knowledge management; change management and multi-model environment, as follows:

*Knowledge management:* systematic approach that allows the capture, codify, use and operation of knowledge and experiences to develop better tools, methods and the ability to use them [9].

*Change management:* process of planning, organizing, coordinating and controlling internal and external components in order to ensure that process changes are implemented with the minimum deviation compared to approved plans and overall changes introduction goals [10].

*Multi-model environment:* involves all cultural aspects and the knowledge that advises the use in each process a mix of best practices from more than one model or standard to achieve the organization's business goals [11].

Besides, the methodology highlights three concepts throughout all its phases: best practices, business goals and business indicators. These concepts allow to focus the improvement depending on the organization needs.

MIGME-RRC methodology is formed of fourth phases, the phases and their activities are showed in Figure 1.

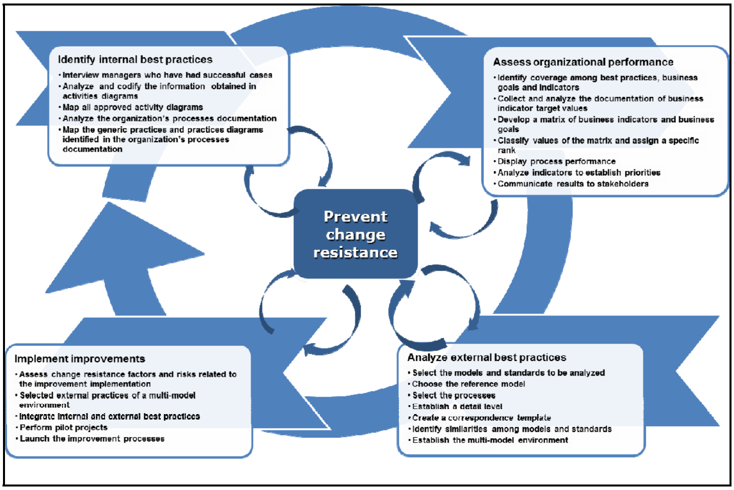



**Fig. 1.** MIGME-RRC methodology phases

As Figure 1 shows, MIGME-RRC proposes a different way to implement a software process improvement in an organization as follows: first it analyses how the organization works by identifying its best practices; after, it establishes the performance of its best practices, comparing the business indicators achievement with the identified best practices; then, it analyzes the best practices of different standards and models and selects those practices that best fit the way the organization works, and finally, depending on the internal and external best practices dependences and their impact on achieving the business indicators, new processes and their implementation sequence are defined. Besides, all its phases are focus on preventing resistance to change.

In order to apply the knowledge from change management and knowledge management throughout performing MIGME-RRC a set of activities has been defined as follows:

1. *Change management activities*

- *Identify internal best practices*: (1) stakeholders' involvement; (2) observe behavior, describe and classify behavior and identify risk focusing on middle management and process users; (3) understand organizational work culture; and (4) establish communication channels.

- *Asses the organizational performance*: (1) communicate the results of process performance; (2) highlight the need to implement a process improvement to achieve the established business goals; and (3) observe behavior, describe and classify behavior and identify risk, focusing on senior management.

- *Analyze external best practices*: (1) select just those models and standards accorded to the organizational work culture; and (2) select the external practices to be candidate to the new processes accorded to the organizational work culture.
- *Implement process improvements*: (1) analyze change resistance factors and risk associated with process implementation and establish actions to prevent them; (2) analyze the difficulty level of adoption of external best practices; (3) select pilots projects, focusing on early adopters; (4) make the material of process presentation based on the target staff (level of interest in change and influence); (5) let the new processes to be available for all stakeholders; (6) establish adequate communication channels as follows: top-down (allow to transmit all relevant information to senior manager from middle management and process users); bottom-up (allow to collect feedback and experience using the new processes) and lateral (allows to reinforce commitment to achieve the work); and (7) allow the organization to adapt the new process at a pace of change supported by them.

2. *Knowledge management activities*

- *Identify internal best practices*: (1) extract knowledge; (2) understand and select knowledge; and (3) characterize and structure knowledge.
- *Asses the organizational performance*: (1) Analyze, understand and select information related to process performance as a historical data; and (2) structure and store information selected as historical data in order to have process assets.
- *Analyze external best practices*: (1) Analyze, select and structure external knowledge through analyzing external best practices and (2) structure the new knowledge, so that, it could be easily adopted within the organization.
- *Implement process improvements*: (1) analyze the impact of external best practices toward the achievement of the business goals; (2) analyze the internal best practices and external best practices dependences; (3) define the new processes taking into account external best practices and internal best practices; (4) structured the new processes based on the organization‘s needs; and (5) collect feedback of new processes and the experiences of their use and store them as process assets.

## 3 Stakeholders' Involvement

A feature of MIGME-RRC methodology is the involvement of stakeholders through all phases as dynamic teams to get better results.

To understand MIGME-RRC stakeholders' involvement, it is important to focus on the main set of stakeholders identified for MIGME-RRC. This section lists the main set of stakeholders that have an important participation throughout MIGME-RRC phases.

- *Senior management*: staff that has the power to take strategic decisions refers to business goals. In this set staff such as account managers, senior managers, improvement facilitator and partners are included.
- *Middle management*: staff that has the power to take operational decisions toward achieving the business goals. In this set staff such as project managers, quality managers or quality management group; and process improvement managers or process improvement group are included.
- *Process users*: staff whose work is directly related to the use of software processes to do their work. Or staff whose job is not directly related to the use of the software process but they need information or product produced as output of software process performance. In this set staff such as team leader, team engineers (planning, quality, process, development and support) are included. Besides, depending on the type of process, the methodology allows to involve in an interactive way stakeholders who are interested in participating, providing important information of how the organization works.

Next, a briefly description of how MIGME-RRC involves the stakeholders is included:

3. *Identify internal best practices*: middle management staff and process users have an important role because they are the source of the organization's tacit knowledge. Therefore, they are the only ones who should validate it. It is important to highlight that in this first phase of the methodology the validations of best practices are considered a key activity in order to formalize the organization's knowledge because organizational knowledge is formalized in processes, using its best practices as a base. Besides, at the end of this phase the "documentation findings" are showed to senior management staff in order to be aware of the real organizational software process and the actual gaps in process documentation so they can appreciate a first methodology work product that helps to increase their trust and confidence in the methodology.
4. *Assess the organizational performance*: senior management staff have an important role in this phase for three main reason: first, they establish the business goals and set target values to them; second, they have access to the internal sensitive data such as projects performance audits data; and third, they are able to take a decision about what criteria must be established in order to prioritize the business goals to be achieved. Besides, because this phase ends with communicating the process performance results and where to address the improvement effort, middle management staff and process users are involved in order to increase the need to implement a software process improvement as a strategy toward achieving the business goals identified.

5. *Analyze external best practices*: middle management staff and process users have an important role in this phase because they are the sources toward selecting models and standards to be analyzed. These models and standards are selected depending on the analysis of the practices they perform and they mention by them in the interviews. Besides, senior management staff provides a list of those models and standards in which they are interested.

6. *Implement process improvements*: senior management staff has an important role in this phase because they take decisions on the analysis and priority of the change resistance factor and risks associated with the process improvement implementation and the activities to be implemented in order to prevent them. So, middle management staff has an important role selecting those pilot projects which should use the new processes and giving feedback that is very important to the success in the launching of the new processes and the success histories using them. Finally, at the end of this phase, the process users' staff has an important involvement in launching the improve processes because they have to use these processes and give their opinion of their experience with using them.

## 4 Reinforcing the Prevention of Resistance to Change

To reinforce the prevention of resistance to change that can arise from stakeholders, MIGME-RRC methodology includes activities focused on preventing resistance to change. Table 1 shows a summary of the activities defined as a part of MIGME-RRC focused on preventing the resistance to change.

All activities should be performed together with the stakeholders, so the resistance to change can be prevented or minimized.

## 5 Case Study Results

This section presents an analysis of implementing MIGME-RRC methodology focusing on stakeholders' involvement. Then, this section shows the type of stakeholders that have used either the methodology or the new processes gotten by implementing the methodology, their expertise area, and how they accept and perceive the new processes.

The case study was performed at everis. everis is a multinational consulting firm with factories in Europe and Latin America Region. It offers services which provide solutions to large companies in any sector and it is based on three pillars: innovation, methodologies and efficiency. Since its creation in 1996, it has grown both in revenue and staff in a steady and organic way. Turnover in 2009 where the case study was carried out, everis was over 404M€ and the company employs more than 7,000 people. They have over 1,000 projects opened every month.

**Table 1.** Activities focused on preventing change resistance

| Phase | Activities focused on preventing change resistance |
| --- | --- |
| Identify internal best practices | 1. Present the software process improvement initiative to stakeholders.<br>2. Involve stakeholders in the extraction and validation of tacit knowledge.<br>3. Establish a communication plan.<br>4. Understand the organizational work culture.<br>5. Perform three activities: observe behaviour, describe and classify behaviour and identify related risk focusing on senior management. |
| Assess the organizational performance | 1. Show the process performance with the actual internal best practices.<br>2. Highlight the need to improve the processes to achieve the established business goals.<br>3. Perform three activities: observe behaviour, describe and classify behaviour and identify related risk focusing on senior management |
| Analyze external best practices | 1. Select the models and standards taking into account the business goals and the organizational work culture.<br>2. Establish a multi-model environment as a reference model. |
| Implement process improvements | 1. Analyze change resistance factors and risk factors.<br>2. Select external best practices depending on: impact and adoption difficulty.<br>3. Select early adopters' staff for pilot project.<br>4. Establish efficient communication channels (top-down, bottom-up and lateral).<br>5. Perform continuous support: before, during and after the processes implementation.<br>6. Prepare the material for processes training taking into account the stakeholders identified, their influence in the change and the proper way to address them. |

It is important to mention that before the MIGME-RRC methodology was implemented, it should be validated by the delivery management group of everis. Then, meetings with the delivery management group were performed. The meetings were focus on presenting all methodology content as well as the training material. Performing these meetings allowed to get feedback that was used for improving both methodology activities and the training material.

### 5.1 Implementation

Everis need to develop a project management method as a part of its Corporate Methods methodology (COM), therefore, the new improved processes were grouped as the projects management method they needed. The method obtained was validated and approved by everis' quality and methodology group.

After, as proposed in the last phase of MIGME-RRC methodology, pilot projects were performed. Besides, in order to have better results and reduce risk of performing the new project management method pilots with specific features were selected.

The features of pilot projects were as follows: 1) medium sized projects (no longer than 3 months); 2) a staff of 4-7 people working in the project; 3) budget around €100,000-15,000; and 4) project manager junior profile leader.

After the pilots results were analysed and the COM project management method refined, the new COM project management method was launched through everis' intranet. everis intranet allows the improvement process to be available for everis project managers.

### 5.2 Stakeholders Type

The scope of the experimentation was focused in everis' project management processes because it has a broad impact on the organization business goals. Therefore, this section shows an analysis of the kind of managers included in this case study.

a) *Managers by office*: as mentioned before everis has factories in both Europe and Latin America Region. Then, managers from both regions have used the project management processes. The distribution of the offices in both regions help to ensure that applying the methodology was possible to implement new processes that reflect the way everis works. Figure 2 shows the number of managers for offices.

   As Figure 2 shows, most of the managers who used the new processes where from Spain (Madrid, Barcelona, Murcia, Sevilla) (72%), because the main everis' offices are in this country; around 7% where from countries such as Italy and Portugal group as Rest of Europe; and around 21% of managers where from countries such as Peru, Chile, Argentina and México group as Latin America.

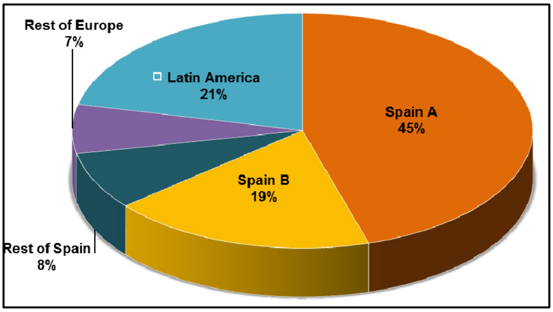



**Fig. 2.** Managers by office's region

b) *Managers' expertise area*: we consider an important data to have identified what kind of managers has used the new processes, this helps to ensure that the new processes are used by managers no mather their expertise area or level. Figure 3 shows managers by expertise.

As Figure 3 shows, most of the managers who have used the new processes (72%) are from Solutions area that offers to generate complex and complete solutions to meet customers' needs based on experience, best practices and other projects; 11% of managers are from Business area that comprises highly-specialized areas of business strategy; 10% of managers are from Outsourcing area that offers the best solutions thus assuring a high added value to achieve both the alignment and evolution of our clients' information systems; the rest of managers are from the follow areas: 4% are managers form Structure, 1% are managers from BPO, 1% are managers from everis initiative and 1% are managers from everis center.

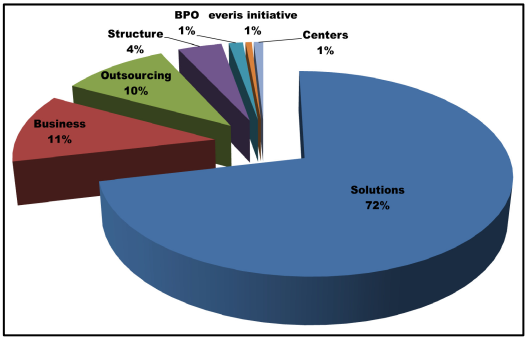



**Fig. 3.** Managers by expertise

### 5.3 Analysis of Results

To understand how stakeholders accept and perceive the new processes, this section is focused on the project management method acceptance analyses. Then, the analyses were focused on project management carried out by managers using COM project management method and how they evaluated the COM method after using to manage their projects.

These analyses are focused on process use and process usefulness to know how well or not users accept the new processes. On the one hand, analysis was done by analysing surveys carried out by managers involved in and used the method to manage their processes. The collected data were from 2009-2010 (FY'09) period. Next, each analysis is showed.

a) *Processes use*: as Figure 4 shows, 48% of project managers uses COM project management method to manage their projects; Around 21% of the managers do not use the COM project management method because the must use the methodology of the customer. About 23% of managers use their experience in order to manage their projects and finally around 8% do not perform any kind of management in order to manage their projects. Therefore, we can say that the new processes contained in the COM project management method have a good acceptation by managers.

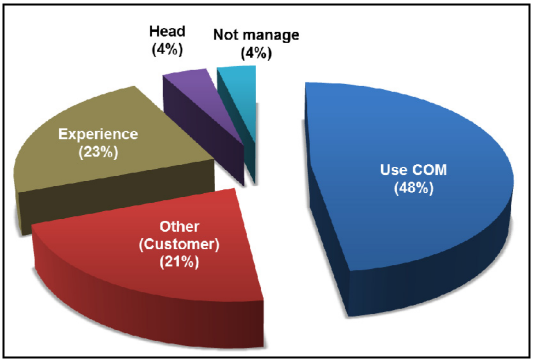



**Fig. 4.** Use of COM by managers

b) *Processes usefulness*: as Figure 5 shows 62% of managers how has used COM project management method to manage its projects perceive the method as usefulness; 16% of managers perceive the method as immature; 12% of managers perceived the method as unprofitable effort; 6% of managers do not know the method; 3% of managers has another reason to not use the method and finally 1% managers perceived that the method do not apply to their projects. After analyzing the percentage of managers who perceived the method as usefulness we can say that the new COM method, which contains the new project management processes, has been perceived as usefulness by managers who have used it to manage their projects.

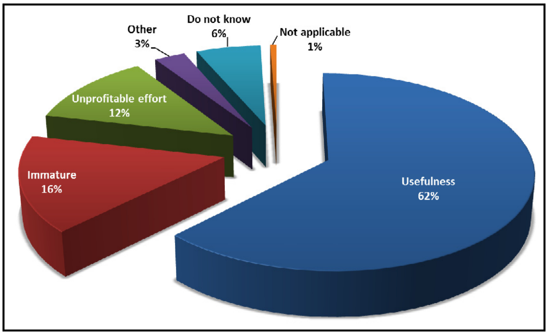



**Fig. 5.** Perception of COM project management method by managers

## 6 Conclusions

Organizations need to create strategic advantages with respect to its competitors in order to be competitive and software process improvement is one of the most widely used strategies to achieve this. However, not all software improvement implementations have the expected results. This research work highlights the involvement of stakeholders in order to implement successful software process improvements. The use of MIGME-RRC methodology allows to involve stakeholder all time throughout the implementation of the software process improvement. As a result, stakeholders have a better acceptation of new processes because they perceived them as their own processes. Besides, they feel as an important element of the process improvement, therefore, they become improvement promoters because they believe in the new process and are convinced of using them in order to be more efficient and to achieve the organizational business goals. As results obtained shows, the new processes had a good acceptation reflected by the number of managers who use them to manage their projects. Then, we can say that the resistance to change of adopting the new processes has been minimized. Finally, it is important to mention that actually COM method is having an evolution according to the actual business goals needs. Besides, we are making an evolution of MIGME-RRC methodology, so it can be easily applied in SMEs and other domains.

**Acknowledgements.** This work is sponsored by Centro de Investigación en Matemáticas (CIMAT), everis Foundation and Polytechnic University of Madrid through the Research Chair in Software Process Improvement for Spain and Latin American Region.

## References

1. Soto-Acosta, P., Martínez-Conesa, I., Colomo-Palacios, R.: An empirical analysis of the relationship between IT training sources and IT value. Information Systems Management 27(3), 274–283 (2010)
2. Gupta, J.N., Sharma, S.K., Hsu, J.: An Overview of Knowledge Management. In: Jennex, M. (ed.) Knowledge Management: Concepts, Methodologies, Tools, and Applications, pp. 1–22. Information Science Reference, Hershey (2008), doi:10.4018/978-1-59904-933-5.ch001
3. Molina, J.L., Marsal, M.: Herramientas de gestión del conocimiento; Gestión del cambio. In: La gestión del conocimiento en las organizaciones, pp. 60–68, 87–94, Colección de Negocios, Empresa y Economía. Libros en red (2002)
4. Turban, E., Aronson, J.E., Liang, T.-P.: Knowledge management. In: Decision Support Systems and Intelligent Systems, pp. 1–23. Perentice Hall, Uppers Saddle River (2005)
5. CMMI working group, The Economics of CMMI®, NDIA System Engineering Division, Software Engineering Institute, 1, 9-21 (2009), http://www.sei.cmu.edu/library/assets/Economics%20of%20CMMI.pdf
6. O'Connor, R., Basri, S.: The Effect of Team Dynamics on Software Development Process Improvement. International Journal of Human Capital and Information Technology Professionals 3(3), 13–26 (2012)

7. Jahn, K., Nielsen, P.A.: A Vertical Approach to Knowledge Management: Codification and Personalization in Software Processes. International Journal of Human Capital and Information Technology Professionals 2(2), 26–36 (2011)
8. Calvo-Manzano, J.A., Cuevas, G., Gómez, G., Mejia, J., Muñoz, M., San Feliu, T.: Methodology for process improvement through basic components and focusing on the resistance to change. Journal of Software Evolution and Process 24(5), 511–523 (2012)
9. Williams, T.: How do organizations learn lessons from Projects—And do they? IEEE Transactions on Engineering Management 55(2), 248–266 (2008)
10. Burke, G.D., Howard, W.H.: Knowledge management and process improvement: a union of two disciplines. CrossTalk, The journal of Defense Software Engineering 18(6), 28 (2005)
11. Muñoz, M., Mejia, J., Calvo-Manzano, J.A., San Feliu, T., Alor, G.: Advantages of using a multi-model environment in software process improvement. In: 2011 IEEE The Electronics, Robotics and Automotive Mechanics Conference (CERMA), Cuernavaca, Morelos, pp. 397–402 (2011), doi:10.1109/CERMA.2011.85

# Agile Maturity Model: A Synopsis as a First Step to Synthesis

Tomas Schweigert[1], Detlef Vohwinkel[1], Morten Korsaa[2], Risto Nevalainen[3], and Miklos Biro[4]

[1] SQS Cologne, Germany
{tomas.schweigert,detlef.vohwinkel}@sqs.com
[2] Delta, Copenhagen, Denmark
MKO@delta.dk
[3] FiSMA (Finnish Software Measurement Association), Helsinki, Finland
risto.nevalainen@fisma.fi
[4] Software Competence Center Hagenberg GmbH, Hagenberg, Austria
miklos.biro@scch.at

**Abstract.** The paper describes the current status of agile maturity models. It shows where such models can be found and it contains a structured top level compilation of the currently available agile maturity models. In the second section, the paper describes an approach to analyse these agile maturity models, extracts their content, maps it to a reference model and then synthesizes the real agile maturity issues. The paper also describes the needs for scientific research in this topic. The paper will not present its own Agile Maturity Model. This will be the task for further research. It intends however to compile current agile maturity model thinking linking it to philosophical issues partly also raised in recent initiatives like the SPI Manifesto, the ECQA PI Manager Certification Scheme and SEMAT.

## 1 Introduction

Long time ago, SPI was based on a set of incomparable models. Eventually, the SPI Community learned that results of appraisals and assessments need to be comparable. Then came agile principles that was often positioned as opposed to the SPI community. But soon discussions between SPI experts and agile evangelists ran out of arguments. This paper tries to help building a bridge between the agile and the SPI world.

It also takes into account that there is a critical view on modern software engineering argued for example by the SEMAT group [30]. The bridge building components in this paper are a collection of recently published Agile Maturity Models and an analysis approach that should help understanding the real nature of the currently available agile maturity models [55].

Currently approximately 40 agile maturity models are published. A subset of these models was subject to a deeper analysis to find out the content and to map them to ISO/IEC 15504 Part 5:2012 [29]. The big question was whether agile maturity really deals with maturity or rather with practice interpretation.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 214–227, 2013.

## 2 Approach and Method

By analysing the agile maturity issue the following key questions were starting point:

- Is an agile maturity model something of relevance
- Does a common accepted agile maturity model exist
- How would such a model map to CMMI
- Does this model fit to common accepted standards as stated in ISO/IEC 15504 Part 2
- How would such a model map to ISO/IEC 15504 Part 5:2012
- If such a model could be found would it more be like a stairway or more like a spider web.

For the 1st question a survey was performed and –with the result that agile maturity is something of interest- published in 2012 [15]. To answer the second and third question an internet search was undertaken and its results were sampled including 40 sources dealing with agile maturity. As completeness criterion somewhat as a jack knife algorithm was used, assuming that –starting with a GOOGLE search and following the links in the found documents as well as checking given references- no more sources will be found when a source is found the 3rd time by following a chain of links. The result contains one cluster of models that somewhat refer to CMMI and other models that follow own maturity approaches. The detailed preliminary analysis is documented in Chapter 3. As there was no common accepted agile maturity model found, questions 4 and 5 became more interesting. The approach and its result are described in Chapter 4. The 1st answer of question 6 is mentioned in chapter 5. As it was clear that this paper did not have the intention to deliver a complete detailed analysis or a complete agile maturity model, chapter 6 deals with some useful follow up actions.

## 3 The Current Discussion on Agile Maturity Models

While it is common opinion, that it is possible for an agile organisation to reach high CMMI maturity levels or high SPICE Capability levels [13], it seems that some agile gurus are not happy with the support offered by CMMI or SPICE. As one of the consequences, we have about 40 different models that call themselves Agile Maturity Models. There are several types of agile maturity models published, mainly in the Internet. There are also some principal thoughts published about agile maturity. The discussion about agile maturity is influenced by ideas of the CMMI Model. See also Schweigert [55]. So, it seems to be adequate to group the published agile maturity models into those which are close to the level structure of CMMI, those which have a level structure at all and those which don't use explicit levels.

### 3.1 A Compilation of Currently Available Agile Maturity Models

Most of the published agile maturity models do have maturity levels or as a minimum a described roadmap to agile maturity. In the table beneath models that explicitly refer to CMMI are included. Even if these models use CMMI as some type of reference even the naming of the levels is mostly different. In the table below the level names of these models are mapped by using the CMMI maturity frame.

| | |
|---|---|
| **CMMI Level 1: Initial.**<br>Agile Maturity Level 1:<br>• Initial [25], [45], [58], [63], [64]<br>• Analysis Ability [6]<br>• Regressive [27]<br>• Iterative and incremental [31] | **CMMI Level 2: Managed.**<br>Agile Maturity Level 2:<br>• Managed [63], [64]<br>• Explored [45]<br>• End to End Traceability [6]<br>• Repeatable [27] |
| **CMMI Level 3: Defined.**<br>Agile Maturity Level 3:<br>• Defined [45], [63], [64]<br>• Organised [25], [58]<br>• Stabilize System Metrics [6]<br>• Consistent [27] | **CMMI Level 4: Quantitatively managed.**<br>Agile Maturity Level 4:<br>• Quantitatively managed [63], [64] [27]<br>• Improved [45]<br>• Disciplined [25], [58]<br>• System thinking and a learning organization [6] |
| **CMMI Level 5: Optimising.**<br>Agile Maturity Level 5:<br>• Sustained [45]<br>• Optimising [27], [63], [64]<br>• Anticipated ROI and the Failure tolerant Organization [6]<br>• Adapting Practices [31] | |

There are also models published that use individual names for maturity levels or stages. The compilation below shows used level 1 to 5 naming.

| | |
|---|---|
| **Level 1 naming.**<br>• Rhetorical stage [4]<br>• Team Level Maturity [48]<br>• Neutral or Chaotic [23], [32]<br>• Emergent Engineering Best Practices [12]<br>• Introductory [44]<br>• Collaborative [8], [57]<br>• Dormant [2]<br>• Stage 1 – No Agile BI [62]<br>• Waterfall [50]<br>• Non-Agile [25], [58]<br>• Core Agile Development[3], [24]<br>• Adherence to Agile Principles (Purity) [1]<br>• Getting Started [53]<br>• Improvising [9] | **Level 2 naming.**<br>• Certified stage [4]<br>• Department Level Maturity [48]<br>• Collaborative [23], [32]<br>• Continuous Practices at Component Level [12]<br>• Learn [20]<br>• Novice [44]<br>• Evolutionary [8], [57]<br>• Speed: Focusing on being expeditious [2]<br>• Stage 2 – Early Adoption [62]<br>• Forming [50]<br>• Minimum [25], [58]<br>• Discipline Agile Delivery[3], [24]<br>• Repeatable Process across the Organization [1]<br>• Scrum at project level [53]<br>• Practicing [9] |
| **Level 3 naming .**<br>• Plausible stage [4]<br>• Business Level Maturity [48]<br>• Operating (Consistent exhibition of competence) [32], [23]<br>• Cross Component Continuous Integration [12]<br>• Leverage [20]<br>• Intermediate [44]<br>• Effective [8], [57]<br>• Reactive: Focusing on acting relative to change from the perspective of the moment rather than a longer timeframe [2]<br>• Stage 3 – Self Service [62]<br>• Agile [50]<br>• Consolidated [25], [58]<br>• Agility at Scale[3], [24]<br>• Scalability –SCRUM of SCRUMS [1] | **Level 4 naming .**<br>• Respectable stage [4]<br>• Project Management Level Maturity [48]<br>• Adaptive (Expertise to adapt to change) [23], [32]<br>• Cross Journey Continuous Integration [12]<br>• Advanced [44]<br>• Adaptive [8], [57]<br>• Responsive: Focusing on acting relative to change from the perspective of the moment balanced with a longer timeframe [2]<br>• Stage 4 – The Lake Effect [62]<br>• Performing [50]<br>• Items on the right [1]<br>• Scrum at Enterprise Level [53]<br>• Governed [9] |

| | |
|---|---|
| • Developed Scrum Capability [53]<br>• Streamlined [9] | |
| **Level 5 naming.**<br>• Measured stage [4]<br>• Management Level Maturity [48]<br>• Innovating (Creative evolution of practice, and spread these practices throughout the organization) [23], [32]<br>• On Demand Just in Time Releases [12]<br>• Optimise [20]<br>• Insane [44]<br>• Ambient [57] [8]<br>• Scaling [50]<br>• Coexistence with non-Agile [1]<br>• Enterprise Transformation [53]<br>• Matured [9] | **Other level naming.**<br>• Regressive (An explicit blame level) [23], [32]<br>• Harmonization with industry standards and frameworks [1]<br>• Support for IT governance and compliance [1]<br>• Scrumming Scrum [53] |

We also see Agile maturity models that do not deal with levels but with features [34] [38], scaling factors [3], [24], Recommendations [61], Management Principles [52], Enablers [56], or Key Questions [40]. The table below shows these aspects in order to deliver a complete picture of agile maturity thinking.

| | |
|---|---|
| **Key Features.**<br>• Build Management and continuous integration [27]<br>• Environments and deployments [27]<br>• Release Management and Compliance [27]<br>• Testing [27]<br>• Data management [27]<br>• Knowledge [34], [49]<br>• Learning [34]<br>• Technology [34]<br>• Change [34]<br>• Mission & Principle [38]<br>• Organization [38] | **Scaling Factors**<br>• Team size[3], [24]<br>• Geographical distribution[3], [24]<br>• Regulatory compliance[3], [24]<br>• Domain complexity[3], [24]<br>• Organizational distribution[3], [24]<br>• Technical complexity [3], [24]<br>• Organizational complexity[3], [24]<br>• Enterprise discipline[3], [24]<br>**Enablers**<br>• Staffing the engineering team correctly [56]. |

| | |
|---|---|
| • Process [38]<br>• Measures [38]<br>• Roles and responsibilities [38]<br>• Policies and standards [38]<br>• Team commitment [35]<br>• Management Support [35]<br>• Shared Responsibility [8], [46]<br>• Build [8], [46]<br>• Requirements [8], [46]<br>• Testing [8], [46]<br>• Responsiveness [8], [46]<br>• Assurance [8], [46]<br>• Simplicity [8], [46]<br>• Configuration Management [8], [46]<br>• Communication [8], [46]<br>Governance [8], [46] | • Assuring Quality is in your team's DNA [56].<br>• Reducing overhead in the release process [56].<br>• Feeding the beast [56].<br>• Managing stakeholder expectations [56].<br>• Continuously learning from your markets [56]. |
| **Recommendations.**<br>• Active user involvement is imperative [61]<br>• The team must be empowered to make decisions [61]<br>• Requirements evolve but the timescale is fixed [61]<br>• Capture requirements at a high level; lightweight & visual [61]<br>• Develop small, incremental releases and iterate [61]<br>• Focus on frequent delivery of products [61]<br>• Complete each feature before moving on to the next [61]<br>• Apply the 80/20 rule [61]<br>• Testing is integrated throughout the project lifecycle – test early and often [61]<br>• A collaborative & cooperative approach between all stakeholders is essential [61]<br>• Use a goal driven approach [5]<br>• Use a set of implementation patterns [21] | **Management Principles.**<br>• Reduce uncertainties by addressing architecturally significant decisions first [52].<br>• Establish an adaptive lifecycle process that accelerates variance reduction [52].<br>• Reduce the amount of custom development through asset reuse and middleware [52].<br>• Instrument the process to measure cost of change, quality trends, and progress trends [52].<br>• Communicate honest progressions and digressions with all stakeholders [52].<br>• Collaborate regularly with stakeholders to renegotiate priorities, scope, resources, and plans [52].<br>• Continuously integrate releases and test usage scenarios with evolving breadth and depth [52].<br>• Establish a collaboration platform that enhances teamwork among potentially distributed teams [52].<br>• Enhance the freedom to change plans, |

| | |
|---|---|
| • Measure the degree on how an organization is able to satisfy its customers, stakeholders, and employees [19] | scope and code releases through automation [52].<br>• Establish a governance model that guarantees creative freedoms to practitioners [52]. |
| **Key Questions.**<br>• Was all the agreed on functionality delivered? [40]<br>• Was a high quality product delivered? [40]<br>• Was the team responsive to new requirements or changes in requirements? [40]<br>• Was there open communication during the project? [40]<br>• Did the stakeholders have proper visibility into progress of the project? [40]<br>• Was there smooth co-ordination between the Agile project and other projects and activities of the organization? [40]<br>• Was there a high level of individual satisfaction? [40]<br>• Was the team productivity high? [40]<br>• Was there a high growth opportunity for team members? [40]<br>• Do you think the success of the project is repeatable? [40] | **Agile improvement procedure.**<br>• Write a simple story that describes the process you followed. Examples are included in the spreadsheet [43]<br>• Rate your process on 12 criteria based on the Agile Alliance principles [43]<br>• Enter weights and view results [43]<br>• Create a list of steps to address deficiencies. Follow the normal agile process to estimate these steps and add to the backlog [43]<br>• Describe the Problem [7]<br>• Visualize the workflow [7]<br>• Identify factors affecting performance [7]<br>• Make process policies explicit [7]<br>• Eliminate Waste [7]<br>• Limit work in progress [7]<br>• Establish an input cadence [7]<br>• Implement changes [7]<br>• Adjust policies [7]<br>• Look for further improvements [7] |

### 3.2 The Discussion of Agile Maturity Models

There were also some discussions or comments on the recently published agile maturity models [15]. There is also the question, if the agile community really needs an agile maturity model [15]. Elssamadisy [21] recommends the use of a set of implementation patterns. He is critical however about an agile maturity model, because he does not see agility as a necessary goal of software development. According to Rothman [51], Agile is also a matter of organizational culture. So agile maturity has to deal with cultural issues like fixed mindset vs. growth mindset, power distance and uncertainty avoidance. All of these models have in mind the approach which was one of the goals of the BOOTSTRAP methodology about 20 years ago: “Change and reorganize the software development and maintenance activities so that the software production as a whole better complies with business needs. [41]. The methods changed, the goals remain.

This result creates however a heavy challenge for the SPI community. Like at the early times of SPICE, lots of models (e.g. ISO 9000, Trillium, TickIT, CMM, BOOTSTRAP, ISO 12207, ISO 15288) were on the market and there was no method to make their results comparable. The need for structured modelling is agreed in the whole IT business. (see e.g TestSPICE for the software testing business [16], [59])

The challenge for SPICE assessors in an agile environment is the heterogeneity of agile implementations. So there is a need to distinguish between malpractice and next practice [39]. To do so, it might be helpful to identify anti patterns to mark malpractices [17] and also implementation patterns [22]. Looking deeper into the agile maturity issue, it becomes clear that improving agile development and management is not only an issue of formal capability or maturity levels but also an issue of values, emotions and culture [42], [10]

## 4 Do Agile Maturity Models Really Measure Agility?

The first idea, when analysing agile maturity models in detail, was to check which capability or maturity levels they are supporting. Trying to map the available agile maturity models to capability indicators of SPICE turned out to be very difficult as many authors of agile maturity models have an undisciplined wording. So there is no clear connotation if a model describing a level, a process attribute, a process, an outcome or an indicator like a base practice. Even a rough analysis shows, that none of the analysed agile maturity models fulfil the requirements of ISO7IEC 15504 Part 2.
So typically, it is very hard (sometimes also impossible) to develop a direct mapping between agile maturity models and SPICE capability levels.

The next step was to atomize a sample of agile maturity models in order to check what they really contain. Using a sample of 12 of the 40 agile maturity models and extracting atomized characteristics out of these models, we synthesized a set of 600 atomized characteristics such as: ID; Author; Type: {Value, Principle, Process, Purpose, Outcome, Practice, Work Product, Process Attribute, Level}; Article; and Content (The extracted statement) Sometimes, it was not possible to define the exact type of the statement. In this case more than one type was chosen This approach was pragmatic. There exist more sophisticated approaches [33], the above one proved however to be effective for the development of a first sample. At the end of this step, 600 prequalified characteristics were ready for further processing.

During a workshop that took place at the EuroSPI 2012 in Vienna, a subset of 250 characteristics of the sample was mapped by the participants to ISO/IEC 15504 Part 5 2012 [29]. The detailed result is the following:

- 1 (0,4%) Characteristic was related to the acquisition process
- 142 (56,8%) Characteristics were found to be related to the content of system engineering processes from which

- 70 (28%) Characteristics were found to be related to software life cycle processes
- 12 (4,8%) Characteristics were found to be related to capability levels 2, 3 and 5
- 5 (2%) Characteristics were found to be related to a release management process
- 21 (8,4%) Characteristics were not mappable at all. May be they are the real agile nucleus.

A first rough interpretation of these findings is that agile maturity does not deal with capability in the classic sense (even if there is a substantial level of support for this view) but deals rather with process/practice implementation in an agile style. It is evident that this interpretation should be considered as a 1st hypothesis rather than a proven academic truth.

## 5 General Interpretation

Currently, there is no commonly accepted model for agile maturity. There are lots of level descriptions and also a lot of supporting characteristics. As shown, it is a challenge to develop a synthesis of all these agile maturity models that covers all aspects of agile maturity, gains the acceptance of the SPICE and the agile community and is easy and flexible to use.

To get started, we need to revisit the initial sources of agile and process improvement: The Agile Manifesto [11], the SPI manifesto [36] [37] [47] and the SPI body of knowledge as currently described in ISO/IEC 15504 Part 4 [28]. They are similar regarding the idea that business value is the main driver.

The situation was shown to be analogous to mathematics in the sense that different models applied in given theories can lead to a more or less elegant and extensible approach to the same or even revolutionarily new concepts [15]. In our case, we also have many different approaches and statements that are claimed to be effective in leading to the achievement of the same business goals. The key analogy with mathematics is that there are different ways (CMMI, SPICE, Agile maturity models as discussed in our survey) to get to the conclusion that a method is effective in leading to the achievement of business objectives. These ways are just as different in their elegance (esthetics) and usability as there are very different approaches in mathematics to the same concepts. It is also true that elegance (esthetics) depends on subjective taste, while effectiveness and usability may be objectively measurable. By consequent, there will always be differences in the opinions regarding the elegance of approaches while at the same time all parties may measurably support their claims regarding the effectiveness or usability of their approach.

We can also recur to one of the key principles applied by SEMAT as well: Separation of Concerns. Subjective concerns like elegance (esthetics) for example should be separated from objectively measurable concerns like effectiveness and usability, especially if the objective measurements are not differentiating enough.

Returning to the need for a commonly accepted model for agile maturity, we can claim that classical capability thinking is about raising efficiency and reducing risk [28]. It is one component of the way to performance excellence [60]. This type of thinking requires a simple to use one dimensional scale [26]. However, as we see from the sample of agile maturity models and the workshop results, agile maturity doesn't have a one dimensional scale but can be considered as a spider web. Looking for the potential axes of this spider web, we can observe that one of the axes is definitely traditional capability. Looking for other axis, we can recur to the workshop results.

- 30% are related to organizational issues
- 18% are related to software implementation issues
- 16,4% are related to project issues
- 10% are related to technical system implementation issues.

Keeping in mind that agility means the creation of business value by frequently producing shippable software [11], we have to consider technical quality as also a component of agile maturity. It is also a fact that process and product need to be reviewed together to get a realistic view on maturity [54]. So, it is arguable to propose the following axes for an agile maturity spider web:

- Process Capability
- Technical quality (may be measured in technical debt)
- Organizational support for agile development
- Agile teaming
- Agile culture

## 6 Conclusion and Further Work

Currently we cannot find a commonly accepted Agile Maturity Model. Similar to the early times of SPICE and BOOTSTRAP [41], there are lots of incomparable models on the market. Even if there is a common acceptance of the need for an agile maturity model [15] there is a question if the current situation might lead to a new common accepted methodology of systems and software development or just to the death of agile by applying malpractices [14]. In order to create a commonly accepted Agile Maturity Model, intensive research has to be performed. Consideration should be given to the approach proposed by Clarke and O'Connor [18] to extract an acceptable first draft out of available information. From a practical perspective, answering the following questions requires further research:

- What is the measurable level of acceptance of the currently available agile maturity models?
- What are the most important characteristics of common accepted models from an agile and from an SPI perspective?
- What is the factual – emotional arguments neglected – relationship between agile and SPI approaches? Is one a mean for the others goal or is it the other way around? Or both or neither?
- What might be an acceptable synthesis of the available models in terms of content and structure?
- What is an acceptable approach for assessor training and assessment performance [36], [37].

## References


1. Aiello, B.: Agile Process Maturity - Introducing the Framework, Configuration Management Best Practices, Web publishing, http://cmbestpractices.com/articles/agile-process-maturity-framework
2. Alhir, S.S.: Maturity Models: Leanness, Agility, Competitiveness, and Collaboration, Web Publishing (2009), http://salhir.wordpress.com/2009/06/26/maturity-models-leanness-agility-competitiveness-and-collaboration/
3. Ambler, S.: The Agile Scaling Model (ASM): Adapting Agile Methods for Complex Environments. IBM Rational White Paper, Web Publishing ftp://ftp.software.ibm.com/common/si/sa/wh/n/raw14204usen/RAW14204USEN.PDF
4. Ambler, S.: The Agile Maturity Model (AMM), Web publishing http://drdobbs.com/architecture-and-design/224201005, http://drdobbs.com/tools/224201005
5. Ambler, S.: Disciplined Agilists Take a Goal-Driven Approach, web publishing, http://disciplinedagiledelivery.wordpress.com/2013/01/21/disciplined-agilists-take-a-goal-driven-approach/
6. Anderson, D.J.: Agile Management for Software Engineering, Applying the theory of constraints for business results. Prentice Hall (2004), web publishing http://books.google.co.uk/books?id=hawMF31KCRsC&pg=PA105&lpg=PA105&dq=anderson+agile+maturity+model&source=bl&ots=Zj7aDyaLUT&sig=EV6F3Y1PbfuJHY4ijA5blIOY-dQ&hl=en&ei=TKT5SfqWEouJ_QahuIS2BA&sa=X&oi=book_result&ct=result&resnum=1#v=onepage&q&f=false
7. Anderson, D.J.: KANBAN, Successful Evolutionary Change for Your Technology Business
8. Banerjee, U.: Agile Maturity Model – Three Different Approaches, web publishing http://cloudcomputing.ulitzer.com/node/2081511, http://ca.sys-con.com/node/2081511, http://setandbma.wordpress.com/tag/agile-maturity-model/, http://setandbma.wordpress.com/2011/11/30/agile-maturity-model/
9. Bavani, R.: Distributed Agile: The Maturity Curve, web publishing http://www.blogs.mindtree.com/distributed-agile-maturity-curve-%E2%80%93-part-1
10. Beck, K.: Extreme Programming – Das Manifest. Addison Wesley, München (2000)
11. Beck, K., et al.: Agile Manifesto (2001), http://agilemanifesto.org/
12. Benefield, R.: Seven Dimensions of Agile Maturity in the Global Enterprise: A Case Study. In: Proceedings of the 43rd Hawaii International Conference on System Sciences – 2010 (2010), Web Publishing http://www.agileleantraining.com/download/HICSS43-Benefield.pdf
13. Bianco, C.: Agile and SPICE Capability Levels. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 181–185. Springer, Heidelberg (2011)
14. Biro, M.: The Software Process Improvement Hype Cycle. Invited contribution to the Monograph: Experiences and Advances in Software Quality (Guest editors: Dalcher, D., Fernández-Sanz, L.) CEPIS UPGRADE, vol. X(5), pp. 14–20, https://vm.mtmt.hu/download/1290381.pdf, http://www.cepis.org/files/cepisupgrade/issue%20V-2009-fullissue.pdf (accessed: February 21, 2012)

15. Biro, M., Korsaa, M., Nevalainen, R., Vohwinkel, D., Schweigert, T.: Agile Maturity Model – Go back to the Start of the Cycle. In: Industrial Proceedings of the 2012 EuroSPI Conference, pp. 5.9–5.30 (2012)
16. Blaschke, M., Philipp, M., Schweigert, T.: The Test SPICE Approach – Test process assessments follow in the footsteps of software process assessments. Testing Experience, S.56
17. Brown, W.J., Malveau, R.C., McCormick III, H.W., Mowbray, T.J.: Anti Patterns. mitp, Bonn (2004)
18. Clarke, P., O'Connor, R.: The situational factors that affect the software development process: Towards a comprehensive reference framework. Information and Software Technology 54(5), 433–447 (2012)
19. Derby, E.: Achieving Agility: Means to an End, or End in Itself, Web Publishing http://www.estherderby.com/2010/06/achieving-agility-means-to-an-end-or-end-in-itself-2.html
20. Druckman, A.: Agile Transformation Strategy White Paper, http://www.agilejournal.com/pdf/CollabNet_Agile_Transformation_Strategy.pdf
21. Elssamadisy, A.: Does the Agile Community Need a Maturity Model? Web Publishing http://www.infoq.com/news/2007/10/agile_maturity_model
22. Elssamadisy, A.: Agile Adoption Patterns A Roadmap to Organizational Success. Addison Wesley (2008)
23. Gujral, R., Jayaraj, S.: The Agile Maturity Model, http://whattodowearelikethatonly.blogspot.com/2008/08/agile-maturity-model.html, http://agilephilly.ning.com/page/agile-maturity-model
24. Heydt, M.: Agile Process Maturity Model (2009), Web Publishing http://42spikes.com/post/Agile-Process-Maturity-Model.aspx
25. Hibbs, C.: Towards an Agile Process Maturity Model, original source lost, last source web citation (Download), http://www.guj.com.br/java/94510-modelo-de-maturidade-para-metodologias-ageis
26. Hörmann, K., Dittmann, L., Hindel, B., Müller, M.: SPICE in der Praxis. dpunkt, Heidelberg (2006)
27. Humble, J., Russell, R.: The Agile Maturity Model Applied to Building and Releasing Software, ThoughtWorks White Paper, Web Publishing http://www.thoughtworks-studios.com/sites/default/files/resource/the_agile_maturity_model.pdf
28. ISO/IEC 15504 Part 4:2004 Guidance on use for process improvement and process capability determination
29. ISO/IEC 15504 Part 5:2012: An exemplar Process Assessment Model
30. Jacobson, I., Huang, S., Kajko–Mattsson, M., McMahon, P., Seymour, E.: Semat – Three Year Vision, Web Publishing http://www.semat.org/pub/Main/SematDocuments/Semat_-_Three_Year_Vision13Jan12.pdf (accessed: February 21, 2012)
31. Janus, A.: Konzepte für Agile Qualitätssicherung und -bewertung in Wartungs- und Weiterentwicklungsprojekten. Shaker-Verlag, Aachen (2013)
32. Jayaraj, S.: The Agile Maturity Model, Web Publishing http://whattodowearelikethatonly.blogspot.com/2008/08/agile-maturity-model.html

33. Jeners, S., Lichter, H., Dragomir, A.: Towards an Integration of Multiple Process Improvement Reference Models Based on Automated Concept Extraction. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 205–216. Springer, Heidelberg (2012)
34. Kelly, A.: Changing Software Development: Learning to Become Agile. Wiley (2008)
35. King, J.: A mathematical formula to make agile work, Web Publishing http://kingsinsight.com/2011/07/14/a-mathematical-formula-to-make-agile-work/#more-529
36. Korsaa, M., Biro, M., Messnarz, R., Johansen, J., Vohwinkel, D., Nevalainen, R., Schweigert, T.: The SPI manifesto and the ECQA SPI manager certification scheme. Journal of Software Maintenance and Evolution: Research and Practice (published online: August 25, 2010), doi: 10.1002/smr.502
37. Korsaa, M., Johansen, J., Schweigert, T., Vohwinkel, D., Messnarz, R., Nevalainen, R., Biro, M.: The people aspects in modern process improvement (published online: November 2, 2011), doi:10.1002/smr.570
38. Kroll, P., Ambler, S.: Applying lean thinking to the governance of software development. IBM White Paper, Web publishing http://public.dhe.ibm.com/common/ssi/ecm/en/raw14000usen/RAW14000USEN.PDF
39. Kruse, P.: Next practice. Gabal, Offenbach (2004)
40. Kuruppath, K.: Maturing Agile Processes to Deliver Better Value. Razorfish White Paper (2009)
41. Kuvaja, P., Simila, J., Krzanik, L., Bicego, A., Saukkonen, S., Koch, G.: Software Process Assessment & Improvement, The BOOTSTRAP Approach. Blackwell, Oxford (1994)
42. Lundak, J.: Agile Prozesse – Fallstricke erkennen und vermeiden. entwickler.press, Frankfurt (2009)
43. Malik, N.: Simple Lifecycle Agility Maturity Model, Web Publishing http://blogs.msdn.com/b/nickmalik/archive/2007/06/23/simple-lifecycle-agility-maturity-model.aspx
44. Minick, E., Fredrick, J.: Enterprise Continuous Integration Maturity Model. Urbancode Whitepaper, Web publishing http://www.agilejournal.com/pdf/Enterprise_Continuous_Integration_Maturity_Model.pdf
45. Patel, C., Ramachandran, M.: Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices. Int. J. of Software Engineering, IJSE 2(1), 3–28 (2009), Web Publishing http://www.ijse.org.eg/Content/Vol2/No1/Vol2_No1_1.pdf
46. Pettit, R.: An "Agile Maturity Model?" (2007), Web publishing http://www.agilejournal.com/articles/columns/the-agile-manager/52-an-qagile-maturity-modelq
47. Pries-Heje, J., Johansen, J., et al. (eds.): SPI Manifesto, Alcala, Version A.1.2.2010 (2010), http://www.eurospi.net/images/documents/spi_manifesto.pdf (accessed: February 21, 2012)
48. Proulx, M.: Yet Another Agile Maturity Model (AMM) – The 5 Levels of Maturity (2010), Web publishing http://analytical-mind.com/2010/07/12/yet-another-agile-maturity-model-the-5-levels-of-maturity/
49. Rohrbeck, R.: Corporate Foresight. Physica Verlag, Heidelberg (2011)
50. Ronen, S.: Agile Testing Maturity Model - practical view, Web publishing http://www.slideshare.net/shirlyronen/agile-testing-maturity-model-practical-view

51. Rothman, J.: Where is Agile Going? (2011), Web publishing http://www.slideshare.net/johannarothman/where-is-agile-going-withculture
52. Royce, W.: Improving Software Economics, Top 10 Principles of Achieving Agility at Scale. IBM Rational White Paper, Web Publishing http://www.agilejournal.com/pdf/RAW14148USEN.pdf
53. Schwaber, K., Sutherland, J.: Software in 30 Days. Wileys, Hoboken (2012)
54. Schweigert, T.: Crossing the Border between Process and Product, the Requirements Catalogue Maturity Model. In: Proceedings of the EuroSPI 2008 Conference, pp. 3.11–3.23 (2008)
55. Schweigert, T., Nevalainen, R., Vohwinkel, D., Korsaa, M., Biro, M.: Agile Maturity Model: Oxymoron or the Next Level of Understanding. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 289–294. Springer, Heidelberg (2012)
56. Sehlhorst, S., Blain, T.: Agile Maturity Model – What's Next? Web publishing http://tynerblain.com/blog/2009/06/30/agile-maturity-model/
57. Sidky, A., Arthur, J.: A Disciplined Approach to Adopting Agile Practices: The Agile Adoption Framework, web publishing http://arxiv.org/ftp/arxiv/papers/0704/0704.1294.pdf
58. Sims, C., Ambler, S.: Revisits Agile Process Maturity Models, Web Publishing http://www.infoq.com/news/2009/04/Agile-Maturity-Models
59. Steiner, M., Blaschke, M., Philipp, M., Schweigert, T.: Make test process assessment similar to software process assessment—the Test SPICE approach. Journal of Software Maintenance and Evolution: Research and Practice (article first published online: August 25, 2010), doi: 10.1002/smr.507
60. Wagner, K.W., Patzak, G.: Performance Excellence, Der Praxisleitfaden zum effektiven Prozessmanagement. Hanser, München (2007)
61. Waters, K.: 10 Key Principles of Agile Software Development, web publishing http://www.agilejournal.com/resources/directory?sobi2Task=sobi2Details&catid=10&sobi2Id=14
62. Woods, D.: An Agile BI Maturity Model, Forbes (2011), web publishing http://www.forbes.com/sites/danwoods/2011/10/26/an-agile-bi-maturity-model/3/
63. Yin, A.P.G.: Scrum Maturity Model, Dissertacao para obtencao do Grau de Mestre em Engenharia Informática e de Computadores
64. Yin, A., Figueiredo, S., da Silva, M.M.: Scrum Maturity Model. In: Proceedings of the ICSEA 2011: The Sixth International Conference on Software Engineering Advances, pp. S. 20–S. 29 (2011)

# An Agile Management Process Group for TestSPICE®

## How to Assess and Improve Agile Management

Tomas Schweigert[1], Mohsen Ekssir-Monfared[2], and Magda Ofner[3]

[1] SQS Software Quality Systems AG, Stollwerck Str. 11, 51149 Köln
tomas.schweigert@sqs.de
[2] BDC EDV-Consulting GmbH, Gredlerstrasse 4/2, 1020-Wien
mohsen.ekssir@bdc.at
[3] ANECON Software Design und Beratung G.m.b.H., Alser Straße 4 / Hof 1, A-1090 WIEN
magda.ofner@anecon.com

**Abstract.** The paper presents an outline for an Agile Management Process Group. It deals with the well-known question to what extent agile management helps to reach SPICE capability levels and also discusses to which extent the features of SPICE help to improve agility. The paper also deals with the question of agile maturity in general, agile maturity in the Certified Agile Tester (CAT) syllabus and agile anti patterns. Based on this analysis the paper presents the proposal for an Agile Management Process Group in TestSPICE®

**Keywords:** SPICE, ISO 15504, TestSPICE, SCRUM, XP, CAT, Certified agile tester, Maturity, Capability, CMMI, Agile Maturity Model.

## 1 Introduction

When the question comes up, how formal PRM and PAM on the one site and agile methods support each other, several studies are proving that agile methods support the achievement of CMMI or SPICE levels [1] [3] [6] [8] [18] [19] [23]. On the other hand no source is available that explains the benefit of SPICE to the agile community. So it is not surprising that the agile community started internal discussions about agile maturity which led to roundabout 40 published agile maturity models. So we are far away from an agile PAM [4] [25].

Looking for a valid source than can serve as a basis for the assessment of agile management that is aligned with the principles of the Agile Manifesto [15] it is likely to check pitfalls and anti-patterns [31] that are communicated inside the agile community. Looking at the result it can't be something like just another maturity model. If the challenge is to improve the SPICE support for agile, agile core processes have to be identified, described in terms of purpose and outcomes and enhanced by useful indicators. Currently the TestSPICE SIG [29] is discussing a proposal for such an agile management process if it has the right content and if it should be integrated in the TestSPICE® Process Reference Model or if as a 1st step it should be mentioned in an appendix of the TestSPICE® PAM [30] which style is also used by ISO/IEC

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 228–236, 2013.

15504 Part 5:2012 [13]. As reported [5] [16] [24] [26] the TestSPICE® PAM [30] evolved since the founding of the TestSPICE SIG, but in the beginning focusing more on the technical [24] and test center related processes.

## 2 Synergies between Agile Management and SPICE

Setting the stage for an Agile Management Process Group a short revisit of the relationship between SPICE and Agile seems to be useful:

### 2.1 Contributions of Agile Development Life Cycles as XP or SCRUM to SPICE Capability Levels

It is proven from a broad CMMI experience, that the proper implementation of agile practices helps to gain CMMI maturity levels [6] [12] [13] [16] [23]. Having this as a common understanding of agile CMMI synergy, the same synergy applies for SPICE as most of CMMI best practices are also included in the SPICE PAM [30].

This leads to the intermediate result that –from this perspective- synergy is probable [3] [7].

### 2.2 Contributions of SPICE Assessments to Agile Improvement

Wanting to effectively utilize SPICE in an agile environment, we have to reflect the whole improvement life cycle [22]. ISO/IEC 15504 Part 4:2004 [11], describes an improvement life cycle that does not start with an assessment but with the analysis of business drivers the same principle applys in the SPI Manifesto [20] and the Improvability approach [21]. It is not likely that an organization has on perception of business drivers that motivates process improvement and another perception that motivates the introduction of agile practices. So if most likely the business drivers for process improvement and agility are the same, the outcome of a business driver analysis is meaningful for process improvement as well as for decisions about agility.

Having a business driver analysis as a starting point, the target capability profile will be the next deliverable. This capability profile shows a what: processes and capability levels. But the capability profile shows also a first agile picture covering reuse, knowledge management, training, improvement and some other key processes. A mature agile organization needs these processes at a minimum of Level 3.

So if agile management is the management standard of an organization the PA 3.2 assessment clearly delivers an insight if agile management practices are really established in the organization. Even if it is currently difficult to achieve commitment what agile maturity really is, in a standard assessment report strength and weaknesses for each process are mentioned. The strength of a process are clearly related to business needs (as well are the weaknesses). If a business environment is likely to produce frequent changes the management style has to deal with this frame condition. If not a weakness has to be reported.

So even if the target process profile does not require processes on capability level 3 we can see that discussing strength and weaknesses allows giving feedback on the implementation of agile processes. So a SPICE Assessment is able to deliver a feedback if an organization needs agile management and if the need is proven the feedback shows if the organization obtained an agile management style.

## 3 Sources of Agile Management Processes

Looking for sources of agile management processes one option is to analyze how agile gurus tend to improve organizations or to use common accepted pitfalls and anti-patterns.

The 1st idea when analysing agile maturity models in detail is to check if they contain a nucleus for a Process Reference Model (PRM) or enhancements of the Measurement Framework which will be part of the Process Assessment Model (PAM). Trying to map the available agile maturity models to process performance or process capability indicators of SPICE requires that agile maturity models are described in the same rigorous wording style as the SPICE PAM which is also required by ISO/IEC 15504 Part 2 and Part 3 [9] [10]. If there is no clear connotation if a model describes a level, a process attribute, a process, an outcome or an indicator like a base practice, the mapping requires a deep scientific analysis of these models based on atomized statements an mapping these statements to capability characteristics like levels, attributes, process, outcomes, practices or work products.

During a workshop that took place at the EuroSPI 2012 in Vienna a set of 250 atomized statements was mapped by the participants to ISO/IEC 15504 Part 5 2012. The result was:

- 57% Characteristics were found related to the content of system engineering processes
- 28% Characteristics were found related to software life cycle processes
- 5% Characteristics were found related to capability levels 2, 3 and 5
- 2% Characteristics were found related to a release management process and
- 8% Characteristics were mot mapable at all. May be they are the real agile nucleus

So it is not likely that agile maturity models are a reasonable source for an Agile Management Process Group. Looking at the CAT Syllabus [28] the result is similar. The Syllabus addresses core testing processes as Test Design and Test Execution.

## 4 The TestSPICE® Answer to the Agile Obstacle

The current proposal for an Agile Management Process Group focusses on the following topics: The handling of Backlogs; Impediments; Technical debt; WIP Limits; Knowledge debt; and Organizational Capacity Management.

The handled processes are the most used activities in agile projects. These are abstract and generic activities, which could be handled in different agile software development methods, like SCRUM, XP KANBAN or LEAN. The Organisational Capacity Management Process was introduced due to the fact that implementation and the improvement of agile processes need –as every improvement [2], [17]- an organizational framework [31].

## 4.1 AMP.1 Backlog Management

| | | |
|---|---|---|
| Process ID | AMP.1 | |
| Process name | Backlog management | |
| Process purpose | The purpose of the Backlog Management process is to make sure that relevant items like requirements are collected, described, properly stored, prioritized estimated and solutions are delivered. | |
| Process outcomes | Outcome 1 | The backlog is visible at a defined storage place |
| | Outcome 2 | The items in the backlog are agreed between the relevant stakeholders |
| | Outcome 3 | The backlog contains the complete set of items<br>Note: no shadow backlogs are used |
| | Outcome 4 | The items in the backlog are prioritized<br>Note: the classification of prioritizing shall be clear. It shall be defined in which manner the backlogs are to be prioritized (e. g. regarding business value, complexity, etc.) |
| | Outcome 5 | The items in the backlog are estimated |
| | Outcome 6 | The assignment of items to an iteration (or increment) is visible |
| | Outcome 7 | Delivery of items are visible |

## 4.2 AMP.2 Impediment Management

| | | |
|---|---|---|
| Process ID | AMP.2 | |
| Process name | Impediment Management | |
| Process purpose | The purpose of Impediment Management process is to improve the performance of a development or test team by constantly and consequently removing impediments that hamper the progress or the performance of an agile team. | |
| Process outcomes | Outcome 1 | Impediments are unique identified |
| | Outcome 2 | The impact of an impediment is analyzed |
| | Outcome 3 | The root cause of an impediment is analyzed |

| | | |
|---|---|---|
| | Outcome 4 | The potential solutions of the impediment are derived, prioritized and agreed with the team |
| | Outcome 5 | The stakeholders need to implement a solution are identified |
| | Outcome 6 | A solution of the impediment is agreed with the relevant stakeholders |
| | Outcome 7 | The impediment is implemented |
| | Outcome 8 | The impediment is tracked to closure |

### 4.3 AMP.3 Service Class and WIP Limit Management

| | | |
|---|---|---|
| Process ID | AMP.3 | |
| Process name | Service Class and WIP Limit management | |
| Process purpose | The purpose of the Service Class and WIP (work in progress) Limit Management process is to identify and structure the workload of a team and to identify actual as well as potential bottlenecks in the development process. | |
| Process outcomes | Outcome 1 | The services of a team are identified |
| | Outcome 2 | The services of a team are analyzed and structured |
| | Outcome 3 | The capacity of the team is split across agreed service classes |
| | Outcome 4 | WIP limit for each service class is defined |
| | Outcome 5 | The handling of emergency requests is defined |
| | Outcome 6 | Queuing mechanisms for the service classes of a team are defined |
| | Outcome 7 | Requests, work in progress and delivered items are properly visualized |
| | Outcome 8 | Utilization of team capacity is readjusted as needed |

### 4.4 AMP.4 Technical Debt Management

| | | |
|---|---|---|
| Process ID | AMP.4 | |
| Process name | Technical debt Management | |
| Process purpose | The purpose of the Technical debt Management process is to identify technical problems that create a business risk and to implement solutions to deal with technical debt. | |
| Process outcomes | Outcome 1 | Technical debts are identified |
| | Outcome 2 | The impact of technical debts are analyzed |
| | Outcome 3 | Root causes for technical debts are analyzed |

| | | |
|---|---|---|
| | Outcome 4 | Corrective actions are identified and prioritized |
| | Outcome 5 | Preventive actions are identified and prioritized |
| | Outcome 6 | Technical debts and their impacts as well as corrective and preventive actions and their benefits are effectively communicated |
| | Outcome 7 | Corrective and preventive actions and their impacts on team capacity are agreed with the relevant stakeholders (e.g. development team, product owner etc.) |
| | Outcome 8 | Corrective actions are implemented as agreed |
| | Outcome 9 | Preventive actions are implemented as agreed |
| | Outcome 10 | Technical debts are tracked to closure |

## 4.5 AMP.5 Knowledge Debt Management

| | | |
|---|---|---|
| Process ID | AMP.5 | |
| Process name | Knowledge debt Management | |
| Process purpose | The purpose of the knowledge debt Management process is to assure that "just enough documentation" is produced and that relevant knowledge is properly shared. | |
| Process outcomes | Outcome 1 | The needs for documentation and knowledge sharing are identified |
| | Outcome 2 | Gaps are identified |
| | Outcome 3 | The impact of the gaps are analyzed |
| | Outcome 4 | Potential knowledge bottlenecks are identified |
| | Outcome 5 | The need for corrective actions is identified |
| | Outcome 6 | Corrective actions are agreed with all stakeholders |
| | Outcome 7 | Gaps are tracked to closure |

## 4.6 AMP.6 Definition of Done (DoD) Management

| | |
|---|---|
| Process ID | AMP.6 |
| Process name | Definition of Done (DoD) Management |
| Process purpose | The purpose of the Definition of Done Management process is to assure that the delivered functionalities and activities are really ready to ship. |

| | | |
|---|---|---|
| Process outcomes | Outcome 1 | It is defined for which following level the DoD is to be developed:<br>DoD for a functionality or an activity (user story or a product backlog)<br>DoD for a sprint (a list of collected features for the sprint)<br>DoD for a release version |
| | Outcome 2 | The team member and stakeholder have already provided the relevant DoD |
| | Outcome 3 | The list of DoD for the relevant level is developed and communicated in team and with stakeholder |
| | Outcome 4 | Criteria for Done are analyzed, collected and communicated in team |
| | Outcome 5 | Effort caused by each DoD is estimated |
| | Outcome 6 | The DoD are agreed between the team the individuals and the stakeholders |
| | Outcome 7 | The DoD are integrated in the delivery and capacity planning |

## 4.7 AMP.7 Organizational Capacity Management

| | | |
|---|---|---|
| Process ID | AMP.7 | |
| Process name | Organizational Capacity Management | |
| Process purpose | The purpose of the Organizational Capacity Management process is to align the available team capacity with the delivery needs of the organization in order to avoid micro management. | |
| Process outcomes | Outcome 1 | The organizational need for delivered software is analyzed |
| | Outcome 2 | Global priorities are set by senior management |
| | Outcome 3 | The capacity of the available teams is analyzed based on their ability to deliver |
| | Outcome 4 | Gaps between the organizational need and the ability to deliver are identified |
| | Outcome 5 | Strategies to deal with these gaps are developed, agreed, communicated and implemented |
| | Outcome 6 | Micro management attempts are identified and worked upon |
| | Outcome 7 | Team utilization and frequent delivery is constantly monitored |

## 5 Conclusion and Further Work

The AMP process group is intended as a first step on a way to transform TestSPICE® in a way that a TestSPICE® helps agile as well as classical organizations to improve. The Authors will not only propose the processes but also indictors (base practices and work products). Even if this approach is a first answer, we need also investigation how the team focus of agile can be translated to a process without needing an own team assessment model [27].

## References

1. Angermo Ringstad, M., Dingsoyr, T., Brede Moe, N.: Agile Process Improvement, Diagnosis and Planning to improve Teamwork. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 167–178. Springer, Heidelberg (2011)
2. Bayona, S., Calvo-Manzano, J.A., San Feliu, T.: Critical Success Factors in Software Process Improvement: A Systematic Review. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 1–12. Springer, Heidelberg (2012)
3. Bianco, C.: Agile and SPICE capability levels. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 181–185. Springer, Heidelberg (2011)
4. Biro, M., Korsaa, M., Nevalainen, R., Vohwinkel, D., Schweigert, T.: Agile Maturity Model, Go back to the Start of the Cycle. In: EuroSPI 2012 Industrial Proceedings, Delta (2012)
5. Blaschke, M., et al.: The TestSPICE approach, Test Process Assessments follow in the footsteps of software process assessments. Testing Experience, 56 (December 2009)
6. Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, S.: CMMI or Agile, Why not Embrace Both? Technical Note SEI (2008)
7. Certified agile Tester, `http://www.agile-tester.org`
8. Irrazabal, E., Vásquez, F., Díaz, R., Garzás, J.: Applying ISO/IEC 12207:2008 with SCRUM and Agile Methods. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 169–180. Springer, Heidelberg (2011)
9. ISO/IEC 15504 Part 2:2003 Performing an assessment
10. ISO/IEC 15504 Part 3:2004 Guidance on performing an assessment
11. ISO/IEC 15504 Part 4:2004 Guidance on use for process improvement and process capability determination
12. ISO/IEC 15504 Part 5:2006: An exemplar Process Assessment Model
13. ISO/IEC 15504 Part 5:2012: An exemplar Process Assessment Model
14. ISO/IEC 29119 Part 2: Draft International Standard 2012: Test Process
15. Beck, K., et al.: Agile Manifesto (2001), `http://agilemanifesto.org/`
16. Knüvener, C.: TestSPICE – SPICE für Testprozesse. SQ-Magazin 17, 26–27 (2010)
17. Lepmets, M., Ras, E., Renault, A.: Organizational Support for Process Improvement – Results of an international Survey. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 133–144. Springer, Heidelberg (2011)
18. Mc Mahon, P.: Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement. Addison-Wesley Professional, Boston (2010)

19. Pikkarainen, M.: Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices, Espoo, p. 119. VTT Publications 695 (2008), http://www.vtt.fi/inf/pdf/publications/2008/P695.pdf
20. Pries-Heje, J., Johansen, J., et al. (eds.): SPI Manifesto, Alcala (2010)
21. Pries-Heje, J., Johansen, J.: Improve IT, Horsholm, Delta (2007)
22. Rout, T.: Toward a Body of Knowledge for Process Assessment. In: Rout, T., Lami, G., Fabbrini, F. (eds.) SPICE 2010 Proceedings. Edizioni ETS, Pisa (2010)
23. Russwurm, W.: Hidden Treasure: The Implementation of CMMI Practices by Agile Methods. In: Procedures of the 2010 SEPG Conference (2010), http://www.sei.cmu.edu/library/assets/presentations/2201_Russwurm.pdf
24. Schweigert, T., Nehfort, A.: Technical Issues in Test Process Assessment and their current and future Handling in Test SPICE. In: EuroSPI Industrial Proceedings, Delta (2011)
25. Schweigert, T., Nevalainen, R., Vohwinkel, D., Korsaa, M., Biro, M.: Agile Maturity Model: Oxymoron or the Next Level of Understanding. In: Mas, A., Mesquida, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2012. CCIS, vol. 290, pp. 289–294. Springer, Heidelberg (2012)
26. Steiner, M. et al.: Make test process assessment similar to software process assessment - the TestSPICE approach. Journal of Software Maintenance and Evolution (2010), published online 2010 at Wiley online library, http://wileyonlinelibrary.com, doi:10.1002/SMR 507
27. Stettina, C.J., Heijstek, W.: Five Agile Factors: Helping Self-management to Self-reflect. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 84–96. Springer, Heidelberg (2011)
28. The CAT SIG, Certified Agile Tester Manual, Version 2.1, Berlin, ISQI (2010)
29. The TestSPICE PAM Version 2012 (2012), http://www.testspice.info
30. The TestSPICE SIG, http://www.testspice.info
31. Mahmood, F.: Agile Adoption Mistakes You Must Avoid, London (2012)

# Evidence of Agile Adoption in Software Organizations: An Empirical Survey

Efi Papatheocharous[1,2], and Andreas S. Andreou[3]

[1] University of Cyprus, Department of Computer Science, Nicosia, Cyprus
efi.papatheocharous@cs.ucy.ac.cy
[2] SICS Swedish ICT, Swedish Institute of Computer Science, Kista, Stockholm, Sweden
efi.papatheocharous@sics.se
[3] Cyprus University of Technology, Lemesos, Cyprus
andreas.andreou@cut.ac.cy

**Abstract.** This paper aims at empirically investigating the levels of agile adoption by software development organizations in comparison with the type of practices followed, the business sectors and the countries of origin, as well as revealing the geographical organization and distribution of teams both within agile and traditional development environments. Through a dedicated survey conducted, the paper also discovers, analyzes and presents the business strategies, levels of expertise, benefits and concerns of agile adoption within the various participating organizations. The results obtained indicate that agile methods seem to offer opportunities for improved products in terms of quality and suggest a number of critical factors that affect the software process and the adoption of agile methods in general.

**Keywords:** Agile Software Development, Business Strategies, Level of Agile Adoption, Organizational Agile Expertise, Empirical Survey.

## 1 Introduction

Agile software development methods [1], [2], [3] have gained increasing interest by software engineers and researchers worldwide over the last few years. However, scientific research and evidence based on a collection of information gathered through structured surveys is yet scarce [4], [5]. This paper reports recent results from an online survey conducted in 2012, which aimed to discover, organize, analyze and present findings related to the adoption of the agile approach within software organizations worldwide. The results from the survey are related both with companies that are not using the agile paradigm and companies that have already adopted agile methods in their development strategies but to a varying degree. The overall aim is to identify the level of agile adoption among the participants, as well as present the associated business strategies, levels of expertise and benefits gained by following the agile approach within various organizations from different countries that responded to the questionnaire. In particular, this work investigates the following questions:

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 237–246, 2013.

- What is the prevailing geographical location and distribution scheme of teams in agile organizations?
- Which business strategies are commonly adopted by organizations developing software based on the agile paradigm?
- What is the level of expertise found in software organizations using agile methods and what is the degree of agile methods adoption?
- What are the main limitations and benefits from adopting agile within organizations?

According to our findings the adoption of agile methods improves to a large extent productivity and efficiency of the development process, while it also improves the overall levels of quality. These findings are supported by the responses of agile practitioners and a series of statistical tests performed. The motivation and significance of this work is supported by the fact that very few surveys have focused on the identification of the critical factors that affect agile adoption. Most of the research reported in literature thus far compares traditional and agile approaches and identifies advantages and weaknesses on the use of the agile paradigm through the review of empirical studies [6], [7]. In addition, previously, other researchers, through case studies, have identified the need for empirical research frameworks in agile methods in order to make the study of agile development methodologies more comparable [8].

The rest of this paper is structured as follows: Section 2 presents the demographical analysis of the sample that participated in the survey, while section 3 extends this analysis reporting the geographical distribution and dispersion of teams, business strategies, levels of expertise and reflections on the adoption of agile methods. The main limitations and benefits of agile adoption by organizations are also identified here. Section 4 provides the statistical significance and reliability of the answers obtained. Finally, Section 5 summarizes the paper and describes the main conclusions, shortcomings and future research steps.

## 2 Survey Demographics

The survey was conducted between February and March 2012 by means of an online questionnaire. The potential responders were invited through general mail invitations and mailing lists related to agile development. Many of them were contacted through the Agile Development Manifesto webpage (http://agilemanifesto.org/). A total of 5,900 people were contacted and 377 responses (6.38%) were obtained. The validity and reliability of the questions and responses were evaluated and are analytically described in Section 4.

Some basic information related to the survey and the data collected follows: The structure of the questionnaire is summarized in Table 1. Figure 1 depicts the roles of the participants in the survey; the figure shows that the majority of the respondents were project managers ($N$=189, $f$=50%), team leaders ($N$=173, $f$=46%), developers ($N$=162, $f$=43%), designers/architects ($N$=145, $f$=38%), programmers ($N$=133, $f$=35%) and/or development managers ($N$=120, $f$=32%).

**Table 1.** Questionnaire Structure

| Section | Description |
|---|---|
| 1 | Personal Information, Roles & Experience |
| 2 | Company / Organization Information |
| 3 | Team Members & Communication |
| 4 | Knowledge & Strategies |
| 5 | Agile Adoption |

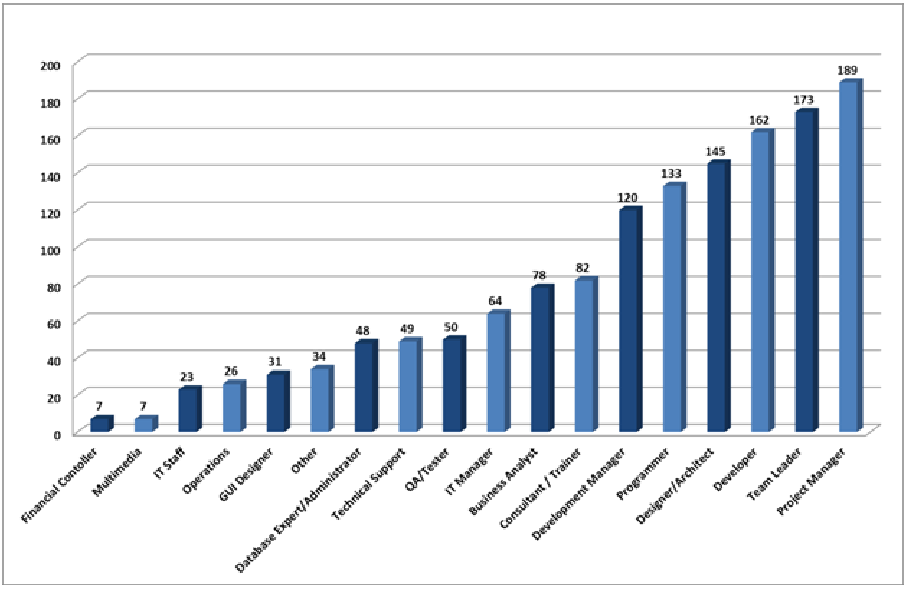



**Fig. 1.** Roles of the respondents to the questionnaire

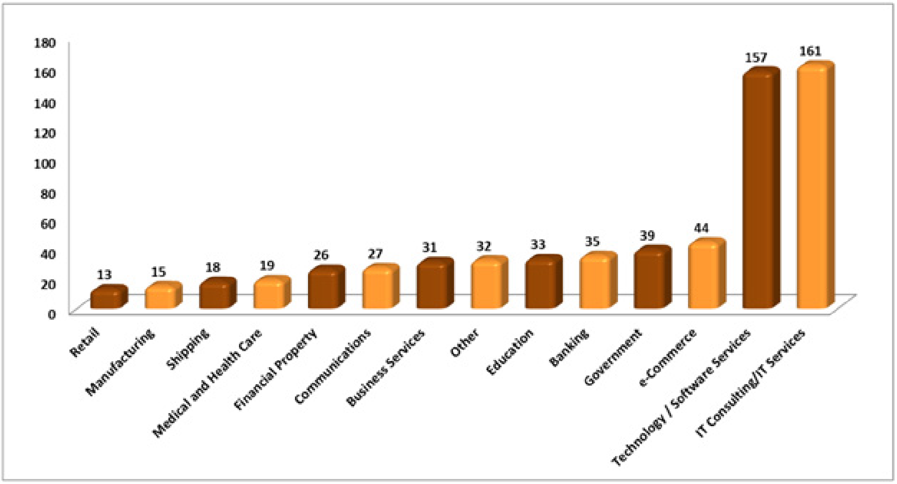



**Fig. 2.** Primary sectors of the organizations

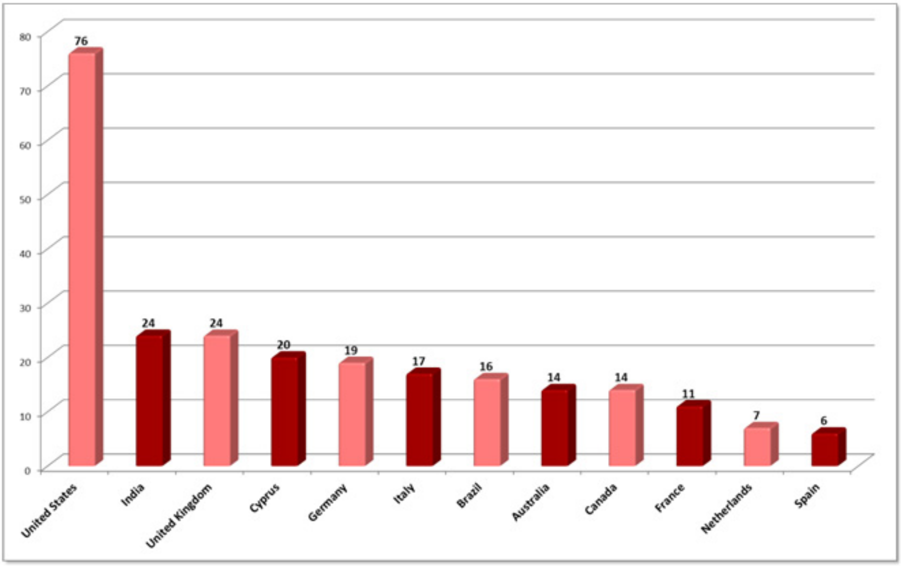



**Fig. 3.** Origin of the organizations

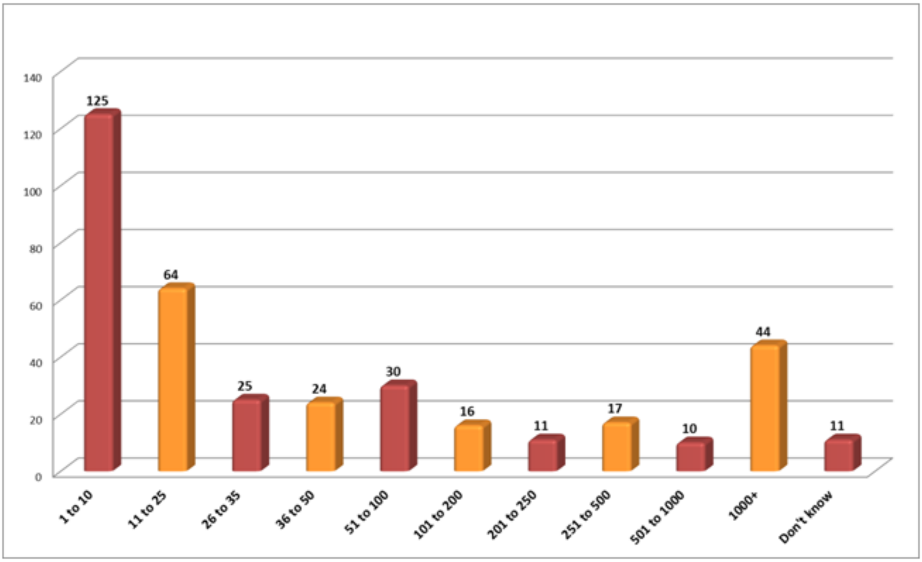



**Fig. 4.** Organization size (number of developers)

Figure 2 summarizes the type of sectors of the organizations involved in the sample and Figure 3 shows their countries of origin. Most of the organizations ($N$=192, $f$=53%) that participated in the survey did not use any Software Capability Quality standards (e.g., ISO9000 series, SPICE and CMMI), a large percentage ($N$=142, $f$=40%) use such standards and a small percentage ($N$=26, $f$=7%) did not know if their organization is using any standards at all.

Figure 4 presents the size of the software organizations in terms of number of developers. It is evident that the majority involves small to medium enterprises numbering between 1 and 25 persons (over 50%). This finding is somehow expected since the survey targeted mostly agile organizations and taking into consideration the fact that agile is often adopted by small teams. Nevertheless, there is a substantial percentage (29%) of organizations employing a large number of developers (above 100) which questions the above argument and indicates that large teams sometimes apply the agile paradigm as well.

# 3 Survey Analysis

This section involves the analysis of the survey results regarding the geographical distribution and dispersion of agile teams and the adoption levels of agile within the organizations that participated in the survey. Furthermore, it presents the business strategies of those organizations, their levels of expertise, as well as the benefits enjoyed as a result of following the agile approach according to the level of adoption. In case agile is not adopted by an organization the main obstacles are reported. The overall aim is to obtain observations and compare them in relation to the agile practices followed so as to extract meaningful information regarding the disadvantages and benefits of adopting agile.

## 3.1 Team Distribution and Organization

Comparing the size of the teams involved in projects within organizations that follow agile methods and traditional (disciplined) ways of software development, a large number of responders ($N$=89, $f$=24%) referred to small teams of 1-10 people. This percentage is the majority of the agile organizations that responded to the questionnaire. In addition, the majority of the organizations following traditional methods of development reported teams comprising of 1-10 people ($N$=36, $f$=10%) and 11-25 people ($N$=25, $f$=7%).

Regarding team distribution within these teams, most teams ($N$=112, $f$=30%) have some people that are working in locations with more than 3 hours time zone difference away. In addition, a large percentage of teams work in the same environment, i.e., in the same room ($N$=93, $f$=25%). The team distribution across the teams is shown in Figure 5. It becomes evident that the majority of the teams belonging to these two categories are also using agile methods, whereas the majority of members of teams following traditional development methods are located in the same building. Table 2 reports the corresponding figures for agile and traditional teams.

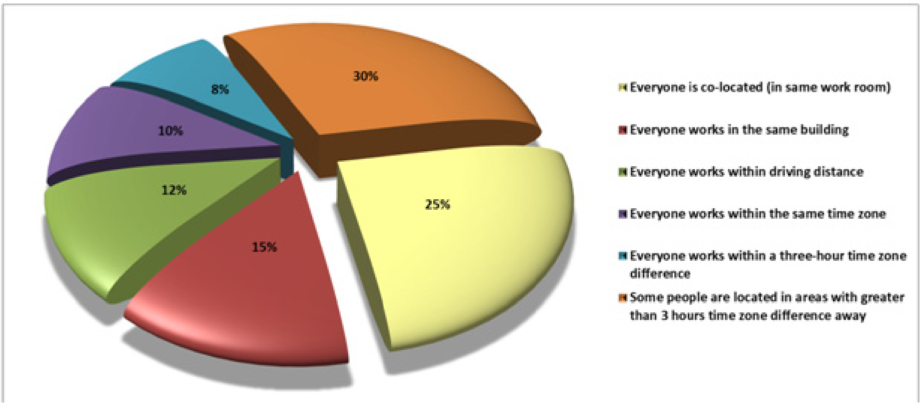



**Fig. 5.** Team geographical organization and distribution

**Table 2.** Team distributions for agile and traditional organizations

| Team Distribution \ Type of Organization | Agile | Traditional |
|---|---|---|
| Everyone is co-located (same room) | 63 (17%) | 30 (8%) |
| Everyone works in the same building | 30 (8%) | 26 (7%) |
| Everyone works within driving distance | 26 (7%) | 18 (5%) |
| Everyone works within same time zone | 26 (7%) | 14 (4%) |
| Everyone works within a three-hour time zone difference | 26 (7%) | 6 (2%) |
| Some people are located in areas with greater than 3 hours time zone difference away | 85 (23%) | 27 (7%) |

### 3.2 Business Strategies

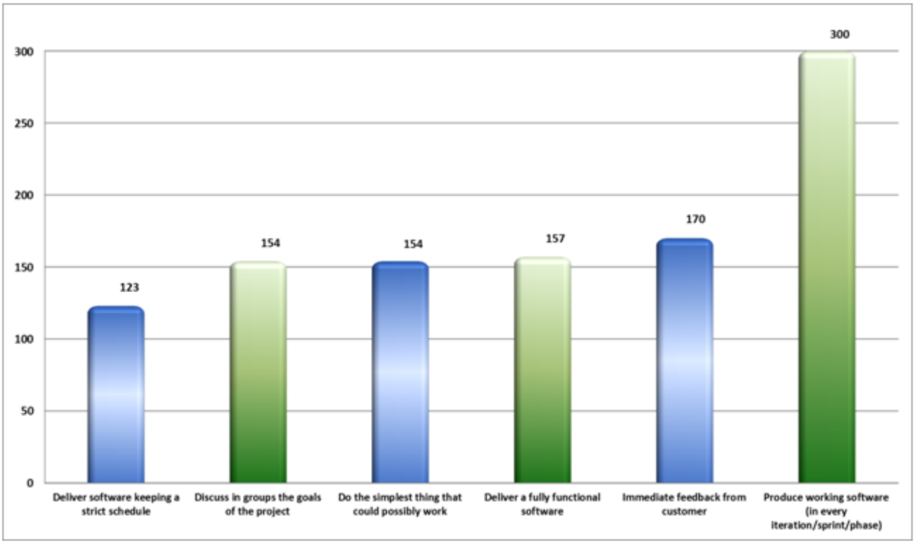



**Fig. 6.** Business strategies followed by the teams

Regarding the business strategies followed by the organizations (reported in Figure 6), it seems that the corresponding teams aim primarily to deliver working software in every iteration/sprint/phase. Also the figure shows that the organizations value most two of the main principles of the Agile Manifesto (i.e., delivering working software and obtaining continuous feedback from the customer). The success of adopting such strategies and the reflections of the teams is discussed further in the following sections.

### 3.3 Levels of Expertise and Adoption of Agile

Regarding knowledge and expertise in agile methods the responders expressed extensive knowledge in 38%, average in 30%, very extensive in 19%, limited in 9% and very limited in 4% of the cases. Moreover, Table 3 shows the level of agile methods expertise reported by the organizations following the agile and traditional development methods. The majority of agile organizations report extensive knowledge (32%), while 18% and 16% report average and very extensive knowledge in agile respectively; traditional organizations report average knowledge on agile (13%).

**Table 3.** Agile methodology knowledge rating for agile and traditional organizations

| Knowledge \ Type of Organization | Agile | Traditional |
|---|---|---|
| Very extensive | 61 (16%) | 9 (2%) |
| Extensive | 122 (32%) | 20 (5%) |
| Average | 66 (18%) | 49 (13%) |
| Limited | 7 (2%) | 27 (7%) |
| Very limited | 0 (0%) | 16 (4%) |

The responders when explicitly asked to state whether their organization followed agile methodologies or not, a high percentage replied positively (*N*=255, *f*=72%), a significantly smaller percentage replied negatively (*N*=86, *f*=24%), while very few replied that they didn't know (*N*=15, *f*=4%). These responses show that even the responders that evaluate their knowledge in agile as average their organizations are in fact to a large extent involved or following agile methodologies (irrespectively of their agile or traditional development process nature).

The responders who mentioned that their organization did not adopt or did not know if they have adopted agile methodologies in the past were asked to identify the main obstacles they might have in adopting agile. Their responses ranked in popularity are as follows: Finding it difficult to change their way of working (*N*=47, *f*=43%), the management of the organization is opposed to change (*N*=35, *f*=32%), lack of documentation (*N*=34, *f*=30%), non-familiarity with agile methodologies (*N*=31, *f*=28%), lack of experts (*N*=30, *f*=26%), development team is opposed to change (*N*=28, *f*=25%), lack of engineering discipline (*N*=28, *f*=25%), lack of planning (*N*=24, *f*=21%), lack of engineering talent/skill quality (*N*=23, *f*=20%), risk of software quality (*N*=21, *f*=19%), lack of predictability (*N*=21, *f*=18%), lack of managerial control (*N*=18, *f*=16%), other (*N*=18, *f*=16%), inability to scale (*N*=15, *f*=13%) and don't know (*N*=9, *f*=8%).

Finally, the responders that follow the agile paradigm within their organization were asked to rate the significance of the benefits obtained. Figure 7 demonstrates the high rating obtained for each of these benefits.

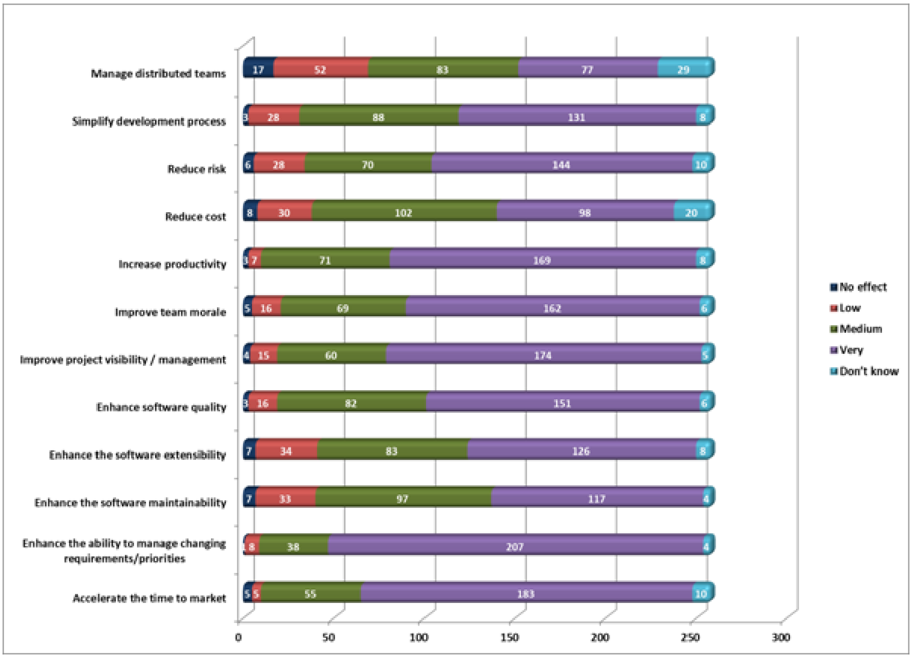



**Fig. 7.** Benefits from implementing agile within organizations

In addition, the significance of agile methods adoption was evaluated by the responders of the questionnaire and the majority rated as very significant the following benefits: Agile methods helped in accelerating the time to market, enhanced the ability to manage changing requirements/priorities and contributed to software maintainability and extensibility. Furthermore, they improved software quality, project visibility/management and team morale. Finally, they helped to increase productivity, reduce cost and risk, and in general simplified the development processes.

## 4 Survey Evaluation

This section involves the statistical evaluation of the answers obtained and analysis of the results of the survey. Initially, we investigated the sample size. According to [9] the minimum sample size to reach to reliable results for the particular type of questionnaire equals to 340 ($CI$=95%, $\alpha$=0.05, $E$=± 5%), a requirement which is satisfied by the sample size of the particular study (with a total of 377 valid answers).

In addition, the internal consistency of the sample was tested based on the Cronbach's $\alpha$ (alpha) coefficient [10] which is widely used in many fields such as social sciences and business. Cronbach's $\alpha$ (alpha) coefficient reports internal reliability, that is, how closely related a set of items are as a group and it is used with multiple questions the answers of which are expressed on a likert-scale. In addition, Pearson's correlation was used for testing the correlation between quantitative responses, whereas other tests were also performed related to content validity, face validity, categorical validity and reliability.

Regarding the significance value of the organizations using agile methods, Cronbach's $\alpha$ (alpha) coefficient was found above 0.5, i.e., 0.881, thus inferring reliability of the values obtained for likert-scale replies (refer to the agile methodology knowledge rating).

## 5 Conclusions

The main target of this work was to present empirical observations and draw some conclusions regarding the use of agile methods in software development organizations worldwide. For this purpose an online questionnaire was designed and used. The results were gathered, analyzed and presented in this paper. Of particular interest was the business strategies followed, the benefits stemming from the adoption of agile methods and the obstacles in adopting the agile paradigm in the various projects developed. Also, other interesting issues were investigated, such as the team size, team location and distribution and reflections upon the use of agile.

The main limitations of this survey relate with the fact that a structured questionnaire was used and responders were not able to request further explanations or clarifications regarding the questions. The content of the questionnaire was examined in terms of reliability through statistical measures and was found sufficient. Other limitations involve the fact that many of the potential responders invited to participate in the survey were found through agile-related web-pages.

Thus, future work will include expansion of the survey, including other issues related to agile adoption and in particular examining a wider population sample. Moreover, focus of future work will be the identification of the difficulties of adopting agile methods in today's competitive software development environments, as well as in particular contexts, and also the examination of the success rates of teams with varying distributions. Future work will also focus on finding out the advantages and problems of following the agile paradigm and also examining the common agile techniques adopted by most organizations today.

## References

1. Cockburn, A.: Agile Software Development. Addison-Wesley, Boston (2002)
2. Agile Alliance Web site, `http://www.agilealliance.org`
3. Highsmith, J.: Agile Software Development Ecosystems. The Agile Software Development Series. Addison-Wesley, Boston (2002); Cockburn, A., Highsmith, J. (eds.)

4. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development Methods: Review and Analysis. VTT Technical report (2002)
5. Erickson, J., Kalle, L., Keng, S.: Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. Journal of Database Management 16(4), 88–100 (2005)
6. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J.: The Impact of Agile Practices on Communication in Software Development. Empirical Softw. Engg 13(3), 303–337 (2008)
7. Dybå, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A systematic review. Inf. Softw. Technol. 50(9-10), 833–859 (2008)
8. Petersen, K., Wohlin, C.: A Comparison of Issues and Advantages in Agile and Incremental Development Between State of the Art and Industrial Case. J. Syst. Softw. 82(9), 1479–1490 (2009)
9. Crimp, M.: The Marketing Research Process, 2nd edn. Prentice Hall International, UK (1985)
10. Cronbach, L.J.: Coefficient alpha and the Internal Structure of Tests. Psychometrika 16, 297–334 (1951)

# Definitions of Agile Software Development and Agility

Maarit Laanti[1], Jouni Similä[2], and Pekka Abrahamsson[3]

[1]Nitor Delta, Finland
Maarit.Laanti@nitorcreations.com
[2] University of Oulu, Finland
Jouni.Simila@oulu.fi
[3] Free University of Bozen-Bolzano, Italy
Pekka.Abrahamsson@unibz.it

**Abstract.** The Agile Manifesto and Agile Principles are typically referred to as the definitions of "agile" and "agility". However, many other definitions exist in the literature. Thus the different definitions provide interesting source for research. For each definition we examine where their emphasis is and compare that to the emphases found in the Agile Principles.

## 1 The Agile Manifesto as a Definition

Agile Software Development is most typically defined via the “Manifesto for Agile Software Development” [1, 2]. The Agile Manifesto was formulated in 2001, when Cockburn invited a group of respected software engineering professionals to ski and to discuss topical issues in software engineering. Cockburn [2] defines Agile Methods as techniques that allow a team to track rapid changes in people, technology, and business. He further explains that although the ideas of agile development are based to some extent on the theory of constraints and Lean Thinking, the agile way of working was born separately.

The Agile Manifesto is a document that is discussed and argued about a lot. The argumentation is partially related to how the Agile Manifesto is understood (or not understood) and whether people agree or do not agree about it. Partially because of this, a non-profit organization called the Agile Alliance was formed in 2001 to promote the principles and values listed in the Agile Manifesto [4].

Because of the controversy some people have an urge to explain the manifesto and even some agile experts such as Ambler [5], would like to make some changes or updates to it. The Agile Manifesto has also been criticized; for example, Coplien has stated that the Manifesto should talk about Usable software rather than Working software in order better to take the usability aspect into account.

There has also been critiques of the Agile Manifesto, stating that it is “too vague” to be used as a basis for scientific work. It has been claimed that it lacks a proper grounding in management theory and philosophy [3]. It has also been stated that even though there are some methods that are called Agile Methods (such as Scrum and Extreme Programming), these methods focus heavily on some of the Agile Principles, but not evenly on all of those. As an alternative, Conboy and Fitzgerald [3] proposed a conceptual framework of Agile Methods, explaining agility as flexibility which

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 247–258, 2013.

reflects the robust, proactive, reactive, and temporal dimensions: "*the ability of an entity to proactively, reactively, or inherently embrace change in a timely manner, through its internal components and its relationships with its environment*."

The problem of the framework proposed by Conboy and Fitzgerald is that it covers only one aspect of the Agile Manifesto, i.e., the ability to embrace change, while it leaves the simplicity of processes, incremental deliveries, and the people aspect all uncovered and unrecognized.

Conboy and Fitzgerald also compare Agile Software Development with Lean Thinking and the Toyota Production System. Zaninotto was apparently the first one to make this connection in a keynote talk at the XP2002 conference that discussed the tie-ins between Agile Methods and Lean Manufacturing [6, 7]. However, we know from the testimony of Cockburn [2] that the Agile Manifesto was born separately and independently from Lean Thinking, Lean Manufacturing, and Agile Manufacturing, although Cockburn admitted in his keynote speech at the ICAM 2005 conference that some of the signatories of the Agile Manifesto may have known about these methods, and that might have had an impact on why they decided to call the new paradigm "Agile Software Development" and not "adaptive software development", which was a name suggested earlier by Highsmith [8].

## 2 What Is Emphasized in Different Agile Definitions

Table 1 contains an analysis of the Agile Principles and where the emphasis is put in each principle. The first principle places an emphasis on: 1. customer satisfaction, 2. continuous delivery, 3. value, and 4. early deliveries. The second principle places an emphasis on 5. adaptability, 6. competitiveness, and customer benefit. Customer benefit (or the customer's competitive advantage) is close to customer satisfaction, so those two can be combined into 1. customer satisfaction/benefit. The third principle places an emphasis on frequent deliveries. This is close to early deliveries, so these two can be combined into 4. early/frequent deliveries. The fourth principle emphasizes 7. collaboration. The fifth principle places an emphasis on 8. motivated individuals, 9. good environment, 10. support, and 11. trust. The sixth principle places an emphasis on 12. efficiency and 13. communication. The seventh principle is the only one on metrics, and places an emphasis on 14. measure progress via deliverables. The eighth principle emphasizes 15. sustainability and 16. people. The ninth principle emphasizes 17. focus on technical excellence and 18. good design as an enabler of agility. The tenth principle emphasizes 19. simplicity and 20. optimize work and the eleventh 21. self-organization. The twelfth, and last, principle emphasizes 22. built-in improvement of efficiency and behavior.

Given that the Agile Principles are widely agreed on as defining Agile Software Development, it would be interesting to compare any other definitions of Agile Software Development and agility against the Agile Principles. The analysis of the emphasis of the Agile Principles is further used as a basis for such a systematic analysis in this thesis.

**Table 1.** Agile Principles and what they emphasize

| **Agile Principle** | **Emphasis** |
|---|---|
| Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | Customer satisfaction, Continuous delivery, value, early deliveries |
| Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | Adaptability, competitiveness, customer benefit |
| Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale. | Frequent deliveries |
| Business people and developers must work together daily throughout the project. | Collaboration |
| Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | Motivated individuals, good environment, support, trust |
| The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. | Efficiency, communication |
| Working software is the primary measure of progress. | Measure progress via deliverables |
| Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. | Sustainability, people |
| Continuous attention to technical excellence and good design enhances agility. | Focus on technical excellence, |
| Simplicity – the art of maximizing the amount of work not done –is essential. | Simplicity, optimize work |
| The best architectures, requirements, and designs emerge from self-organizing teams. | Self-organization |
| At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. | Built-in improvement of efficiency and behavior |

## 3 Tenth Anniversary of the Manifesto

In February 2011 30 leading agile thinkers convened to discuss the manifesto, Out of the discussions came a consensus that the agile community should:

- demand technical excellence
- promote individual change and lead organizational change
- oganize knowledge and promote education
- maximize value creation across the entire process [9]

When this list of four items is compared with the emphasis of the twelve Agile Principles we can see that a lot of the original emphasis has been lost: there is no mention of customer satisfaction or benefit, Competitiveness, motivated individuals, a good environment, support, trust, communication, collaboration, sustainability, people, simplicity, or self-organization. Other areas of emphasis are now seen as being even more important, especially the demand for technical excellence.

It should be noted that while the promotion of individual change and leading organizational change is about people, these areas of emphasis are not seen in the same way as was originally the case in the Agile Principles: the new area of emphasis focuses on the dynamics of the organization (i.e., change), whereas the Agile Principles are "taking what is given"; building around motivated individuals. The former is a view of someone developing the organization; the latter (in the Agile Principles) is of a project manager's or entrepreneur's view on selecting the people to be on the team. The viewpoint on Adaptability is different: where the Agile Principles focus on the Adaptability of the Requirements, the view in promoting individual change and leading organizational change is on the adaptability of the organization. Thus it can be seen that this meeting raised promotion of individual change and leading organizational change as a new area. Similarly, the organization of knowledge and promotion of education can also be seen as a new area.

Continuous delivery, value, early or frequent deliveries, efficiency, measuring progress via deliverables, and optimizing the work, as well as built-in improvement of efficiency and behavior, can all be seen as being included in the maximization of value creation across the entire process.

## 4 More Definitions

In 1990 the US Congress became concerned about American industrial capability not matching its competitors, especially Japan. That is why a special technology advisory board was set up to study how US industry should be developed. Agile Manufacturing, an Agile Competitive Environment, and the Agile Enterprise were proposed by this group as answers to the question of how to raise competitiveness. [10] In 1994 some of the group members, namely Goldman, Naegel, and Preiss, published a book called Agile Competitors and Virtual Organizations that was based on this work.

Goldman defines agility as: "*a comprehensive response to the business challenges of profiting from rapidly changing, continually fragmenting, global markets for high-quality, high-performance, customer-configured goods and services. It is dynamic, context-specific, aggressively change-embracing, and growth-oriented. It is not about improving efficiency, cutting costs, or battening down the business hatches to ride out fearsome competitive "storms", it is about succeeding and about winning: about succeeding in emerging competitive arenas, and about winning profits, market share, and customers in the very center of the competitive storms many companies now fear.*" [11]

Interestingly, Cockburn [12] describes this as "the best description he has found for agility". Kettunen [13] claims that there is no uniform definition of "Agile Software Development" but provides a comprehensive list of different definitions of Agile Software Development, in chronological order. A copy of that table is presented in Table 2, but a third column is now added in this dissertation work. The third is an analysis of each agile definition, explaining where the emphasis is in the corresponding definition.

**Table 2.** Definitions of agile software development [adapted from Kettunen 2009, third column added]

| **Source** | **Definition of agile** | **Emphasis of the corresponding definition** |
|---|---|---|
| Cockburn 2001 | Being effective and maneuverable. Use of light-but-sufficient rules of project behavior and the use of human and communication-oriented rules. | Effective, steerable, rule-based, people, communication |
| Anderson 2003 | Ability to expedite. | Speed |
| Larman 2003 | Rapid and flexible response to change. | Speed, flexibility, responsiveness |
| Schuh 2004 | Building software by empowering and trusting people. Acknowledging change as a norm, and promoting constant feedback. Producing more valuable functionality faster. | People, empowerment, change, feedback, value, speed |
| Lyytinen 2006 | Discovery and adoption of multiple types of Information Systems Development innovations through garnering and utilizing agile sensing and response capabilities. | Delivery, innovations, responsiveness |
| Subramaniam 2005 | Uses feedback to make constant adjustments in a highly collaborative environment. | Feedback, adaptability, collaboration |
| Ambler 2007 | Iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams with "just enough" ceremony that produces high-quality software in a cost-effective and timely manner which meets the changing needs of its stakeholders. | Iterative, incremental, self-organizing, less process-driven, collaborative, cost-conscious, speed, customer-driven |

**Table 2.** *(Continued)*

| | | |
|---|---|---|
| Nerur and Balijepally 2007 | Define Agile software development via strategic thinking (of uncertainty), holographic organization theory, "emergent metaphor of design" and Agile Methods as people-centric, competent people and their relationships, high customer satisfaction through quick delivery of quality software, active participation of concerned stakeholders; creating and leveraging change. Evolutionary delivery through short iterative cycles, intense collaboration, self-organizing teams and high degree of developer discretion. Learning, teamwork, self-organization and personal empowerment. Responsiveness and flexibility. Interchangeability of roles and jobs based on autonomy. | Strategic thinking, uncertainty, chaos theories, holographic organization, non-traditional, emergent design, people-centric, competent people and their relationships, high customer satisfaction, quick delivery, active participation, creating and leveraging change, short iterative cycles, intense collaboration, self-organizing teams, developer discretion, learning, teamwork, self-organization, personal empowerment, responsiveness, flexibility, heterarchy, role interchangeability and autonomy. |
| IEEE 2007 | Capability to accommodate uncertain or changing needs up to a late stage of the development (until the start of the last iterative development cycle of the release). | Iterative, responsive |
| Wikipedia 2007 | Conceptual framework for software engineering that promotes development iterations throughout the life-cycle of the project. | Iterative, conceptual framework |

Cockburn's [14] definition of Agile Software Development places the emphasis in such a way that agile is defined as 1) effective, 2) steerable, 3) rule-based, 4) (about) people, and 5) communication. Only the latter two are the same as can be found in the Agile Principles; see Table 2.

Anderson [15] places the emphasis only on 1) speed. Besides speed, Larman [16] also places the emphasis on 2) flexibility and 3) responsiveness. Schuh [17] also places the emphasis on 1) speed, but also on 2) people, 3) empowerment, 4) change, 5) feedback, and 6) value.

Lyytinen [18] places the emphasis on 1) feedback, 2) adaptability, and 3) collaboration. Subramaniam [19] emphasizes 1) feedback, 2) adaptability, and 3) collaboration. Ambler [20] states that agile is 1) iterative, 2) incremental, 3) self-organizing, 4) less process-driven, 5) collaborative, 6) cost-conscious, 7) (about) speed, and 8) customer-driven.

Nerur and Balijepally [21] state that the field of software development has progressed by leaps and bounds like also 1) strategic thinking facing 2) uncertainty and 3) non-traditional, 4) emergent design as well as 5) chaos theories and 6) holographic organization theories have and define agile software development by defining Agile Methods as 7) people-centric, 8) competent people and their relationships, 9) high customer satisfaction through quick delivery of quality software, 10) active participation of concerned stakeholders; 11) creating and leveraging change. Important features are evolutionary delivery through 12) short iterative cycles, 13) intense collaboration, 14) self-organizing teams and high degree of 15) developer discretion. Agile development's value depends largely on 16) learning, 17) teamwork, 18) self-organization and 19) personal empowerment. 20) Responsiveness and 21) flexibility are achieved through 22) "heterarchy" characterized by self-organizing collaborating teams. Holistic teams encourage 23) interchangeability of roles and jobs based on 24) autonomy.

IEEE [22] states that agile is 1) iterative and 2) responsive, whereas Wikipedia 2007 definition [23] stated that agile is 1) iterative and 2) a conceptual framework.

Those definitions that name only one or two points of emphasis can be considered narrow. The other definitions partially cover the same points of emphasis as the Agile Principles (see Table 2.1) but use slightly different terms or viewpoints on agility. When Cockburn states that agile is about communication – which is also one point of emphasis in the Agile Principles – Ambler states it is about collaboration. These kinds of nuances might seem irrelevant, but they can cause confusion in a large organization when Agile Methods are being used. In practice, people will come and ask what it is that the organization is aiming to do – and is collaboration better than communication?

Obviously, the length of the definition is not the only measure of its goodness, although the longer the definition is, the more points that are emphasized it can cover. When the Agile Principles are compared to Goldman's definition, it can be seen that there is no single point that is emphasized that can be stated as being the same. Instead of customer satisfaction or benefit Goldman talks about "customer-configured goods and services". Even the contrary is true: while one aspect found in the Agile Principles is "efficiency", Goldman [11] states that being agile "is not about efficiency".

Nerur's and Balijepally's [21] definition is broadest of all definitions. They examine agile software development from the perspective of Agile Methods in general, and compare the changes that have happened in software development to how other disciplines have changed. This kind of broad view puts Agile Methods into perspective and thus is very useful read to practitioners; however this paper still lacks for very coincided definition that would be of practical use. The definition have some common emphases with Agile Principles however there is also emphases to points missing from other definitions – perhaps rooting to other disciplines than software development.

## 5 Agile Software Development on the Project Level

In order to understand how we can scale Agile Software Development and achieve agility we can take a look at the further definitions that are available, especially from sources that look into agility from a perspective that is wider than that of just one team.

One of these attempts to define agility in a larger context took place in 2005, when Cockburn gathered a group of project managers together to discuss agility within a project context and from a project management viewpoint. This gathering resulted in the Declaration of Interdependence (DOI), which links people, projects, and value with agile and adaptive approaches. The Declaration of Interdependence states: [24]

We are a community of project leaders that are highly successful at delivering results. To achieve these results:

- We increase return on investment by making continuous flow of value our focus.
- We deliver reliable results by engaging customers in frequent interactions and shared ownership.
- We expect uncertainty and manage for it through iterations, anticipation, and adaptation.
- We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
- We boost performance through group accountability for results and shared responsibility for team effectiveness.
- We improve effectiveness and reliability through situationally specific strategies, processes, and practices.

Analyzing the Declaration of Independence gives hints as to how the emphasis of Agile Methods may change when the viewpoint is changed from a team perspective to a project perspective. Table 3 contains an analysis of the DOI and states the emphasis of each statement.

**Table 3.** Emphasis in agility definition in the Declaration of Interdependence

| DOI statement | Emphasis |
|---|---|
| We increase return on investment by making continuous flow of value our focus. | Maximizing return on investment, flow of value |
| We deliver reliable results by engaging customers in frequent interactions and shared ownership. | Customer engagement, delivery accuracy, collaboration |
| We expect uncertainty and manage for it through iterations, anticipation, and adaptation. | adaptability, proactivity, anticipation |
| We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference. | Innovativeness, people, environment |
| We boost performance through group accountability for results and shared responsibility for team effectiveness. | Shared responsibility, effectiveness |
| We improve effectiveness and reliability through situationally specific strategies, processes, and practices. | Reliability, situationality |

The first statement in the DOI places the emphasis on 1) maximizing ROI and 2) flow of value. The second statement in the DOI emphasizes 3) customer engagement, 4) delivery accuracy, and 5) collaboration. The third statement emphasizes 6) adaptability, 7) proactivity, and 8) anticipation. The fourth statement emphasizes 9) innovativeness, 10) people, and 11) environment. The fifth principle emphasizes 12) shared responsibility and 13) effectiveness. The sixth principle emphasizes 14) reliability and 15) situationality.

When Table 3 is compared with Table 1 and the emphasis in the Agile Principles it can be seen that five of these aspects are the same (value, collaboration, adaptability, people, and environment) but the majority are new (maximizing ROI, delivery accuracy, proactivity, anticipation, innovativeness, reliability, and situationality) or the viewpoint is slightly different (customer engagement rather than customer satisfaction, shared responsibility rather than self-organization). This raises the question of whether these differences can be explained just with a different viewpoint, or if the understanding of Agile Methods has evolved, as the DOI was written four years after the Agile Manifesto. New points of emphasis (maximizing ROI, delivery accuracy, proactivity, anticipation, innovativeness, reliability, and situationality) can also be considered as requirements from the project management level for Agile Methods.

## 6 More Recent Definitions

Research papers seem to avoid defining Agile Software Development and agility, or define it via references to a few existing sources, e.g., the definition of Conboy and Fitzgerald [3], or define agility via the methods researched. Assuming that the understanding of Agile Methods increases as people practice these methods more, it is interesting to see how the definitions have evolved and how the emphasis differs from the Agile Manifesto and Agile Principles. Since research provides little help in this respect, the latest agile literature is researched. Advancing Agile Methods seems to be primarily driven by industry practitioners, not by academic researchers.

Wikipedia [25] defines agile as 1) a group of software development methodologies, 2) iterative, 3) incremental, 4) self-organizing, 5) cross-functional, and 6) evolutionary. Poppendieck [26] defines agile as 1) a system development frame, 2) technical practices, and 3) effectiveness. Larman and Vodde [27] place the emphasis in agility as 1) ability to change quicker and easier than the competition. Later [28], Larman and Vodde define agile via House of Lean, i.e., as systems improvement using Agile Practices. Appelo states that Agile is context-specific, having its roots in complexity theory. Leffingwell [29] sees Agile as context-specific processes including a set of practices bringing business benefits. Cohn [30] does not give a definition of Agile Software Development but asks the reader to refer to his earlier books. Here, agile is seen as 1) non-sequential, 2) non-traditional, 3) high-quality, 4) speed, 5) meets user needs, 6) low-cost, 7) process, 8) productivity, 9) visibility, 10) predictability, and 11) in-control.

From Table 4 it can be seen that even these recognized gurus do not have a unified vision of what Agile Software Development is and are struggling with the definition.

If the Wikipedia definition is omitted as practical (and thus as an outlier), the rest of the definitions can be categorized as follows:

- A null definition (agile is agile ; differing from all the previous and succeeding definitions) Larman's 2009 [27] recursive definition that agile is agile and a list of negative statements about what it is not, which contradicts almost all of the succeeding definitions of Agile Software Development and agility.
- Traditional. Cohn's 2009 definition as a negative statement of what it is not [30, 29] also aligned himself with the previously discussed (positive) aspects.
- Set of practices. Agile as a set of practices that should be used together with Systems Thinking [31, 28, 26].
- Context-specific. Agile as a context-specific set of processes and practices [29] which may vary as the problem that we are solving varies, rooted in complexity theories [32].

Most of the newest definitions of Agile Software Development have stopped talking about effectiveness, but describe agile rather as a set of practices that you can try when doing systems improvement. But agile must be more than just a set of practices that are applied: while the first attempts at putting agile into use in large organizations were about trying out some practices [24], there was a lot of complaining that agile must mean a lot more than some teams (or even some individuals) following some Agile Practices only: for example, you could well do pair coding and still follow a traditional process.

An organization needs to know when it has become agile. The agile literature defines no point after which an organization has adopted enough practices to be called agile. Rather, the literature presents various operational models but little guidance as how to get to that dream state. It has also been stated that agility is rather the mindset with which to approach the problems at hand, but an organization cannot simply change to an agile mode by simply stating that it has done so. A large organization would need something it can develop, deploy, and measure.

Appelo [32] states that most people have got agility wrong, because they have not understood that agile originates from complexity theories — or rather that they do not understand complexity theories. Thus any simplistic, linear model (or attempt to create one) is bound to fail, and we should rather focus on the adaptability, not the predictability. In fact, conflict is a natural aspect of Complex Systems and a prerequisite for creativity and innovation [32]. This provides an additional challenge for developing large-scale agile models.

## 7 Conclusion

In a way, the definitions can also be taken as a promise what Agile Software Development has to offer. This interpretation would explain how Agile Software Development is sometimes criticized as being a "silver bullet". It might just be better for the clarity, if we would start talking about different agile practices and what benefits those do bring instead of speaking generally about "agile" and "benefits of agile". Another view on this data is that what we understand with "agile" has been developing over time. As new concepts are born, we have struggle on understanding those until finally enough time has passed and we become familiar with those concepts with their own terms. [34] This is why we often create new concepts on the basis of old concepts – e.g. the concept of "airport" is built upon the concept of "port" although the two have only little in common. As Agile Software Development is an abstract concept, it will likely take even more time before it is fully understood. The different perceptions that people have on "agile" and "agility" make deployment of Agile Methods very difficult. The final conclusion is, that people really do mean different things when they are talking about Agile Software Development and agility. Before creating too many misunderstandings, it might be better just to check what the other person's perceptions are. While the other person may completely focus on people's interactions within one team and how to create hyper-productive teams the other one may be worried about optimizing the whole organization as a system or leading the product development organization from the basis of complexity science.

## References


1. Agile Manifesto (2001), `http://www.agilemanifesto.org` (accessed on July 2011 and May 2012)
2. Cockburn, A.: Two Case Studies Motivating Efficiency as "Spendable" Quantity. In: Proceedings of the International Conference on Agility (2005)
3. Conboy, K., Fitzgerald, B.: Toward a Conceptual Framework for Agile Methods: a Study of Agility in Different Disciplines. In: Proc. ACM Workshop on Interdisciplinary Software Engineering Research (WISER), pp. 37–44 (2004)
4. Fowler, M., Highsmith, J.: The Agile Manifesto. Software Development (August 2001)
5. Ambler, S.: Examining the Agile Manifesto (2011), `http://www.ambysoft.com/essays/agileManifesto.html` (accessed on July 2011 and May 2012)
6. Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit. Addison Wesley (2003) ISBN 0-321-15078-3
7. Fowler, M.: Is Design Dead? (2004), `http://www.martinfowler.com/articles/designDead.html` (accessed on July 2011 and May 2012)
8. Highsmith, J.: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing Company (1999) ISBN-10: 0932633404
9. Stevens: What is next for the Agile Manifesto (2011), `http://www.dennisstevens.com/2011/02/13/whats-next-for-the-agile-manifesto/` (accessed on August 2011 and May 2012)
10. Preiss, K.: Agility – the Origins, the Vision and the Reality. In: Proceedings of the International Conference on Agility (2005)
11. Goldman, S., Naegel, R., Preiss, K.: Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer. Wiley (1994) ISBN 0471286508
12. Cockburn, A.: Agile Software Development: the Cooperative Game, 2nd edn. Addison-Wesley (2006) ISBN-10: 0321482751
13. Kettunen, P.: Agile Software Development in Large-Scale New Product Development Organization: Team-Level Perspective. Helsinki University of Technology, Doctoral Dissertation. TKK Dissertations 186 (2009) ISBN 978-952-248-113-9

14. Cockburn, A.: Agile Software Development. Addison-Wesley (2001) ISBN-10: 0201699699
15. Anderson, D.J.: Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Prentice Hall (2003) ISBN-10: 0131424602
16. Larman, C.: Agile & Iterative Development. A Manager's Guide. Addison-Wesley Professional (2003) ISBN-10: 0-13-111155-8
17. Schuh, P.: Integrating Agile Development in the Real World. Charles River Media, Inc. (2004) ISBN-10: 1584503645
18. Lyytinen, K., Rose, G.M.: Information System Development Agility as Organizational Learning. European Journal of Information Systems 15, 183–199 (2006)
19. Subramaniam, V., Hunt, A.: Practices of an Agile Developer – Working in the Real World. The Pragmatic Bookshelf (2005) ISBN-10: 097451408X
20. Ambler, S.W.: Disciplined Agile Software Development: Definition (2007), `http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm` (accessed on November 2007 and May 2012)
21. Nerur, S., Balijepally, V.G.: Theoretical Reflections on Agile Development Methodologies. Communications of the ACM 50(3) (March 2007)
22. IEEE, Draft Recommended Practice for the Customer-Supplier Relationship in Agile Software Projects. P1648/D5 (2007)
23. Wikipedia, `http://en.wikipedia.org/wiki/Agile_software_development` (accessed on July 2007)
24. DOI, Declaration of Interdependence (2005), `http://www.pmdoi.org/` (accessed on July 2011 and May 2012)
25. Wikipedia, `http://en.wikipedia.org/wiki/Agile_software_development` (accessed on July 2007)
26. Poppendieck, M., Poppendieck, T.: Leading Lean Software Development: Results are Not the Point. Addison Wesley (2010) ISBN-10: 0-321-62070-4
27. Larman, C., Vodde, B.: Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum. Addison Wesley (2009) ISBN-10: 0-321-48096-1
28. Larman, C., Vodde, B.: Practices for Scaling Lean & Agile Development. In: Large, Multisite, and Offshore Product Development with Large-Scale Scrum. Addison-Wesley (2010) ISBN-10: 0-321-63640-6, ISBN-13: 978-0-321-63640-9
29. Leffingwell, D.: Agile Software Requirements. In: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley (2011) ISBN-10: 0-321-63584-1, ISBN-13: 978-0-321-63584-6
30. Cohn, M.: Succeeding with Agile: Software Development using Scrum. Addison-Wesley (2009) ISBN-10: 0-321-57936-4
31. Senge, P.: The Fifth Discipline. The Art & Practice of the Learning Organization. Random House Business Books (2006)
32. Appelo, J.: Management 3.0. Leading Agile Developers, Developing Agile Leaders. Addison-Wesley (2011) ISBN-10: 0-321-71247-1, ISBN-13: 978-0-321-71247-9
33. Kähkönen, T.: Agile Methods for Large Organizations – Building Communities of Practice. In: Proc. Agile Development Conf. (ADC), pp. 2–10 (2004)
34. Bassett, P.G.: Framing Software Reuse: Lessons from the Real World. Yourdon Press Computing Series (1997) ISBN 0-13-327859-X

# Mass-Market Application Development Using Agile Techniques: How Agile Are We Really?

Alberto Heredia, Roberto Esteban-Santiago, Javier Garcia-Guzman, and Antonio de Amescua

Carlos III University of Madrid
Av. de la Universidad, 30, 28911, Leganés, Madrid, Spain
{alberto.heredia,roberto.esteban,javier.garcia,
antonio.amescua}@uc3m.es

**Abstract.** Agile methodologies are a good approach towards mobile application development because of their similar characteristics. While many popular agile methodologies have proven to be effective for developing software, they still need to be adapted to suit the specific needs of the development of mobile apps oriented to mass markets. We present an agile software development process specifically targeted for the development of mass-market applications. The main characteristic of this process is that it includes marketing activities because they are crucial to properly promote a mass-market application. This paper also measures the agility of this process through exploring what we are actually doing on our development team. This process has been followed and refined for over two years in the development of the institutional apps at Carlos III University of Madrid and it is currently in validation.

**Keywords:** Mass-market applications, software development process, agile methodologies, marketing, agility measurement.

## 1 Introduction

Agile methodologies are nowadays crucial within software engineering because one of the goals of any organization is to increase the speed and flexibility in the development of new commercial products [1]. Agile approaches differ from the traditional ones as they put more emphasis on interactions among individuals and adaptive planning than on heavy documentation and predictive planning [2]. In addition, they focus on customer satisfaction through continuous delivery of valuable, quality working software, active participation of customers and stakeholders, and harnessing change for the customer's competitive advantage.

The principles behind this philosophy apply to any discipline that operates in conditions of complexity, uncertainty and change [3]. The characteristics of agile methodologies make them, therefore, appropriate for mobile software development. The development of an app should be relatively quick and straightforward to meet time-to-market demands, involving small teams to produce and deliver frequently new versions adapted to changing customer's needs and expectations [4].

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 259–269, 2013.

Some popular agile methodologies, such as eXtreme Programming [5], Scrum [6] and Kanban [7], have proven to be effective [8]. These agile methodologies, however, attempt to strive for universal solutions as opposed to situation appropriate, so they need to be adapted to suit the needs of different contexts [9]. More specifically, these methodologies do not consider several key aspects for the development of mobile apps oriented to mass markets.

First, mass-market applications are seldom requested by a clearly stated client but they are developed to fulfill the needs of a group of people within society. There will not be only one app but as many potential apps to develop as the number of different needs or opportunities identified. This portfolio of potential apps must be managed therefore to select the best one to be developed.

On the other hand, mass-market applications considered in the scope of this research are not directly delivered to a client who pays the development costs, but they are usually published on a market of mobile apps. Thus, releasing the app implies some additional activities such as defining and implementing a marketing strategy. Moreover, the organizations controlling the markets are also stakeholders to consider because they introduce rules, guides and recommendations that may influence the technical development of the mass-market application.

Regarding feedback, agile methodologies usually acquire feedback prior to the release of the application. Publishing mass-market applications on a market of mobile apps opens new channels to acquire feedback that should be considered, such as the comments that customers write in the market itself or in the social media, for instance.

The most significant aspect that agile methodologies lack of is marketing, which is essential for a mass-market application to be more likely to succeed [10]. Mass-market application development changes the philosophy of traditional software development because the app seldom provides fixed profits from its sale to a particular client but it depends on other different factors such as its price and the number of downloads (i.e., buyers).

Marketing must not be a single process completely isolated from development; market-oriented activities should be efficiently incorporated into the different phases of the project, from the conception of the app to its launch in the market of mobile apps [11]. At the beginning, identifying current trends and competitors is necessary to establish the differential value that makes the app distinguishable from others. Identifying the target audience is also important since benefits will depend on the amount of customers who find the app useful.

A market analysis is also needed for mitigating uncertainties and risks [10] before deciding if the app is developed or not. Knowing the market size, assessing its profitability and comparing the estimated costs to the expected benefits will provide relevant information needed for any decision-making situation in the project. A marketing plan must be also created to publicize the app and make it well-known to public. The plan must set the goals of the marketing campaign; evaluate the strengths, weaknesses, opportunities and threats involved in the project; establish the channels through which the app will be publicize; and estimate the expected results of the marketing effort and the indicators to measure those results.

Some other marketing-related information about the app must be carefully prepared before submitting the app to the market. The screenshots of the app, a keyword list and the description of the app are key elements to sell the app because many customers will decide whether or not to buy the app depending on this information. Finally, once the app is published on the market, feedback provided by users about the developed app and data about downloads and application use must be gathered to analyze if the app is meeting the goals previously set in the marketing plan.

To conclude this section, we have presented some aspects that agile methodologies lack of when used for development of mass-market applications. To fill this gap, a software development process based on agile techniques was tailored specifically to develop this kind of applications. This process is briefly described in the following section. Then, the agility of this software development process is measured. Finally, some conclusions are drawn.

## 2 miSEL Software Development Process (miSEL-sdp)

This section describes a process based on agile methodologies specifically adapted to the development of mass-market applications named miSEL-sdp. Figure 1 shows the flow through the different processes considered in this proposal, which are briefly described below.

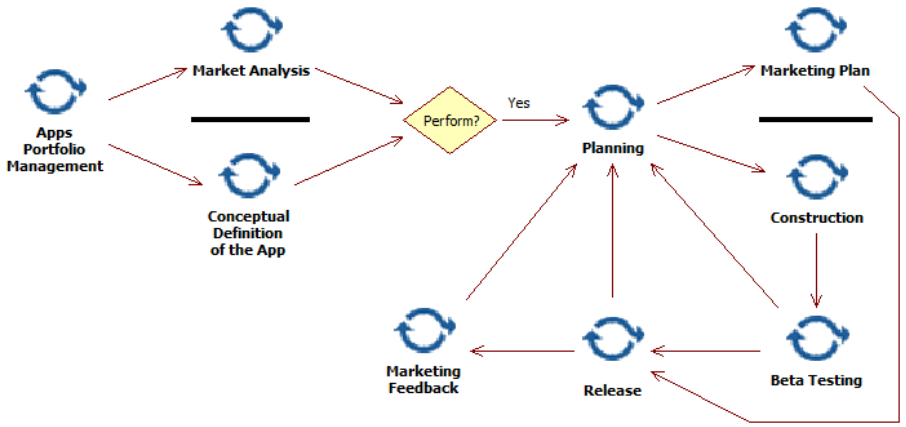



**Fig. 1.** Processes defined in miSEL-sdp

*Apps Portfolio Management* aims at selecting the app to be developed. To do so, the project team must identify customer's needs or opportunities that arise. A portfolio of apps that may satisfy those needs or take advantage of those opportunities is established, providing an overview of all the potential apps to develop. This list of apps must be prioritized and sorted in descending order to better identify the top-priority apps. The team should select the app with the highest priority for its development.

The objectives of the *Conceptual Definition of the App* are to define the features and restrictions of the app and to make a first estimate of the effort and cost of the project for future planning. The first activity within this process consists in the identification of the characteristics of the mass-market application to develop, which should represent something unique, provide fresh functionalities and be aligned with current trends. Identifying the target audience is also a very important activity in the process, and to do so, segmenting the market could be helpful. On the other hand, both direct and indirect competitors should be also identified because we can learn from them. Finally, the project team must estimate the effort and cost for the development, and also for implementing the forthcoming marketing strategies. This estimation is crucial because it will help decide whether or not the app is going to be developed.

The *Market Analysis* is carried out in parallel with the *Conceptual Definition of the App* and it determines whether the market is sufficiently attractive to launch the app. In this process, the size and the growth rate of the market in which the app will be distributed are first identified. Then, the profitability of the market must be assessed by identifying potential buyers as well as how much they will be willing to pay for the app, how much it will cost to produce the app, and why they will want to buy our app and not the competitors' ones; advertising revenues should also be considered if the app is intended to be free and ad supported. A cost analysis is performed later comparing all the monetary costs that will occur during the development of the app with the benefits that the app will bring.

*Planning* should be based on the features and restrictions previously identified, which will help create a list of tasks to be carried out by the different roles involved in the project. Every task has to be estimated in terms of time and cost, and these estimations should be compared to the ones obtained in the *Conceptual Definition of the App* process for the whole project; some tasks may be re-estimated in consequence. To conclude this process, each team member willingly accepts the responsibility of achieving the tasks he/she thinks he/she can do better, so that the project manager do not imposes tasks on individuals.

The *Construction* of the app is an iterative cycle in which the app is first designed, then coded and tested to validate the requirements, and finally test results are analyzed to provide feedback. If the feedback is positive, this process finishes and a beta version of the app is ready to be validated by the stakeholders; otherwise, the cycle starts again with a re-design to fix the issues found in the tests.

The objective of the *Marketing Plan* is to specify a plan to publicize the app. First, the marketing manager must set the goals to achieve, considering not only the objectives of the app, but also objectives related to sales, app price and earnings. On the other hand, a SWOT Analysis must examine the strengths, weaknesses, opportunities and threats of the app in comparison to similar ones to obtain relevant information needed for any decision-making situation in the project. Another activity included in this process is the establishment of different channels to publicize the app, which should include a website and its promotion through social networks. Later, expected results of the marketing effort must be estimated. Finally, metrics must be determined to monitor the marketing campaign.

The *Beta Testing* aims at finding defects missed in the tests executed by the development team in the *Construction* process. The stakeholders must validate the beta version of the app is fully functional and includes all the features requested. In case a stakeholder does not agree with any of the features, a change in the features may be proposed indicating the new desired functionality. If and only if the stakeholders provide satisfactory results from their validation, the result of this process will be an app ready to be released on the selected market of mobile apps.

The *Release* makes the app available in the market for customers to buy it and thus obtain the expected benefits. Some elements are required to distribute the app in the market, such as screenshots of the app, a keyword list, a description of the app, and the packaged app. In addition, markets usually perform a validation process to assure that the app meets the criteria established for publishing apps on that market. If the app does not pass the market validation, the project team will receive feedback from the market reviewers regarding the issues found which made the app being rejected.

Finally, the *Marketing Feedback* gathers the users' opinion about their experience with the app, besides other relevant data, to analyze them. This feedback could come from the comments that users write in the market itself, but it could come also from other sources such as comments in social media, for instance. The marketing manager has to check if the marketing plan was followed properly according to this information. The analysis of the feedback will be also useful to identify issues to fix in the current version of the app or to define new functionalities that may be included in future versions. Hence, that information should go back to the *Planning* process.

## 3 Experience with miSEL-sdp

miSEL-sdp has been followed by the miSEL research cluster at Carlos III University of Madrid (UC3M) for two years. UC3M is a Spanish public university located in four different campuses in and around the city of Madrid. UC3M has almost 19,000 students and receives around 1,000 foreign students each semester. Almost 2,000 people work in teaching or research, and almost 1,000 people are part of the administrative and service staff at UC3M. Considering these numbers, those people compose a mass market for applications related to university life in the scope of this research.

A multidisciplinary team was created at UC3M to research on mobile applications and technologies to improve the quality of university life. A close collaboration was established among miSEL research cluster, the Office of Computing and Communications Services (SdIC), the Office of the Vice-President for Culture and Communication, and the Office of the Vice-President for Infrastructures and Environmental Affairs.

The use of miSEL-sdp has led to the successful development of several mass-market applications, including some of the official apps launched by UC3M. The set of official apps is mainly composed of an institutional app, a map-based app and an app for events. The institutional application provides information about the university, studies offered, means of transport, faculty and student directory, news, and videos of events. The map application locates classrooms, faculty offices and other relevant places. The last application provides information about the different events and activities promoted by UC3M. Other applications developed following miSEL-sdp were an app to provide information to conference attendees for the VII International Congress on IT Governance and Service Management (ITGSM), an app for the Computer Science and New Technologies Fair (T3CHFEST) that took place at UC3M, and some other apps developed by undergraduates for their final-year projects. These apps are available for Android and iOS devices in Google Play and App Store, respectively.

The set of official apps launched by UC3M was very well received by students, faculty and staff. Apps are currently rated over 3.3 on a 1-to-5 scale in Google Play and over 4 on a 1-to-5 scale in App Store, being UC3M maps app the highest rated with 4.5 stars. The first version of the official apps hit a total of 3,629 downloads from App Store and 2,230 downloads from Google Play in the first month. A year after the first release, the official apps have been downloaded more than 13,600 times from App Store and more than 11,600 times from Google Play. In addition, 2,251 iOS users recently updated the UC3M institutional app to the newest version, 912 users updated the UC3M maps app, and 923 users updated the UC3M events app. Regarding Android, 1,672 users have the UC3M institutional app currently installed, 632 active users for UC3M maps app, and 539 active users for UC3M events app.

## 4 How Agile Are You? The Initiative

The characteristics of all agile methodologies are based on the values and principles of the Agile Manifesto [2]. Unfortunately there are no official criteria for determining whether or not a project team is taking an agile approach [12], so teams may claim to be agile when they are not, and vice versa.

There are, however, some initiatives like the one promoted by Scott Ambler and Associates [13] who perform surveys under the title "How Agile Are You?" that try to explore the issue of how agile project teams which claim to be agile actually are. Based on observations, Ambler proposes five criteria [12] he has used for several years:

- Does the team regularly produce value for their stakeholders?
- Does the team validate their work to the best of their ability?
- Are stakeholders actively involved?
- Is the team self-organizing?
- Does the team strive to improve their process?

The first question is directly related to one of the 12 principles behind the Agile Manifesto and tries to evaluate if the team provides a consumable solution which provides value to their stakeholders. It brings the idea that the agile teams produce far more than just working software, they also deliver updates to hardware, they often change the business process around the usage of the system, and they may even impact the organization structure of the end users.

The second question addresses the adoption of various validation strategies by agile teams. Agile developers must test their work to the best of their ability, running continuous regression tests and better yet taking a Test-Driven Development (TDD) approach. Moreover, they may work with independent test teams, particularly in complex situations or in regulatory environments.

Regarding the third question, agile teams should work closely with their stakeholders, or a representative of them. Stakeholders should be involved in the development process prioritizing requirements, providing information and making decisions ideally on a daily basis.

The fourth question evaluates whether or not the team is self-organized, which means that the team members themselves plan, estimate and coordinate their own work. It does not mean that agile teams are out of control but they should also work within an appropriate governance framework to help guide and monitor their efforts.

The last question addresses the continuous improvement of the process. Agile teams must regularly reflect on how they work together to identify potential ways to improve their software process. Agile teams should also make the effort to track their progress over time and share their ideas with other teams.

## 5 Measuring the Agility of miSEL-sdp

Considering the five criteria presented in the previously, the objective of this section is to assess the agility of miSEL-sdp through exploring what we are actually doing on our development team.

### 5.1 Does the Team Regularly Produce Value for Their Stakeholders?

Table 1 summarizes the different aspects considered to analyze how is our team providing value to stakeholders, each one classified depending on whether our development team complies with it or not.

**Table 1.** How is our team providing value to stakeholders?

| Yes | No |
|---|---|
| Identify stakeholders and goals | Project inception |
| Producing working software | Personnel changes |
| Regular stakeholder discussions | |
| Definition of done | |
| Business process improvement | |
| Consider usability | |
| Supporting/Project documentation | |

At the beginning of the project, our stakeholders are identified so they help us to identify needs and opportunities that arise, and they also identify and prioritize potential apps. As it was shown in Figure 1, we periodically produce a consumable solution which is delivered to the stakeholders in the Beta Testing process to validate if it provides the expected value to them. Meetings are often held with stakeholders to discuss new features or improvements in the app as a consequence of this beta testing, or to manage the portfolio of apps. If and only if the stakeholders provide satisfactory results from their validation we consider the work as done. Opportunities to improve the business process also arise as a result from these meetings. Finally, we take into account usability because it is critical when developing mobile apps, as well as producing user manuals when they are needed. On the other hand, we do not have a project inception phase because we deliver a basic working solution since the first iteration. We did not have significant personnel changes as a result of our projects.

### 5.2 Does the Team Validate Their Work to the Best of Their Ability?

Table 2 summarizes the different aspects considered to analyze how is our team validating its own work, each one classified depending on whether our development team complies with it or not.

**Table 2.** How is our team validating its own work?

| Yes | No |
|---|---|
| Developer regression testing | Developer TDD |
| Iteration demos | Static-code analysis |
| Acceptance TDD | Dynamic code analysis |
| All-hands demos | Parallel independent testing |
| Non-solo development | End-of-lifecycle testing |
| External reviews | |

Our development team usually carries out its own regression and system testing during the construction of the app. With regard to demos, they are validated every iteration not only by the stakeholders but also by a selected group of final users. Techniques such as pair programming are sometimes used during the development and other software engineers external to the development team occasionally review our work.

The TDD approach we follow is taken at the requirements level via system tests, but not usually at the design level via unit testing. Furthermore, we do not include neither static nor dynamic code analysis in our builds. All tests are carried out by the team itself, so there is not an independent team to perform testing in parallel to development.

### 5.3 Are Stakeholders Actively Involved?

Table 3 summarizes the different aspects considered to analyze how is our team working with stakeholders, each one classified depending on whether our development team complies with it or not. Our team works closely with our stakeholders. Not only they prioritize requirements but they are usually at hand to help the team to solve problems. A Product Owner is usually part of our development team and is responsible for the first communication with the stakeholders and the requirements envisioning. A demo sandbox is sometimes deployed to allow stakeholders to validate some requirements working with an interim version of the system. Finally, stakeholders work with business analysts who provide requirements to the Product Owner, but not to our team directly.

**Table 3.** How is our team working with stakeholders?

| Yes | No |
|---|---|
| Iteration demos | Requirements specifications |
| Product Owner | Business analysts to developers |
| Requirements envisioning | |
| As needed | |
| Daily access | |
| Business analysts to Product Owner | |
| Demo Sandbox | |

Our stakeholders do not usually detail requirements in a document, although minutes of the meetings with stakeholders are always written to keep track of all the agreements. Additionally, the project team always designs paper prototypes of user interfaces that help developers and stakeholders to agree on the requirements.

### 5.4 Is the Team Self-organizing?

Table 4 summarizes the different aspects considered to analyze how is our team organizing its work, each one classified depending on whether our development team complies with it or not.

**Table 4.** How is our team organizing its work?

| Yes | No |
|---|---|
| Iteration planning | Standards identified |
| Product Owner prioritizes | Standards followed |
| Daily standups | |
| Work assigned | |
| Senior management attends | |
| Senior management helps | |
| Infrastructure identified | |
| Infrastructure leveraged | |
| Project dashboard | |
| Manual reports | |
| Status reports | |

As it is show in Figure 1, at the beginning of an iteration we hold a planning meeting in which each one of the team members takes charge of different tasks needed to meet the requirements previously prioritized by the Product Owner. Our team also holds daily stand-up meetings to coordinate the activities and solve the problems as soon as possible. Two senior managers provide support for the team in meeting objectives and they often attend the stand-up meetings. We have a common technical infrastructure that allows us to manage the tasks, the source code and the documentation from the different project, and it provides status reports when needed. Finally, a dashboard is also used to track the status of the different tasks.

On the other hand, our team does not follow explicit programming standards or data conventions. The programmers generally use face-to-face communication to define user interface conventions.

### 5.5 Does the Team Strive to Improve Their Process?

Table 5 summarizes the different aspects considered to analyze how is our team improving the process, each one classified depending on whether our development team complies with it or not.

**Table 5.** How is our team improving the process?

| Yes | No |
|---|---|
| Active improvement | As-needed retrospectives |
| Iteration retrospectives | External audit |
| Post-mortem reviews | |
| Measured improvement | |

Our process is not static but it has continually evolved since its first definition. The whole team and specially the senior managers actively think of new ways to improve the process. Retrospective sessions and post-mortem reviews help us to identify potential improvements. We also assess and track our progress when adopting improvements to our process in order to verify if they were effective.

Finally, external auditors do not review our work during the project. It would be interesting, however, to externally audit our process to identify potential improvements from a different point of view.

## 6 Conclusions

Mobile app markets have shown an impressive growth during the last year. The increasing number of apps available on markets and the revenues that developers and companies get is relevant enough to seriously consider the way apps are developed. Agile methodologies are a good solution for mobile application development, which requires quick delivery of quality software and effective responses to changes in requirements, among other things. Although agile values and principles have proven to be useful to develop mobile apps, agile methodologies still need to be adapted to suit the specific needs of the development of mass-market applications.

This paper briefly describes an agile mass-market application development process called miSEL-sdp, which also includes marketing activities as they are necessary to publicize a mass-market application. This process has been followed and refined for over two years and its use has led to the successful development of several mass-market applications, including some of the official apps launched by Carlos III University of Madrid. As a first step towards the validation of miSEL-sdp, this paper tries to measure the agility of this process through exploring what we are actually doing on our development team. Our future research will focus on a more quantitative evaluation of the agility of miSEL-sdp. A stable and common accepted assessment model would be helpful to do so, as currently there is no agreement in the criteria to use for assessing the agility of a process or an organization.

## References

1. Colomo-Palacios, R., González-Carrasco, I., López-Cuadrado, J.L., García-Crespo, A.: ReSySTER: A hybrid recommender system for Scrum team roles based on fuzzy and rough sets. International Journal of Applied Mathematics and Computer Science 22(4), 801–816 (2012)
2. Alliance, A.: Manifesto for Agile Software Development (2001), `http://agilemanifesto.org` (last access: April 22, 2013) (retrieved)
3. Sorofman, J.: Agile Isn't Just for Geeks Anymore. Gartner, Inc. (2013), `http://blogs.gartner.com/jake-sorofman/agile-isnt-just-for-geeks-anymore` (last access: April 22, 2013) (retrieved)
4. Abrahamsson, P.: Agile Software Development of Mobile Information Systems. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 1–4. Springer, Heidelberg (2007)
5. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Boston (1999)
6. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall, Upper Saddle River (2002)
7. Anderson, D.: Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, Sequim (2010)
8. Dyba, T., Dingsoyr, T.: Empirical studies of agile software development: a systematic review. Information and Software Technology 50(9-10), 833–859 (2008)
9. Cao, L., Mohan, K., Xu, P., Ramesh, B.: A framework for adapting agile development methodologies. European Journal of Information Systems 18(4), 332–343 (2009)
10. Rahimian, V., Ramsin, R.: Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach. In: 2nd IEEE International Conference on Research Challenges in Information Science Symposium, pp. 337–342. IEEE Press, New York (2008)
11. Hughes, J.: iPhone and iPad Apps Marketing. Que Publishing, Indianapolis (2010)
12. Ambysoft. The Criteria for Determining Whether a Team is Agile (2013), `http://www.agilemodeling.com/essays/agileCriteria.htm` (last access: April 22, 2013) (retrieved)
13. Ambysoft. Surveys Exploring The Current State of Information Technology Practices (2013), `http://www.ambysoft.com/surveys` (last access: April 22, 2013) (retrieved)

# INCUVA: A Meta-framework for Sustaining the Value of Innovation in Multi-cultural Settings

Mohamed Sheriff[1], Elli Georgiadou[2], Geetha Abeysinghe[3], and Kerstin Siakas[4]

[1,2,3] Middlesex University, School of Science and Technology,
The Borroughs, London NW4 4BT, UK
{m.sheriff,e.georgiadou,g.abeysinghe}@mdx.ac.uk
[4]Alexander Technological Educational Institute of Thessaloniki,
Department of Informatics, P.O. Box, 141, GR-57400 Thessaloniki, Greece
siaka@it.teithe.gr

**Abstract.** Innovation is increasingly seen as the main vehicle for value creation in business organisations as well as in civic societies. Consequently, funding agencies and businesses continue to invest huge amounts of money and other resources on innovative projects with a view to creating and sustaining the desired value outcomes. However, creating value is not the same as sharing and sustaining value, especially when such value needs to be shared and sustained in a multicultural space. This paper analyses the interaction of three key elements, namely Innovation, Culture and Value, that could facilitate or inhibit the sharing and sustenance of value created through innovative projects. The paper also proposes a meta–framework based on work done during the EU-funded project VALO, aimed at developing and disseminating effective ways of maximising and sustaining the value created in innovative projects.

## 1 Introduction

In a competitive world the need to innovate often underpins the development strategies of most business organisations and even states and nations. Modern organisations thus expend relentless effort, time and resources on innovative ventures and projects in order to gain advantage over competitors. Implicit in this drive is the assumption that innovation is positively correlated with value creation, the ultimate goal of every business organisation, if not every human activity. While there may be only mixed evidence of the veracity of this assumption, the trend in the Information Systems and Competitive Advantage literature suggests that many organisations see innovation as the Holy Grail for creating value and gaining competitive advantage. For example Allee and Taug [1] report that "*research shows that in order to sustain competitive strength and continued growth Western companies need to build innovation into their cultures and structures as an essential condition for value creation*". They further suggest that "*it is critical to competitiveness to get a clear assessment of innovation capabilities…*". Phillips [2] drawing on SanJay Dalal's innovation index, claims that "*innovative firms derive significantly more value in terms of market value and shareholder value than other firms in the same industries*". The positive link between innovation and business value, at least in the minds of

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 270–281, 2013.

business executives, is hence reasonably well established. Nonetheless, in practice, the value attributable to many innovative projects tends to run out in concert with the allocated funding. The value created is thus not sustained beyond the lifetime of the projects. Effective innovation should not only facilitate the creation of value but should also ensure that such value is sustained and shared to its optimum potential.

This is obviously a challenge to funding agencies/sponsors as well as beneficiaries of the myriads of innovative projects around the world. The European Union recently responded to this challenge by funding a specific 'Valorisation' project – here after called VALO.*'Valorisation is defined as the process of sustaining value created through innovation and hence optimising its impact among the direct and indirect beneficiaries'*

The rationale for the VALO project is derived from the observation that many projects are often carried out in isolation and hence they provide very little or no lasting impact. These projects tend not to imagine and plan for continuing dissemination and exploitation of their results and deliverables beyond the allocated funding period; and even when they do, there is little evidence that they succeed. The impact/benefit of a sustainable project translates into added value gained by a diverse group of stakeholders and/or specific target groups well beyond the lifespan of the project. All projects therefore, need to valorise their results for maximising achievements and increasing sustainability after their lifetime.

However, there are many obstacles to successful valorisation, most of which relate to lack of understanding of the complex nature of the stakeholders and their value expectations as well as the cultural diversities which often lead to misunderstandings and conflict. The key obstacles to valorisation seem to stem from the intricate relationships between Innovation, Culture and Value. Our starting position in this paper is that these relationships constitute a complex mix of opportunities and obstacles that govern the purposes, processes and outcomes of most projects that take place in a multicultural space. Our objective is to investigate and analyse these relationships in order to:

- Provide new insights into the nature and extent of the common obstacles that often inhibit the creation and sustenance of value in innovative ventures;
- Identify and highlight the opportunities for creating, disseminating and sustaining the value derived from innovative projects;
- Propose a meta-framework for identifying, disseminating and sustaining, i.e. valorising, the positive outcomes of innovation ventures in multicultural settings.

This is a position paper which is reporting this model-based research findings using the qualitative argumentation research method. [3] specifies that qualitative argumentation research is "Creative research that is reliant on viewpoint and speculation rather than observation. It is beneficial in constructing theory that can be repeated for testing". According to [4] qualitative argumentation is used to analyse the behaviour of complex systems. In particular it is "recognised as a means for resolving issues of belief in situations characterised by incomplete, uncertain, inconsistent, and imprecise knowledge".

The rest of this paper will proceed as follows: First we outline the key features of the three main elements of interest in this study: Innovation, Culture and Value; and explore their relationships and interactions (see Figure 1). In particular, we highlight the synergies and tensions that arise at the interplay between Innovation and Culture, Innovation and Value and Culture and Value. We next analyse these synergies and tensions with a view to finding ways to positively exploit the synergies and to mitigate the tensions during the valorisation of project findings. The paper concludes by proposing a meta-framework as a guide for sustaining the value derived from innovative projects in multicultural settings.

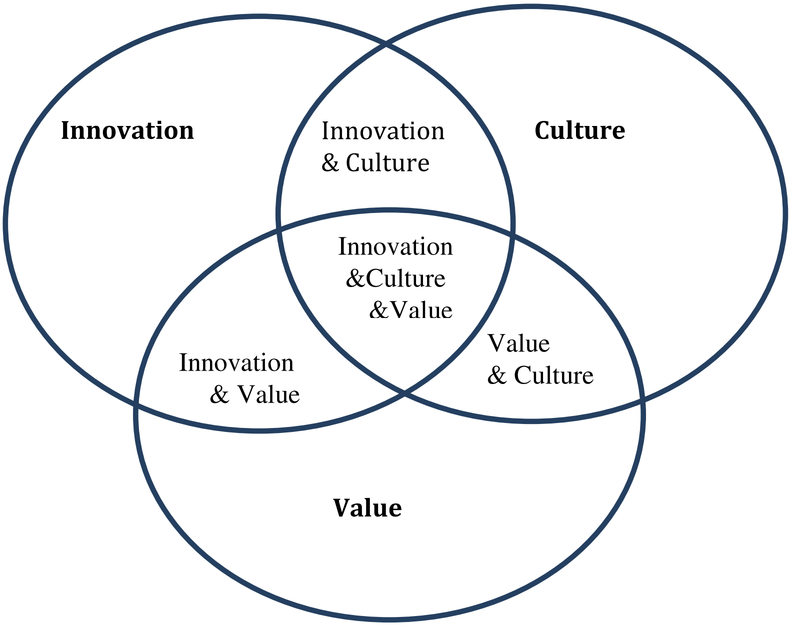



**Fig. 1.** Conceptual framework relating Innovation, Culture and Value

## 2 Innovation

Researchers suggest different definitions/ explanations for the term innovation. [5] states that *"Innovation is not a single action but a total process of interrelated sub processes. It is not just the conception of a new idea, nor the invention of a new device, nor the development of a new market. The process is all these things acting in an integrated fashion. Innovation is the management of all the activities involved in the process of idea generation, technology development, manufacturing and marketing of a new (or improved) product or manufacturing process or equipment*".

[6] suggest:*"Innovation can be defined as the application of new ideas to the products, processes, or other aspects of the activities of a firm that lead to increased value"*. It seems that the degree of novelty and unique offerings in products or services, and/or in processes (faster, lower cost, customisation) is decisive for value creation of innovation. [7] argues that innovation is an invention implemented and

taken to market, hence bringing strategic and competitive advantage. Complexity (offering something that others find difficult to master), legal protection of intellectual property rights (others need to pay licence), timing (first-mover or fast follower advantage), robust design (platforms others can build on), rewriting the rules (different ways of doing things – old ones redundant) and reconfiguring the parts (rethinking how bits of systems work together) can also be considered innovations providing strategic advantage and value [8]. It is not always the innovation or the technology in itself that matters, but innovation-in-use. Hence for innovation to achieve its optimum potential, it must be deployed to ensure that it reaches all that could benefit from it.

## 3 Culture

Culture is a very elusive concept hence the many attempts by researchers to understand and define culture. [9] reported: “In 1952 [10] found 164 different definitions, with different meanings for the concept of culture. Culture has different layers or levels often overlapping”. Lee and Varey [11] drawing on Schein [12] and Robey and Azevedo [13] classify culture into the following three levels: The deepest level consists of patterns of assumptions that organisational members hold without awareness. The intermediate level refers to the values and beliefs of organisational members, which are readily articulated by members in their normative statements. The surface level is concerned with the organisation’s symbols and artifacts, its routines and practices.

Culture can also be classified according to the element of interest. Culture viewed at societal/national level would be different from, even if overlapping with, organisational culture. It is also not uncommon to identify sub-cultures within and across national/societal boundaries, such as cultures of specific industries, business practices and professional groupings.

Organisational culture is defined as a set of assumptions, beliefs, attitudes and values that are shared by existing members and taught to new members of an organisation. It includes prevailing patterns of behaviour and sentiments. Organisational culture is mainly created and maintained in existing frameworks by the founders and the leaders of an organisation through their value system. [12] identified three levels of organisation culture that refer to the degree of visibility to the observer, which can operate at the same time:

- artifacts and behaviours: any tangible identifiable elements or objects in an organisation, such as structures and dress code.
- espoused values: organisation’s stated values and rules of behaviours, such as mission statement and strategy
- basic assumptions and beliefs: deeply embedded, taken-for-granted behaviours (usually subconscious)

The first level, artifacts and behaviours, are easy for an outsider to recognise but hard to understand. Espoused values – the second level defines how things ‘should’ be as compared to how they are at present. A strong organisational culture will promote what is considered the appropriate behaviour in response to particular circumstances. Basic Assumptions and Beliefs – this level defines the cultural foundation, it guides and constructs the organisational reality.

Strong business cultures will have all three levels present. The more members of the workforce who accept, use and talk the language of the culture the stronger it will be. In doing so this forcefully influences the attitudes and actions of the organisation as a whole. In the dynamically changing environment organisations are facing today innovation and creativity is increasingly becoming a way of life. For continued competition organisations need to come up with structures where innovation and creativity is accepted as part of the culture, especially when the current organisational culture may create a conflict with the demand for innovation.

## 4 Value

Value is as universal a concept as it is difficult to define. A variety of definitions and characterisations could be found in literature from disciplines ranging from Axiology to Zen philosophy. The original economic notion of value, as portrayed by Adam Smith, is now increasingly recognised as only one of the many notions of value [14]. The conception of value as the worth of an object, either to an individual or as determined in exchange for another object, constitutes only one perspective of value. Other conceptions of value tend to be more fluid and indefinite and often highly context dependent.

The question of where value lies in a subject – object interaction constitutes one of the main difficulties in determining an unequivocal definition of value. [15] suggests that the value concept has been employed in two distinctly different ways: one attributes value to objects as a function of their properties and the other associates value with human subjects. [16] on the other hand attributes value to the interaction between a subject and an object as illustrated in the following definition. Value is "*affectivity occurring in the relational contexture determined by the reaction of an organism to a stimulus object*". Value is also often characterised as either instrumental or terminal, where instrumental value represents a means to an end while terminal value relates to end results. It is worth noting that instrumental value (the means to an end) does not always faithfully translate into terminal value (the desired or realised end).

From an individual user perspective [17] proposed three forms of value, namely, Conceived value, Operative value and Object value. Conceived value describes the user's projection of the potential benefit they might derive from an object, Operative value refers to the extent to which a user likes or dislikes the content or process of use of an object or service, and Object value approximates to the quality of the object or service, or what it affords the user by virtue of its characteristics. Insight into these forms of value should help us better understand user behaviour in various situations. For example, low operative value may result in users avoiding or underutilising specific objects. Low conceived value may cause users to exert less effort and care since they do not believe much will come out of their effort, and low object value suggests that the available object may be inappropriate for the task at hand. Depending on the situation, such an object could be underrated, underutilised or completely ignored.

Sinden and Worrell [18] combined the various perspectives on the nature and sources of value to propose that value is a function of utility, environmental conditions and circumstance of evaluator at the time of evaluation, which can be expressed as follows: **Value = f (Utility, Environmental conditions, Circumstance of evaluator at time of valuation)**. Since utility is often largely a function of the inherent characteristics of objects, this definition could be seen as a combination of the objective, subjective and interactive conceptualisations of value. Value can thus be said to be largely context-mediated.

The key features of value from the various conceptualisations can be summarised as follows:

- Value is a complex and multi-dimensional phenomenon
- Value can be instrumental and/or terminal
- Value is often temporal or time variant
- The sources of value include: the objective characteristics of an object, the subjective disposition or opinion of a subject (perceiver) and the interaction between a subject and an object
- Value manifestations are largely context dependent
- Value manifestation at individual user level is often in the form of Conceived value, Operative value, Object value or a combination of these forms of value.

## 5 Synergies and Tensions

We can classify the outcomes of the interactions between Innovation, Culture and Value into two main categories, namely: those that facilitate the creation of opportunities and those that present obstacles to valorisation. In this section we discuss the specific interactions between Innovation and Culture, Innovation and Value, and Culture and Value and categorise them as either sources of opportunity for valorisation or sources of tension that needs to be mitigated. Through this study we ultimately aim to encapsulate our understanding of these interactions into a meta-framework which can be used by organisations for maximising the value from their innovations through capitalising on the synergistic factors and mitigating the factors causing tension.

### 5.1 Innovation and Culture

Herbig and Dunphy [19] explored the relationship between culture and innovation. They identified that existing cultural conditions determine the way in which innovations are adopted. They suggest that cultures, where status is given to entrepreneurial efforts, are more innovative, and that cultures where creativity, technical ability and higher education are valued, are more successful at adopting innovations. They also investigated the relative importance of an individual versus a group and suggest that cultures emphasising individualism and freedom are more likely to be creative and to benefit more from innovative ideas.

There are a number of organisational factors that support or encourage innovation. Martins and Terblanche [20] have synthesised most of these in their article, '*Building organisational culture that stimulates creativity and innovation'*. They proposed five determinants of organisational culture which can have an influence on the degree to which innovation take place in the organisation: strategy (vision and mission, and purposefulness); structure (flexibility, freedom, cooperative teams and group interaction); support mechanisms (reward and recognition, and availability of resources); behaviour that encourages innovation (mistake handling, idea generating, continuous learning culture, risk taking, competitiveness, support for change, and conflict handling); and open communication. [21] point out that there has been little research on '*work environmental factors that may undermine creativity*'. The authors use the term '*impediments*' to refer to such factors. [21] synthesises that the major factor for organisational impediments for creativity and innovation is control, which include: "control in decision making, control of information flow, or even perceived control in the form of reward systems that put too much emphasis on increasing extrinsic motivation".

## 5.2 Innovation and Value

It is generally accepted that the main object of innovation is to create value. However, value manifestations are often complex, temporal and highly context dependent. To create and sustain value, the innovator must not only appreciate the complexity of value, they must also cater for the time-variant and context dependent user conceptions of value. Two key elements of that context are time and culture. Hansen & Birkinshaw [22] describe the innovation value chain as comprising three key stages: Idea generation, conversion and diffusion. Each of these stages often involves different but interrelated value considerations. Idea generation is largely guided by conceived value i.e. what the innovator believes would be of value in future. The conversion stage would focus on object value while the success of the diffusion stage is often determined by operative value i.e. the extent to which the beneficiaries like the innovated object or service. The key challenge is delivering value across various cultures and sustaining that value over time. The starting point is to first understand the nature of that value and how it plays out across various cultures.

## 5.3 Culture and Value

Despite the complex manifestations of value it is evident from the above discussion that different national and organisational cultures have different value expectations. Cultural predispositions as well as context may hinder or facilitate the creation of value [23]. Cultures (both national and organisational) characterised by collectivism and mutual aid and reciprocity are likely to generate a synergistic environment which in turn maximises the chances of creating conceived and object value. Cultures characterised by individualism, masculinity and short termism may hinder effective value creation. However, "*Cultures today are in general characterised by hybridisation. For every culture, all other cultures have tendencially come to be* *inner-content or satellites. This applies on the levels of population, merchandise and information*" [24]. It is in this context that we need to be aware of possible tensions and certainly be aware also of synergistic elements that enable diverse cultural groups and settings to create and share value.

Value perceptions across cultures vary but increasingly it is recognised that we all live in multi-cultural environments both within the national space and within organisations, even within teams and project consortia. Value creation typically requires resources beyond a two-party system, often involving a firm, its customers, suppliers, employees, stakeholders, and other network partners. [25] propose this value co-creation paradigm based on experience across all stakeholders.

Diversity could enrich the experience and open avenues of fruitful co-operation and value creation and co-creation. A sharing, co-operative culture creates synergy as expressed in the co-operative movement slogan "*All for one and one for all*". Consultative, nurturing and feminist (according to [26]) cultures are co-operative and beneficial in the long term. Monolithic cultures (e.g. Patriarchy) are likely to inhibit innovation, create tensions, resentment and an atmosphere of conflict as different stakeholders are likely to have opposing views of value and value expectations.

## 5.4 Innovation, Culture and Value

Table 1 depicts the main characteristics of Innovation, Culture and Value shown along the leading diagonal. The synergistic aspects can be seen at the bottom of the diagonal whilst the aspects that are likely to create tensions are shown above the diagonal.

**Table 1.** Characteristics, Tensions, and Synergies

| | ***Innovation*** | ***Culture*** | ***Value*** |
|---|---|---|---|
| Innovations | ***Innovation:*** *Application of new ideas to the products, processes, or other aspects of the activities of a firm that lead to increased value* | Control in decision making, control of information flow, or even perceived control in the form of reward systems<br>Cultural clashes<br>Culture induced stress | Non-alignment of value perceptions<br>Temporality of value<br>Conceived value does not always equate to Object and/Operative value<br>Short term orientation |
| Culture | Supportive structures with cooperative teams and group interaction.<br>Clearly specified cross-cultural goals<br>Support mechanisms such as appropriate reward and recognition schemes.<br>Behaviour that encourages innovation (mistake handling, continuous learning culture, open communication, competitiveness, support for change, encouragement of risk taking and idea generation. Freedom and autonomy) | ***Culture:*** *A set of assumptions, beliefs, attitudes and values that are shared by existing members and taught to new members of a group (e.g. organisation)* | Conflicting perceptions of value.<br>Cultural idiosyncrasies<br>Past Orientation |

**Table 1.** (*continued*)

| | | | |
|---|---|---|---|
| Value | Innovator's appreciation of the complexity of value<br><br>Catering for the time-variant and context dependent user conceptions of value<br><br>Alignment between the innovator's perception of value and that of the beneficiaries<br><br>The extent to which the beneficiaries like and hence utilise the innovated object or service | Cultures characterised by collectivism, mutual aid, reciprocity, and Diversity<br><br>Future Orientation<br><br>Doing rather than being culture | ***Value:*** *A function of utility, environmental conditions and circumstance of evaluator at the time of evaluation* |

## 6 INCUVA: A Meta-Framework for Valorisation

The ultimate object of innovation is to create and sustain value, preferably across various cultures. A meta-framework for valorisation should therefore comprise (1) defining and understanding value (2) determining the potential value manifestations (3) Understanding the diverse cultural settings in which the innovation would be used and (4) developing and adopting effective dissemination strategies and tools to optimise the value of the innovation. Table 2 outlines the key dimensions of the INCUVA meta-framework.

## 7 Conclusion and Further Work

Bearing in mind the complexities of both culture and innovation we consider the challenges presented in multicultural settings when we attempt to inform about and communicate process and product innovations. This research arose from the VALO project which realised the need to provide support for sustainability and exploitation of innovations. There are a numbers of organisational factors which can inhibit or encourage the creative capabilities of individuals, teams and/or groups. Any organisation which aims to build competitive strategies based on their ability to innovate needs to be aware of these factors. The perception of value, both of the innovator and of the conceiver has an impact on the successful adoption of innovation. The relationship between innovation, culture and value is complex. In order to achieve a sustainable advantage, an organisation needs to look at both tensions and synergies that arise due to the three two way relations: culture ↔ value, culture ↔ innovation, and innovation ↔ value. In this paper we represented these interactions in a conceptual framework (Figure1), explored the synergies and potential tensions arising from the diverse attitudes to value (Table 1) which are underpinned by cultural diversities. We proposed the INCUVA meta-framework (Table 2) which organisations, groups, and project consortia could use in order to formulate strategies,

**Table 2.** The key dimensions of INCUVA Meta framework

| | | |
|---|---|---|
| 1.0 Define Innovation focus | 1.1 Adopt Innovation strategies<br><br>1.2 Develop awareness of Intellectual Property Issues<br><br>1.3 Identify Key Performance Indicators (KPIs) | 1.1.1 Internal innovation policy<br>1.1.2 External Innovation policy<br><br>1.2.1 Copyright<br>1.2.2 Patents<br>1.2.3 Conflict between openness and exploitation<br>1.3.1 Define targets (short, medium and long term)<br>1.3.2 Collect and Analyse Measurements |
| 2.0 Define value focus | 2.1 Exchange value<br><br>2.2 Value-in-use | 2.1.1 Terminal value<br>2.1.2 Instrumental value<br>2.1.3 Timeliness & time span |
| 3.0 Project potential value manifestations | 3.1 Identify possible synergies<br>3.2 Highlight and analyse probable tensions | 3.1.1 Reinforce synergies<br>3.2.1 Mitigate probable tensions |
| 4.0 Appreciate various sources of value | 4.1 Object properties<br>4.2 User perceptions/ disposition<br>4.3 Object-user interaction situations | 4.1.1 Object value<br>4.2.1 Conceived value<br>4.3.1 Operative value |
| 5.0 Develop an understanding of the cultural setting | 5.1 National Culture<br><br>5.2 Organisational Culture<br><br>5.3 Multicultural settings | 5 .1.1 Collectivist, Feminist, Long-term Oriented<br>5.1.2 Individualist, Monolithic, Short-term oriented<br>5.2.1 Consultative, democratic, rewarding innovators<br>5.2.2 Hierarchical, prescriptive<br>5.3.1 Monocultural traits<br>5.3.2 Multicultural, inter -cultural, trans-cultural |
| 6.0 Develop and deploy dissemination strategies | 6.1 Dissemination for awareness<br><br>6.2 Dissemination for understanding<br>6.3 Dissemination for full utilisation | 6.1.1 Internal and external communication (general)<br>6.2.1 Internal and external training - interested parties<br>6.3.1 On-going promotion and support including updates and future enhancements |

policies and actions for maximising the probability of successful innovations and valorisation. We based the development of INCUVA during the process of Materials Development for the VALO project. Qualitative argumentative research according to [3] is unstructured and might lead to ambiguities and subjective interpretations. For this reason further research is necessary in order to obtain both practical experiences from organizations and objective measures. In future the meta-framework will be validated through multiple case studies which will provide both qualitative and quantitative [27] which will help test hypotheses and, as such, will provide added credibility and the basis for generalisation. Future work will also be a longitudinal, multi-project and multi-country study concentrating on the role of new technologies such as social media and on performance measurements.

**Acknowledgements.** The work presented in this paper has been partially supported by the activities in the project "*ECQA Valorisation Expert Training and Certification*" (VALO), number 2011-1-GR1-LEO05-06789, funded by the EC LLP under the Leonardo da Vinci programme with support from the European Commission.

## References

[1] Allee, V., Taug, J.: Collaboration, innovation and value creation in a global telecom. The Learning Organization 13(6), 569–578 (2006)

[2] Phillips, J.: Does innovation add value? (2007), http://innovateonpurpose.blogspot.co.uk/2007/11/does-innovation-add-value.html (accessed February 12, 2013)

[3] Galliers, R.D.: Choosing information systems research approaches. In: Galliers, R.D. (ed.) Information Systems Research: Issues, Methods, and Practical Guidelines, pp. 144–162. Blackwell Scientific Publications, Oxford (1992)

[4] Farley, A.M.: Qualitative Argumentation, Computer and Information Science, University of OregonEugene, Oregon 97403 USA, pp. 89–96 (1997), http://www.qrg.northwestern.edu/.../Farley_1997_Qualitative_Argumentation.pdf (last accessed March 12, 2013)

[5] Trott, P.: Innovation management and new product development, 4th edn. Prentice Hall, Financial Times (2008)

[6] Greenhalgh, C., Rogers, M.: Innovation, Intellectual Property, and Economic Growth, A Report for the UK Intellectual Property Office (March 2010), http://www.ipo.gov.uk/ipreview-doc-h.pdf

[7] Chesbrough, H.: Open Innovation, The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, Boston (2003)

[8] Siakas, K., Messnarz, R., Georgiadou, E., Naaranoja, M.: Launching Innovation in the Market Requires Competences in Dissemination and Exploitation. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 241–252. Springer, Heidelberg (2012)

[9] Siakas, K., Georgiadou, E.: Knowledge Sharing in Virtual and Networked Organisations in Different Organisational and National Cultures. In: Bolisani, E. (ed.) Building the Knowledge Society on the Internet, ch. 3, Part 1, pp. 45–64. Idea Publishing (2008) ISBN: 978-1-59904-816-1

[10] Kroeger, A., Kluckhohn, F.: Culture: A Critical Review of Concepts and Definitions. Harward Business Review, Cambridge (1952)

[11] Lee, H., Varey, R.: Analysing cultural impacts of computer-mediated communication in organisations. In: Ess, C., Sudweeks, F. (eds.) Proceedings Cultural Attitudes Towards Communication and Technology, pp. 321–326. University of Sydney, Australia (1998)

[12] Schein, E.: Organisational Culture and Leadership. Jossey-Boss, San Fransisco (1986)

[13] Robey, D., Azevedo, A.: Cultural analysis of the organizational consequences of information technology. Accounting, Management & Information Technology 4(1), 23–37 (1994)

[14] Boztepe, S.: User Value: Competing Theories and Models. International Journal of Design 1(2), 55–63 (2007)

[15] Rokeach, M.: The nature of human values. Free Press, Collier-Macmillan, London (1973)

[16] Hilliard, A.L.: The forms of value: the extension of a hedonistic axiology. Columbia University Press, New York (1950)
[17] Morris, C.W.: Varieties of Human Value. University of Chicago Press, Chicago (1956)
[18] Sinden, J.A., Worrell, A.C.: Unpriced values: decisions without market prices. Wiley, New York (1979)
[19] Herbig, P., Dunphy, S.: Culture and innovation. Cross Cultural Management: An International Journal 5(4), 13–21 (1998)
[20] Martin, E.C., Terblanche, F.: Building organisational culture that stimulates creativity and innovation. European Journal of Innovation Management 6(1), 64–74 (2003)
[21] McLean, L.D.: Organizational Culture's Influence on Creativity and Innovation: A Review of the Literature and Implications for Human Resource Development. Advances in Developing Human Resources 7(2), 226–246 (2005)
[22] Hansen, M.T., Birkinshaw: The Innovation Value Chain. Harvard Business Review (June 2007)
[23] Siakas, K.V.: What Has Culture to do with SPI? In: EUROMICRO, pp. 376–381 (2002)
[24] Welsch, W.: Transculturality - the Puzzling Form of Cultures Today. In: Featherstone, M., Lash, S. (eds.) Spaces of Culture: City, Nation, World, pp. 194–213. Sage, London (1999)
[25] Uden, L., Naaranoja, M.: Co-creation of value for a public service. Int. J. Services, Economics and Management (2011)
[26] Hofstede, G.: Cultures' Consequences, Comparing Values, Behaviors, Institutions, and Organisations Across Nations. Sage Publications, Thousand Oaks (2001)
[27] Reid, A., Gough, S.: Guidelines for Reporting and Evaluating Qualitative Research: what are the alternatives? Environmental Education Research 6(1) (2000)

# Fostering Innovation and Entrepreneurship in European VET: EU Project "From Idea to Enterprise"

Marek Gavenda[1], Andreas Riel[2], Ana Azevedo[3], Marisa Pais[3], Eva Homolová[1], Jiří Balcar[1], Alessandra Antinori[4], Giuseppe Metitiero[4], Giorgos Giorgakis[5], Photis Photiades[5], Damjan Ekert[6], Richard Messnarz[6], and Serge Tichkiewitch[2]

[1] RPiC-VIP, Ostrava, Czech Republic
{gavenda,homolova,balcar}@rpic-vip.cz
[2] EMIRAcle c/o Grenoble Institute of Technology GSCOP UMR5272, Grenoble, France
{andreas.riel,serge.tichkiewitch}@grenoble-inp.fr
[3] ISQ, Lisbon, Portugal
{aiazevedo,marisa.pais}@isq.pt
[4] CIRCES, Rome, Italy
ale.antinori@fastwebnet.it, g.metitiero@primaforma.net
[5] EUROSC, Nicosia, Cyprus
{george,photis}@eurosc.eu
[6] ISCN GmbH, Graz, Austria
{dekert,rmess}@iscn.com

**Abstract.** The research and development of modern products and systems is coined by the increasingly important requirement to create sustainable innovation. Innovation is intimately linked with entrepreneurship. Ideas, Innovation and Entrepreneurship are considered the keys to a wealthy and sustainable economy. The best way to turn ideas into innovations is to consider key innovation factors from the earliest phases of research all along the way to products and services. Existing entrepreneurship education and training programs, however, do not take into account these early phases, but rather focus on the process of creating a new enterprise. This paper presents "From Idea to Enterprise", a European Project that transfers the innovative "ECQA Certified EU Researcher-Entrepreneur" certified program from academic to VET level.

**Keywords:** Innovation, Innovation Transfer, Entrepreneurship, Lifelong Learning.

## 1 Introduction

At a time when entire economies and industries are reeling from the financial crisis, business leaders are struggling to balance the near-term needs of survival with the long-term demand to find new sources of growth. Never has the need to innovate and be entrepreneurial been more urgent. If it is to make a success of the Lisbon strategy for growth and employment, Europe needs to stimulate the entrepreneurial mindsets of young people, encourage innovative business start-ups, and foster a culture that is friendlier to entrepreneurship and to the growth of small and medium-sized enterprises (SMEs). The important role of education in promoting more entrepreneurial

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 282–293, 2013.

attitudes and behaviours is now widely recognised [1]. However, the benefits of entrepreneurship education are not limited to start-ups, innovative ventures and new jobs. Entrepreneurship refers to an individual's ability to turn ideas into action and is therefore a key competence for all, helping young people to be more creative and self-confident in whatever they undertake. Therefore, the primary purpose of entrepreneurship education should be to develop entrepreneurial capacities and mindsets. In this context, entrepreneurship education programmes can have different objectives, such as:

1. developing entrepreneurial drive among students (raising awareness and motivation);
2. training students in the skills they need to set up a business and manage its growth;
3. developing the entrepreneurial ability to identify and exploit opportunities. Graduates' start-up is one of a range of possible outcomes.

The majority of courses and training programs focus on target 2). The European program ResEUr [2] has been created in 2011 to sensitise young researchers for sustainable entrepreneurship and innovation, which is neglected by most other comparable programs. The research presented in this paper concerns the transfer of this program to VET and secondary education levels as the principal objective of the Leonardo da Vinci Lifelong Learning Programme EU project "From Idea to Enterprise" on the basis of a Europe-wide needs analysis.

## 2 Background

The Global Entrepreneurship Monitor, a program initiated by Babson College and the London Business School to assess entrepreneurial activity around the world [3], explores the link between economic growth and entrepreneurship. The 2008 report discusses this positive relationship in terms of the diverse phases of economic development that vary from country to country: Factor-driven economies, i.e., economies relying on unskilled labour and natural resource extraction, will need to focus on institution-building, infrastructure and the provision of such basic services as healthcare. Efficiency-driven economies, i.e., growing economies in need of improving production processes and quality, will be most concerned with such priorities as domestic and/or foreign market size, financial market sophistication and labour market efficiency. Innovation-driven economies—the most-advanced stage in which businesses compete primarily on the basis of innovation—will have needs related to entrepreneurship-specific education and research and development [4].

Currently the teaching of entrepreneurship is not yet sufficiently integrated in higher education institutions' curricula. Available data show that the majority of entrepreneurship courses are offered in business and economic studies. The diffusion of entrepreneurship is particularly weak in some of the Member States that joined the EU in and after 2004 [1]. However, it is questionable whether Business Schools are the most appropriate place to teach entrepreneurship: innovative and viable business ideas are more likely to arise from technical, scientific and creative studies. So the real challenge is to build inter-disciplinary approaches, making entrepreneurship education accessible to all students, creating teams for the development and exploitation of business ideas, mixing students from economic and business studies with students from other faculties and with different backgrounds.

Entrepreneurship refers to an individual's ability to turn ideas into action. It includes creativity, innovation and risk taking, as well as the ability to plan and manage projects in order to achieve objectives. This supports everyone in day-to-day life at home and in society, makes employees more aware of the context of their work and better able to seize opportunities, and provides a foundation for entrepreneurs establishing a social or commercial activity [5]. Entrepreneurship is today recognised as a basic skill to be provided through lifelong learning. The European Council of Lisbon and the European Charter for Small Enterprises have emphasised this aspect. All these activities are part of a more general effort of the European Commission to promote SMEs and entrepreneurship, which includes the adoption of [6] and of a set of other related documents in January 2003.

## 3 Methodology

ResEUr – ECQA Certified Researcher-Entrepreneur – is the name of the innovation project that has been co-financed by the European Commission from November 2009 to November 2011, and carried out by a consortium of five partners in Europe in order to propose a competence set for entrepreneurial minds, as well as a complete e-Learning based training and certification program [7]. These partners were EMIRAcle (BE), University Politehnica of Timisoara (RO), Grenoble INP (FR), ISCN Ltd. (IE), proHUMAN (SI), and Skills International GmbH (AT). All these partners have a long-time experience in entrepreneurship and innovation, and are active members of the ECQA (www.ecqa.eu). With ResEUr, their target was to define a competence set that is complementary to existing training and education programs in entrepreneurship. ResEUr primarily addresses the phases before the decision of creating an enterprise is made. It aims at sensitising researchers for entrepreneurship and innovation rather than teaching them how to do business plans. This idea results from the conviction that the issue of taking into account innovation and marketing issues already during research is crucial.

"From Idea to Enterprise" has been launched in October 2012 in order to transfer ResEUr to VET and secondary education on a European level, following the national priorities in many EU member states. The project partners are the following: RPIC-ViP s.r.o. (CZ), ISQ (PT), EMIRAcle (BE), ISCN (AT), EUROSUCCESS CONSULTING (CY), CIRSES (IT).

The first part of the project is the specification of the competence set, from which e-Learning based training material in six languages and test questions for certification will be developed. In this paper, the authors focus on the derivation of the competence set from ResEUr by a needs analysis in the transfer target partner countries. The specification is compliant with the European Qualification Framework (EQF) [8], and is based on the concept that the skills which characterize a specific job role define the so called Skill Card (or Skill Set), which contains skill units, which consist of skill elements. The competences expected from a candidate who wants to get certified for a particular skill element are specified by performance criteria. For certification, the candidate is tested on the basis of a pool of test questions that have been specified for each performance criterion. Alternatively, candidates can ask for the assessment of documents that prove that they have successfully applied the principles and associated performance criteria in their professional activities.

## 4 Key Competences

Figure 1 shows the Skill Card which has been developed by the ResEUr project consortium, and which also serves as the basis for the innovation transfer project. The focus is clearly on networking and open innovation. Only one unit is dedicated to the typically taught entrepreneurial skills, which are mainly associated with mastering the process of creating an enterprise.

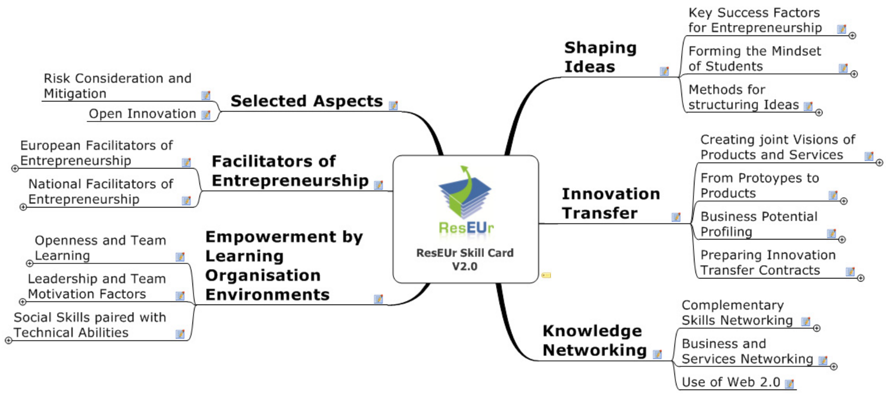



**Fig. 1.** ResEUr Skill Set

### 4.1 Shaping Ideas

This unit deals with key skills that are required to leverage brilliant ideas, starting from creating an innovative mindset in students' heads, passing via methods for structuring ideas, and ending by discussing methods and best practices for presenting ideas.

#### 4.1.1 Key Success Factors for Entrepreneurship

The aim of the element is to describe the main topics which young entrepreneurs have to understand in order to make relevant decisions of how to transfer products and/or services as innovations to the market:

- General life cycle from idea to production;
- key success factors (providing latest market research statistics);
- understanding finances (ex. topics like: dynamics of financing, business models for exploitation, etc.);
- understanding marketing of products;

- understanding legal issues;
- understanding the risks associated with entrepreneurship.

#### 4.1.2 Forming the Mindset of Students

Forming the minds of students means to train them in applying systematic techniques to find innovative solutions to concrete problems [9]. The method envisaged in this skill element is to create a library of case studies which demonstrate the key issues about creating ideas for innovative solutions. Most of them are based on the student's capability of taking into account basic proven principles from several different domains. Some particular subjects addressed are:

- experimental learning;
- use of modern IT facilities (web, knowledge
- databases, etc.) to increase the understanding of the idea, and to gain deeper insight;
- using real working examples (prototyping);
- thinking about potential applications of concrete research in different contexts.

#### 4.1.3 Methods for Structuring Ideas

The aim of the element is to provide methods for structuring the ideas based on the integration of complementary networked skills and in such a way that they address specific problems, and have a long term broader vision for development. Students specifically would learn to

- understand the Belbin and similar innovation theory models where a mix of complementary roles forms the basis for innovative teams and results. The components of the service/product shall be matched onto these skills;
- understand how to map ideas/solutions onto specific problems to provide answers for industry;
- do a joint goals analysis to formulate a vision for the future service/product and structure the idea aligned with the vision.

### 4.2 Innovation Transfer

This unit focuses on issues concerned with the transfer of research in the form of knowledge, technology, prototypes, and services as innovations from the academic environment to the competitive market. These issues are known to represent key success factors of entrepreneurship in the academic domain.

#### 4.2.1 Creating Joint Visions of Products and Services

Several types of modern products create customer value in combination with associated services. Furthermore, in many sectors margins are significantly higher in services than in products. It is thus increasingly important to think about services associated to a certain product in the very early product development phases. This process can be highly facilitated by bringing together as many stakeholders of the product life-cycle as possible, in order to make them exchange knowledge about requirements and constraints with respect to a specific product and its life cycle [10]. This element gives an introduction into methods that help carry out this knowledge networking task with the target to create a vision about the future product and its associated services that is shared by as many different stakeholders as possible.

#### 4.2.2 From Prototypes to Products

A significant number of Start-Ups fail due to the fact that effort for supporting one or several products on the market has been underestimated. The way from a research prototype to a sellable product that provides a certain level of quality is typically very long one. The necessity and the efforts of supporting the product once it is in the hands of customers is also often not sufficiently considered. This learning element focuses on the key factors that have to be taken into account before actually taking the direction to the creation of product out of a product idea or a prototype, including risk management issues.

#### 4.2.3 Business Potential Profiling

The aim of the element is to describe the needs of industry on the one hand and researchers or research organisations on the other hand for successful knowledge and technology transfer. Technology transfer defines the process of transformation of the results of research and development into marketable products or services. Best practices are introduced of how to find out about the needs of industry and academia for collaboration and knowledge and technology transfer:

- needs analysis for knowledge and technology transfer for research organisations and industry;
- innovation audits, innovation scans;
- key criteria and elements of a technology profile;
- formulating the innovation transfer offer and request (for products, services, technology etc.).

The aim is also to describe how to formulate technology offers or technology requests for knowledge transfer for further dissemination, such as databases and service providers for diffusion and dissemination of technology profiles, and researcher networks. The element also highlights the ways of preparing and presenting cooperation needs in the field of knowledge and innovation transfer.

#### 4.2.4 Preparing Innovation Transfer Contracts

The aim of this element is to describe the different types of collaboration that are available for knowledge and technology transfer. Best practices are introduced of how to find out about the best suitable collaboration types for industry and research organisations:

- technology transfer and corporate strategy;
- types of cooperation;
- confidentiality and non-disclosure agreements;
- technology transfer agreement categories

The element also deals with ways of choosing the right type of collaboration for knowledge and technology transfer.

## 4.3 Knowledge Networking

In the ResEUr qualification, Knowledge Networking is considered the core competence area for entrepreneurs. Networking knowledge from several different domains and sectors can create the decisive competitive advantage of modern and future-oriented enterprises. The unit highlights several significant factors of this networking paradigm with special relevance for enterprise creators in the academic domain.

### 4.3.1 Complementary Skills Networking

The aim of the element is to provide examples and experiences from success cases about how to build and become integrated into complementary skills networks. Students will learn about specific success stories including

- the campus company concept (pool of students offering skills as services to industry);
- the company spin off concept (find a community of a mixture of companies, university institutions to support a spin-off);
- the integration in conferences and networking concept;
- the topic driven community building and extracting of ideas concept;
- the main reasons why such models are working.

The element also deals with principles that can be drawn from these case studies.

### 4.3.2 Business and Services Networking

The aim of the element is to provide examples and experiences from success cases about how to interface with real business networks (beyond student and development networks). Students will learn to

- understand the rules and behaviour in business networks;
- know the main differences and potential synergies between research and business networks and how to identify and exploit such synergies;
- typical situations/concepts to get linked up with business networks and build on a group of customers;
- understand the requirements for services to allow business customers accepting solutions from young researchers.

### 4.3.3 Use of Web 2.0

The added value of using modern Web 2.0 facilities for applying the discussed networking skills are pointed out as well.

## 4.4 Empowerment by Learning Organisation Environments

This unit puts the concept of the Learning Organisation in the middle of the successful enterprise creation.

### 4.4.1 Openness and Team Learning

The aim of the element is to provide methods for training openness, for new strategies in knowledge sharing in a team and its advantages, and for feedback and leadership approaches accepting and empowering the integration of new ideas. Students will specifically learn

- the requirements for a team learning culture and how to exploit it;
- the requirements for openness principles and how to socially train/exercise openness;
- the requirements of team (leadership) styles allowing innovation to grow and to empower new ideas.

#### 4.4.2 Leadership and Team Motivation Factors

This skill element focuses on:

- leadership behaviour when considering the entrepreneurial attitude (style, emotions, self motivation) - individual behaviour;
- leadership by a group (leadership teams, action oriented team leadership skills for cross-functional teams) - group/team behaviour.

#### 4.4.3 Social Skills Paired with Technical Abilities

This skill element's objective is to train students in the field of social communication based on the valorisation of their emotional intelligence. The key elements are related to:

- the intellectual capital concept and its relation/effects to entrepreneurship development;
- social communication skills development – elements that are derived from the emotional intelligence effects in the research-entrepreneur field and that are linked with behaviour attitude like adaptation, integration, differentiation, complexity and complementing;
- social responsibility;
- multicultural skills.

The skill element training will be based on case studies and best practices.

### 4.5 Facilitators of Entrepreneurship

This unit provides highly condensed and concise information about key issues of entrepreneurship, which are typically taught in seminars which are currently offered by various institutions. The unit, however, does not want to replace such seminars and courses, but it rather seeks to give the student a convenient means of reflexion on whether she/he needs training in the respective competence areas, and where she/he can find complementary courses.

#### 4.5.1 European Facilitators of Entrepreneurship

The aim of the element is to provide information which institutions, programs, infrastructure, etc is available at European level:

- knowledge about entrepreneurship available on EU level (portals, models);
- facilitating systems at the EU level (scholarships for young researchers, opportunities for funded projects);
- funding opportunities;
- other facilitators (business angels, scholarships, etc.).

### 4.5.2 National Facilitators of Entrepreneurship

The aim of the element is to provide information which institutions, programs, infrastructure, etc is available in certain countries in order to facilitate the entrepreneurship. The content of the element is tailored on the basis of the situation in each country where the seminar is delivered.

- Graduate programs on entrepreneurship;
- facilitating systems at the government level (scholarships for young researchers, opportunities for projects);
- supporting programs at universities and research institutions;
- incubators;
- interest of industry;
- etc.

## 4.6 Selected Aspects

This unit contains a selection of skill elements that are considered very interesting and relevant for the target audience of ResEUr, and which are implicitly part of all the other skill units and elements.

### 4.6.1 Risk Consideration and Mitigation

Risk issues should be taken into account in the elaboration of all the skill elements. Focus should be set on:

- risk identification by their typology and by identifying their potential sources;
- risk awareness and evaluation;
- entrepreneurs' behaviour when they are confronted with risk (risk adversity behaviour);
- risk mitigation.

### 4.6.2 Open Innovation

Innovative enterprises that are successful in the modern knowledge-based networked economy have succeeded to drive their innovation in knowledge networks rather than in hermetically protected research departments. Open Innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology. The boundaries between a firm and its environment have become more permeable; innovations can easily transfer inward and outward. The central idea behind open innovation is that in a world of widely distributed knowledge, companies cannot afford to rely entirely on their own research, but should instead buy or license processes or inventions (e.g. patents) from other companies. In addition, internal inventions not being used in a firm's business should be taken outside the company (e.g., through licensing, joint ventures, spin-offs). This element introduces these key concepts of Open Innovation to the student.

## 5 Needs Analysis Approach and Results

The objective of the need analysis carried out in the transfer project was to identify which knowledge skills and competencies of the ResEUr program are considered relevant for the transfer, and therefore need to be adapted to the particular needs of the partner countries. This means that both the identification of target groups and the identification of their needs had to be done in each partner country that is a target of the ResEUr transfer. The analysis resulted in national reports for each of these countries, based on which the identification and consideration of differences among countries can be carried out, and help maximise the benefits/costs ratio. Each partner specified the characteristics of the national VET system on the basis of "official materials" (strategies, government materials, laws etc.), including a description of potential targeted VET "clients". The assessment of the relevance of the particular skill elements of the ResEUr program, and the current level of their development in national initial VET has been done through expert surveys according to the following key criteria: *At least 25 experts were involved in the survey in each country (min. 5 experts in the labour market or entrepreneurship, and min. 20 managers of initial VET facilities). The survey was based on a questionnaire containing the specific Performance Criteria of each ResEUr Skill Card Element.* Experts were asked to assess the importance of each knowledge element (performance criterion), as well as the current level of the latter's development in initial VET by selecting one of the predefined answers. There were also open questions for comments, ideas and recommendations per knowledge elements. The quantitative results of this survey are summarised in Table 1, where for each target country the column "importance" indicates the mean value of the judgment of the importance of the corresponding knowledge elements, and the column "development" indicates the judgment of their current level of development. The following values were available for choice: *Importance: not important = 0, partly important = 1, important = 2. Level of development: not developed in VET = 0, partly developed in VET = 1, fully developed in VET = 2.* A knowledge element is considered relevant for the transfer only if the mean value of "importance" is equal to or higher than 1.5, and the mean value of "level of development" is lower than 1.5 ("Y" for "Yes" in the columns "important & not developed").

## 6 Conclusions and Outlook

The results of the target group needs analysis proved that there is an identified demand for almost all the ResEUr elements to be trained in the transfer countries. Within the "From Idea to Enterprise" project, these elements will be adapted to the specific needs of the identified VET target groups in terms of the associated e-learning based training material and the test questions for the certification. In particular, the material will be enriched by relevant exercises, examples and case-studies. These modifications will be developed during the training of trainers in spring/summer 2013, and tested during the pilot trainings with target groups from autumn 2013. Interested parties are kindly invited to participate.

**Table 1.** Main results of the expert survey (values rounded due to space constraints)

| | Cyprus | | | Czech Rep. | | | Italy | | | Portugal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Importance | Development | Important & not developed | Importance | Development | Important & not developed | Importance | Development | Important & not developed | Importance | Development | Important & not developed |
| **U1: Shaping Ideas** | **1,6** | **1,1** | **Y** | **1,6** | **0,9** | **Y** | **1,8** | **1,3** | **Y** | **1,9** | **0,7** | **Y** |
| E1: Key Success Factors for Entrepreneurship | 1,7 | 1,2 | Y | 1,7 | 0,9 | Y | 1,9 | 1,5 | N | 1,9 | 0,9 | Y |
| E2: Forming the Mindset of Students | 1,5 | 1,0 | N | 1,6 | 0,9 | Y | 1,8 | 1,1 | Y | 1,9 | 0,7 | Y |
| E3: Methods for structuring Ideas | 1,6 | 1,1 | Y | 1,6 | 1,1 | Y | 1,7 | 1,4 | Y | 1,8 | 0,6 | Y |
| **U2: Innovation Transfer** | **1,6** | **1,1** | **Y** | **1,6** | **0,9** | **Y** | **1,7** | **1,1** | **Y** | **1,8** | **0,7** | **Y** |
| E1: Creating joint Visions of Products and Services | 1,7 | 1,0 | Y | 1,3 | 0,9 | N | 1,7 | 1,3 | Y | 1,8 | 0,7 | Y |
| E2: From Prototypes to Products | 1,7 | 1,3 | Y | 1,6 | 1,1 | Y | 1,8 | 1,2 | Y | 1,9 | 0,9 | Y |
| E3: Business Potential Profiling | 1,5 | 1,0 | N | 1,8 | 0,9 | Y | 1,7 | 1,0 | Y | 1,8 | 0,5 | Y |
| E4: Preparing Innovation Transfer Contracts | 1,6 | 1,0 | Y | 1,6 | 0,6 | Y | 1,5 | 1,0 | Y | 1,8 | 0,6 | Y |
| **U3: Knowledge Networking** | **1,6** | **1,2** | **Y** | **1,5** | **1,0** | **Y** | **1,4** | **1,0** | **N** | **1,8** | **0,8** | **Y** |
| E1: Complementary Skills Networking | 1,4 | 1,0 | N | 1,5 | 0,9 | N | 1,2 | 0,8 | N | 1,8 | 0,6 | Y |
| E2: Business and Services Networking | 1,7 | 1,2 | Y | 1,7 | 0,8 | Y | 1,5 | 1,1 | Y | 1,9 | 0,8 | Y |
| E3: Use of Web 2.0 | 1,6 | 1,2 | Y | 1,4 | 1,3 | N | 1,5 | 1,1 | Y | 1,7 | 1,0 | Y |
| **U4: Empowerment by Learning Organisation Environments** | **1,6** | **1,4** | **Y** | **1,6** | **1,1** | **Y** | **1,8** | **1,4** | **Y** | **1,9** | **0,9** | **Y** |
| E1: Openness and Team Learning | 1,6 | 1,4 | Y | 1,5 | 0,9 | Y | 1,8 | 1,5 | Y | 1,9 | 0,9 | Y |
| E2: Leadership and Team Motivation Factors | 1,6 | 1,5 | Y | 1,7 | 1,1 | Y | 1,8 | 1,5 | Y | 1,9 | 0,9 | Y |
| E3: Social Skills paired with Technical Abilities | 1,5 | 1,3 | N | 1,7 | 1,1 | Y | 1,8 | 1,3 | Y | 1,8 | 0,8 | Y |
| **U5: Facilitators of Entrepreneurship** | **1,6** | **1,1** | **Y** | **1,6** | **0,9** | **Y** | **1,6** | **0,9** | **Y** | **1,8** | **0,7** | **Y** |
| E1: European Facilitators of Entrepreneurship | 1,7 | 1,2 | Y | 1,5 | 0,9 | Y | 1,7 | 0,9 | Y | 1,9 | 0,7 | Y |
| E2: National Facilitators of Entrepreneurship | 1,6 | 1,0 | Y | 1,6 | 1,0 | Y | 1,4 | 1,0 | N | 1,8 | 0,8 | Y |
| **U6: Selected Aspects** | **1,7** | **1,1** | **Y** | **1,5** | **0,8** | **Y** | **1,7** | **1,0** | **Y** | **1,9** | **0,8** | **Y** |
| E1: Risk | 1,7 | 1,3 | Y | 1,6 | 1,1 | Y | 1,9 | 1,2 | Y | 2,0 | 0,8 | Y |
| E2: Open Innovation | 1,6 | 0,9 | Y | 1,5 | 0,5 | N | 1,5 | 0,8 | Y | 1,9 | 0,7 | Y |

**Acknowledgements.** The "From Idea to Enterprise" project is financially supported by the European Commission in the Leonardo da Vinci part of the Lifelong Learning Programme under the project number CZ/12/LLP-LdV/TOI/134007. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

## References

1. Entrepreneurship in higher education, especially within non-business studies. Final Report of the Expert Group. European Commission, Enterprise and Industry Directorate-General. Promotion of SMEs competitiveness Entrereneurship (2008)
2. ResEUr – ECQA Certified Researcher-Entrepreneur. Leonardo da Vinci project number 503021-LLP-1-2009-1-BE-LEONARDO-LMP
3. `http://www.gemconsortium.org/default.aspx` (last accessed on February 28, 2011)
4. J. Gaynor, A. Mackiewicz, R. Ramaswami. Entrepreneurship and innovation. Editorial. The keys to global economic recovery. Ernst & Young (2009)
5. Fostering entrepreneurial mindsets through education and learning. Commission Communication COM, 33 final (2006)
6. The Green Paper on Entrepreneurship in Europe, `http://europa.eu.int/comm/enterprise/entrepreneurship/green_paper/index.htm` (last accessed on February 22, 2011)
7. Draghici, A., Draghici, G., Riel, A., Tichkiewitch, S.: Research Entrepreneurship Development – A Possible Solution at the European Level. In: Proceedings the 5th International Conference on Manufacturing Science and Education (MSE 2011), Sibiu, Romania, June 2-5, vol. II, pp. 135–138 (2011) ISSN 1843-2522
8. The European Qualifications Framework (EQF), `http://ec.europa.eu/education/lifelong-learning-policy/doc44_en.htm` (last accessed on November 22, 2010)
9. Tichkiewitch, S.: Peut on former à l'innovation de produit et à la créativité. In: 3rd International Industrial Engineering Conference, Montréal, May 26-28 (1999)
10. Brissaud, D., Tichkiewitch, S.: Innovation and manufacturability analysis in an integrated design context. Computers in Industry 43(2), 111–121 (2000)

# VALO$_5$ – Innovation, Maturity Growth, Quality and Valorisation

Elli Georgiadou[1] and Kerstin Siakas[2]

[1] Middlesex University, School of Science and Technology,
The Borroughs, London NW4 4BT, UK
e.georgiadou@mdx.uk
[2] Alexander Technological Educational Institute of Thessaloniki,
Department of Informatics, P.O. Box, 141, GR-57400 Thessaloniki, Greece
siaka@it.teithe.gr

**Abstract.** Organisations and individuals maximise the likelihood of success through managing innovation. Ensuring the high quality of both process and product, sustaining and exploiting innovations creates value to the stakeholders. In this paper we explore the nexus of maturity, quality and valorisation. We consider that the growth of organisational maturity changes the nature and role of quality management and characterises valorisation. We propose a Valorisation model based on the INCISIV framework (which incorporates the PDCA Deming Improvement cycle) and the CMMI model for understanding, evaluating, measuring and improving the valorisation process, and the valorisation results.

**Keywords:** PDCA, INCISIV, CMMI, Valorisation, VALO.

## 1 Introduction

According to Trott [1] "*Innovation is not a single action but a total process of interrelated sub-processes. It is not just the conception of a new idea, nor the invention of a new device, nor the development of a new market. The process is all these things acting in an integrated fashion. … Innovation is the management of all the activities involved in the process of idea generation, technology development, manufacturing and marketing of a new (or improved) product or manufacturing process or equipment*". Often organisations, projects and individuals fail to gain adequate value let alone added value from their innovations. On European Union (EU) level '*the term 'added value' is a centrepiece in contemporary debates on the reform of the EU budget. Both at the academic and the political level, calls are being made to revise EU spending on the basis of added value considerations. Yet, as pointed out by many observers, the notion of added value lacks conceptual clarity*' [2]. Nevertheless, the EU funds an enormous number of projects whose outcomes are poorly exploited. In particular projects consisting of purely research oriented and/or technically oriented partners seem to lack awareness of the importance of

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 294–299, 2013.

dissemination, exploitation and valorisation for sustainable development and skills in carrying out such activities [3]. The VALO project[1] intends to address this issue.

The rest of this paper will proceed as follows: First we outline the key features of the two main elements of interest in this study: Valorisation of Innovation and Maturity Models. We present a valorisation maturity model 'VALO5' that builds on the InCISIV framework which includes the Deming PDCA-cycle for improvement. Finally we draw some conclusions and outline further work

## 2 Adding Value to Innovation Through Valorisation

Value attributable to many innovative projects tends to run out in relation to the allocated funding [4]. Thus the value created is not sustained beyond the lifetime of the project. Effective innovation should not only facilitate the creation of value but should also ensure that such value is sustained and shared to its optimum potential. In particular projects consisting of purely research oriented and/or technically oriented partners seem to lack knowledge of the importance of dissemination, exploitation and valorisation for sustainable development [3]. There also seems to be a gap in skills for carrying out actions of dissemination and exploitation. The European Union recently responded to this challenge by funding a specific 'Valorisation' project – here after called VALO. *'Valorisation is defined as the process of sustaining value created through innovation and hence optimising its impact among the direct and indirect beneficiaries' [5]*

The rationale for the VALO project is derived from the observation that many projects are often carried out in isolation and hence they provide very little or no lasting impact. These projects tend not to imagine or plan for continuing dissemination and exploitation of their results and deliverables beyond the allocated funding period; and even when they do, there is little evidence that they succeed The impact/benefit of a sustainable project translates into added value gained by a diverse group of stakeholders and/or specific target groups well beyond the lifespan of the project. All projects therefore, need to valorise their results for maximising achievements and increasing sustainability after their completion.

## 3 Maturity Models

Mettler (2009) studied the parameters needed for the development and application of a maturity model and presents a meta-model of the parameters needed by showing the development process and the application process in two overlapping circles (Figure 1).

Mettler [6] argued, that "*the development of the maturity model is intimately connected with the application phase and therefore should not be reflected separately*". The reason is that the order of the phases impacts on the application of the model.

---

[1] `http://www.ecqa.org/index.php?id=294` and `valo.it.teithe.gr`

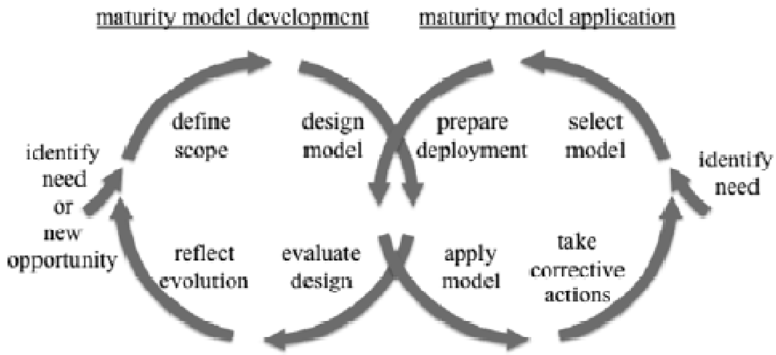



**Fig. 1.** The parameters of the maturity model development & application process (source Mettler, 2009)

Hain and Back [7] identified 55 maturity models in the area of collaboration, knowledge management and e-learning. They distinguished three categories of maturity models, namely: scientific, practitioner-oriented (scientific), and practitioner-based. Most of the maturity models were derived from the per se standard Capability Maturity Model (CMM) [8] and later on Capability Maturity Model Integrated (CMMI), but, only a few were adequately documented to be further evaluated or applied in practice. They also assert that "*non-CMM-based maturity models are rather chaotic and leak in an appropriate form or functioning*". They conclude that an activity is always connected to a maturity level, which means that along the maturity range different topics / activities are of relevance. As a result the required activities change with increased maturity. They argue that this implies that a maturity model is rather a maturity process.

Process capability growth in organisations is depicted in all known maturity models in a ladder-like diagram suggesting an ascent from lower steps to higher steps. In [9] we studied the relationship of maturity and knowledge sharing which in turn improves performance.

## 4 Building on the InCISIV Framework

Under the auspices of the VALO project, we developed the InCISIV framework [9] which facilitates the study of the relationships between Innovation, Communication and Valorisation. The Deming Plan-Do-Check-Act Cycle, also called PDCA cycle [10] is embedded in this new process quality model in an effort to focus project managers, project teams as well as evaluators to identify, plan, monitor, evaluate, improve and manage the valorisation project. The framework is depicted in figure 2 and shows two cycles which interact at every stage, delivering outputs incrementally. InCISIV allows for agile responses to change, planning the quality strategy, continuous reviewing and evaluation of project progress and quality of deliverables as well as improvement suggestions

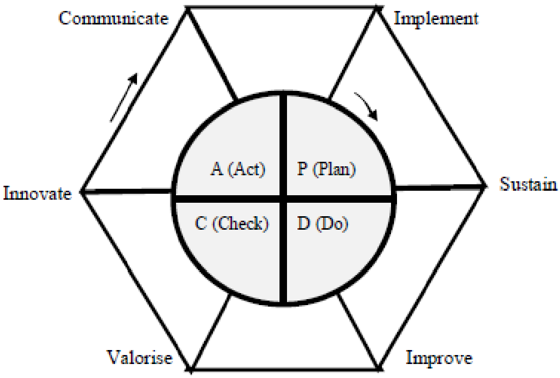



**Fig. 2.** The InCISIV framework [9]

At each stage activities are planned, carried out, outputs are evaluated and processes improved. The quality of outputs is thus continuously monitored, and sustainability is improved. However, we did not venture into looking at value-adding activities beyond the lifetime of projects. Experience from the software industry, where many different maturity models, such as CMMI and ISO 15504 are used for measuring and improving organisational performance, has demonstrated that the more mature an organisation is in terms of following best practices, the higher is the workforce awareness of organisational aims and objectives and the more committed is the workforce to holistic and strategic perspectives. On one hand organisations need to have a process to follow in order to know what to do when difficulties arise. On the other hand the processes need to be flexible enough to allow for agility and innovation. Maturity Modelling is a generic approach that describes the development of an entity over time progressing through levels towards an idealistic ultimate stage (Khoshgoftar and Oshman, 2009). Expressed in terms of direction towards the ultimate goal the maturity model shows the remaining distance to reach the ultimate goal. However, as shown in the InCISIV framework every stage goes through the PDCA cycle and slowly the journey approaches its ultimate goal via different levels that have different characteristics. It is not possible to run before you can walk. Similarly all the steps have to be completed in order to reach a higher level.

## 5 VALO$_5$ Maturity Model

The VALO$_5$ Valorisation Maturity Model (Figure 3) represents the maturity level that characterises a Valorisation process and its likelihood of success within a project team, an organisation, group or partnership. The circles underneath each step (level) depict the PDCA circle. Without improvement in each step it is not possible to reach the following level. The partners in a project team, an organisation, group or partnership can go in several circles without improving enough to reach the following level. It takes awareness and commitment to mature and usually this is a long process.

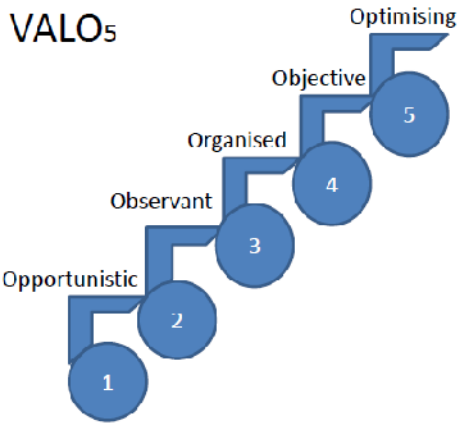



**Fig. 3.** The $VALO_5$ Maturity Model

**$VALO_1$ Opportunistic Level.** At this ad-hoc maturity level the success of the process depends on 'heroics', individuals with flair, new ideas, so some innovations may be produced but exploiting them and sustaining their value is also incidental and opportunistic. Knowledge (mostly tacit) is not shared; the individuals are not recognised or rewarded. Value is thus unlikely to be gained from the current or future projects

**$VALO_2$ Level.** At this repeatable maturity level good practice is identified. Previous successes can be repeated. Knowledge is thus shared within project teams which can apply this knowhow to subsequent projects. Innovators start to be recognised and encouraged. Observing previous successes can improve the chances of valorisation success.

**$VALO_3$ Organised Level.** At this managed level processes are organised and deployed systematically across projects. Roles and responsibilities are specified and plans together with Key Performance Indicators and targets are developed and established. Valorisation forms an integral part of the management process. Knowledge is shared across projects. Innovators are rewarded.

**$VALO_4$ Objective Level.** At this measured maturity level data are collected, innovators are recognised and rewarded and systematically sponsored, knowledge is shared across the whole organisation. Exploitation of innovations is institutionalised. Innovations can be sustainable and successes are objectively measurable.

**$VALO_5$ Optimising Level.** At the optimising level data are collected, analysed, interpreted and knowledge is shared at all levels (teams, projects, departments, partners, stakeholders). Evaluation and feedback is institutionalised. Valorisation is planned, organised, funded and deployed across groups, departments, the whole organisation and across partnerships/consortia. Value-adding activities continue beyond the completion of projects resulting in sustained improvements.

## 6 Conclusion and Further Work

This study arose from the VALO project which realised the need to provide support for sustainability of innovations. In this paper we presented the VALO$_5$ Model which together with the INCISIV framework can help set the foundations initially for the successful Valorisation of the actual VALO project but also of other EU projects and projects in organisations. Future work will involve the deployment of the model to both industry and academia for the scientific validation through the collection of multi-case study data.

**Acknowledgements.** The work presented in this paper has been partially supported by the activities in the project "*ECQA Valorisation Expert Training and Certification*" (VALO), number 2011-1-GR1-LEO05-06789, funded by the EC LLP under the Leonardo da Vinci programme with support from the European Commission.

## References

[1] Trott, P.: Innovation management and new product development, 4th edn. Pearson Education Limited, Edinburgh (2008)
[2] Eulalia, R.: The "added value" in EU budgetary debates: one concept, four meanings. Notre Europe (28) (2011), `http://ftp.infoeuropa.eurocid.pt/database/000047001-000048000/000047220.pdf` (accessed March 15, 2013)
[3] Siakas, K., Messnarz, R., Georgiadou, E., Naaranoja, M.: Launching Innovation in the Market Requires Competences in Dissemination and Exploitation. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 241–252. Springer, Heidelberg (2012)
[4] Sheriff, M., Georgiadou, E., Abeysinghe, G., Siakas: INCUVA: A meta-framework for sustaining the value of innovation in multi-cultural settings. In: EuroSPI 2013 (accepted)
[5] Della Corte, V., Savastano, I., Storlazzi, A.: Service innovation in cultural heritages management and valorization. International Journal of Quality and Service Sciences 1(3), 225–240 (2009)
[6] Mettler, T.: A Design Science Research Perspective on Maturity Models in Information Systems, BE IWI/HNE/03, University of St. Gallen, St. Gallen, Switcherland (2009)
[7] Hain, S., Back, A.: State-of-the-art on Maturity Models for Collaboration, Report Number: 01/2009, Universität St. Gallen Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG), Switcherland (2009)
[8] Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability Maturity Model, Version 1.1. IEEE Software 10(4), 18–27 (1993)
[9] Siakas, K., Georgiadou, E.: Towards Maximising the Quality of Social Impact through Valorisation of LLPs. In: Valtanen, J., Berki, E., Ruohonen, M. (eds.) Eduaction Matters, INSPIRE XVII, pp. 89–100. The British Computer Society (2012)
[10] Deming, W.E.: Out of the Crisis: quality, productivity and competitive position, Massachusetts, USA (1986)

# Customer-Driven Software Product Development Software Products for the Social Media World – A Case Study

Thomas Fehlmann[1] and Eberhard Kranich[2]

[1] Euro Project Office AG, Zeltweg 50, 8032 Zürich, Switzerland
thomas.fehlmann@e-p-o.com
[2] T-Systems International GmbH, Bonn, Germany
eberhard.kranich@t-systems.com

**Abstract.** Nowadays customer experience is driven by software. Software development processes therefore must be orientated towards customer experience. Product Management must listen to the Voice of the Customer (VoC), extract product requirements, and guide the development team accordingly. The customer is not available for explaining experience until it is too late and the product already developed. Traditional questionnaires, customer surveys and sensing groups do not work for analyzing customer experience. However, with the arrival of social media, new sources of customer experience are available that require new methods for analyzing customer's voice. Six Sigma transfer functions provide the methods needed. This paper presents a case study how to use Six Sigma transfer functions based on Net Promoter® Score for Voice of the Customer for product requirements gathering in a software requirement elicitation processes. Transfer functions are an advancement of *Quality Function Deployment* (QFD) and *Design for Six Sigma* (DfSS), based on Eigenvector search techniques.

**Keywords:** Six Sigma, Transfer Functions, Design for Six Sigma, Quality Function Deployment, Requirements Elicitation, Customer Orientation, Eigenvector Search Techniques.

## 1 Introduction

Net Promoter® Score surveys (NPS) have become increasingly popular because they focus on the Ultimate Question: "How likely are you to recommend our product or service to your friends and relatives?" (Reichheld, 2007). The answer to this question is required on a scale ranging from zero to ten, representing the probability in steps of 10% that the respondent actually will recommend. Moreover, the respondent is asked to point out the reasons why he or she considers recommending or not. Thus NPS captures customer's voice, avoiding the trap of sending verbose questionnaires that reflect the viewpoint of the supplier only. NPS surveys can be made easily through social media channels. (Fehlman & Kranich, Social Media Metrices for Embedded Software, 2012)

Respondents giving a score of 9 or 10 are called Promoters; those scoring with 7 or 8 are Passives and the rest are Detractors – reflecting their likeliness to recommend.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 300–312, 2013.

The NPS is the difference between the percentages of promoters and detractors. According Reichheld (Reichheld, 2007), NPS reliably predicts the likeliness of purchasing upgrades and new products and services in the future, and thus serves as customer loyalty metrics. While NPS is well established in many industries, and part of Six Sigma Best Practices, for software industry it is relatively new. Many software product managers use surveys with a classical feature collection approach, missing an opportunity for listening to customer’s voice, and consequently having no guidance how to cope with the need for providing services to new media.

An overview of available literature and how to use NPS as a Six Sigma Voice of the Customer approach can be found in. (Fehlmann & Kranich, Using Six Sigma Transfer Function for Analysing Customer‟s Voice, 2012) For Six Sigma Transfer Functions, consult Hu and Antony (Hu & Antony, 2007) or the authors in (Fehlmann & Kranich, Transfer Functions, Eigenvectors and QFD in Concert, 2011). Transfer functions are widely used for Google’s search algorithm, see (Gallardo, 2007) and (Kressner, 2005), and in statistical process control (Fehlmann T., Statistical Process Control for Software Development – Six Sigma for software revisited, 2006).

# 2 Six Sigma Transfer Functions for NPS

## 2.1 The Case Study

This case study is from a supplier of customer communication software systems who does regular NPS surveys. Its customers are businesses doing customer communications, with personalized marketing, but also with transactional messages such as credit card or phone bills. Voice of the Customer (VoC) means listening to the organizations using the product; surveying end user consumers is less rewarding as they often are not aware of the communication software product behind their personalized communication with the organization.

**Table 1.** Sample NPS Profile According Customer Segments

| | | *NPS* | *Profile* |
|---|---|---|---|
| EP1 | Enterprise Decider | 40% | 0.39 |
| EP2 | Enterprise Influencer | 38% | 0.38 |
| EP3 | Enterprise User | 50% | 0.49 |
| SP1 | Specialist Decider | 50% | 0.49 |
| SP2 | Specialist Influencer | 31% | 0.31 |
| SP3 | Specialist User | 38% | 0.37 |

An NPS survey addresses the products’ user community by surveys of the willingness to recommend this product by asking managers, influencers, and product users. Typical customers include banks and insurances, providing statements; telecoms and utilities, providing bills; postal services, providing online mail, bill presentment and payment service.

The NPS profile is built on three customer types: deciders, influencers, and users; in this particular case from two different industries: enterprises where ICT integrates all aspects of customer communication, and specialized service providers that serve many smaller organizations with dedicated customer communication. This yields six customer segments. The results of the NPS survey are shown in **Table 1**, and the transfer function analysis is based on seven business drivers shown in **Table 2**. The claim is that these seven business drivers explain the observed NPS response for the six customer segments.

**Table 2.** Sample Business Drivers for high NPS for Customer Communication Software

| | *Topics* | *Attributes* |
|---|---|---|
| C1 | Technical Usability | Performs the intended tasks, ease of use, human interface design |
| C2 | Service Integration | Ease of integration, interoperability, servicification, installation, cloud services |
| C3 | Mobile Platforms | Support of mobile platforms, automation, flexible media formats |
| P1 | Deployment & Licensing | Servitization; pricing & licensing schemes meet the needs of the customer |
| P2 | Process Excellence | Ease of doing business; process excellence, Six Sigma |
| Q1 | Time to Market | Service Quality - Always be first to make the product work |
| Q2 | Fitness for Purpose | Product Quality - No bugs, no recalls, just working fine |

An NPS of 40% – 50% is very high for a supplier of software. The company enjoys overall healthy growth and leaves competition behind. Servitization is the innovation of organization's capabilities and processes to better create mutual value through a shift from selling product to selling Product-Service Systems (Vandermerwe & Rada, 1988). Servitization describes the trend in product management that leads to the *Internet of Things* (López, Ranasinghe, Harrison, & McFarlane, 2013). The NPS profile in **Table 1** is calculated by normalization (Fehlmann T. M., The Impact of Linear Algebra on QFD, 2004). The suspected business drivers are listed in **Table 2**.

## 2.2 Importance

The transfer function looks as shown in **Fig. 1**. In terms of statistical indicators, the convergence gap of 0.28 is the vector difference between the measured NPS profile and the validation profile based on measurement with a confidence interval of 93% is just within limits. The confidence interval is an indication for the measurement error. The convergence gap is less than one third of the unit vector. Measured NPS profile and validation profile almost coincide – see graph to the right of **Table 1**.

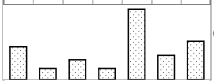
The calculated priority profile of the seven business drivers provide valuable guidance both for managers and for developers with its strong focus on P2: Process Excellence and Q2: Fitness for Purpose. The signal detected is clear enough. The details of how to calculate solutions to transfer functions can be found in (Fehlmann & Kranich, Transfer Functions, Eigenvectors and QFD in Concert, 2011). The numbers in the matrix cells indicate on this scale the frequency of mentioning the respective business driver as reason for the chosen score on a scale 0 to 9.

## 2.3 Satisfaction

Analyzing satisfaction is somewhat more intrinsic: Satisfaction can become negative and resulting profile vectors as well, see (Fehlmann & Kranich, Social Media Metrics for Embedded Software, 2012). In this sample case, negative satisfaction is not abundant, the condition to calculate the solution still holds, and the satisfaction analysis shows an interesting result, see **Fig. 2**. The convergence gap of 0.30 indicates that satisfaction is an equally good explanation for the observed NPS per customer segment; satisfaction with the business drivers important for the likeliness to recommend is equally high. However, it can be seen that customers have an issue with P1: Deployment & Licensing, and with C2: Service Integration.

**Business Drivers Importance Transfer Function**

| Customer Segments | NPS Profile | C1 Technical Usability | C2 Service Integration | C3 Mobile Platforms | P1 Deployment & Licensing | P2 Process Excellence | Q1 Time to Market | Q2 Fitness for Purpose | Explained Profile |
|---|---|---|---|---|---|---|---|---|---|
| EP1 Enterprise Decider | 0.39 | 6 | 2 | 1 | 1 | 9 | 2 | 2 | 0.54 |
| EP2 Enterprise Influencer | 0.38 | 1 | | 3 | 2 | 8 | 3 | 5 | 0.50 |
| EP3 Enterprise User | 0.49 | | | | | 7 | 5 | 5 | 0.42 |
| SP1 Specialist Decider | 0.49 | 3 | 1 | 1 | 1 | 4 | | 3 | 0.30 |
| SP2 Specialist Influencer | 0.31 | 4 | 1 | 3 | 1 | 3 | 1 | 3 | 0.29 |
| SP3 Specialist User | 0.37 | 5 | 1 | 3 | 1 | 4 | 1 | 3 | 0.35 |
| Eigencontrols for Business Drivers | | 0.34 | 0.12 | 0.21 | 0.11 | 0.76 | 0.26 | 0.41 | Convergence Gap 0.28 |

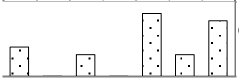


0.20 Convergence Range
0.30 Convergence Limit
Confidence Interval 93%

**Fig. 1.** Importance Transfer Function

**Business Drivers**

| NPS Profile | C1 Technical Usability | C2 Service Integration | C3 Mobile Platforms | P1 Deployment & Licensing | P2 Process Excellence | Q1 Time to Market | Q2 Fitness for Purpose | Explained Profile |
|---|---|---|---|---|---|---|---|---|
| 0.39 | 4 | -3 | 1 | -1 | 3 | 3 | 3 | 0.33 |
| 0.38 | 1 | | 4 | | 8 | -1 | 6 | 0.55 |
| 0.49 | | | | | 9 | 6 | 6 | 0.60 |
| 0.49 | 3 | 0 | 2 | -0 | 4 | | 4 | 0.35 |
| 0.31 | 4 | 1 | 2 | -0 | 2 | -1 | 1 | 0.20 |
| 0.37 | 6 | 2 | 2 | | | 1 | 4 | 0.26 |
| ivers | 0.31 | -0.01 | 0.23 | -0.04 | 0.67 | 0.22 | 0.59 | Convergence Gap 0.30 |



Confidence Interval 91%

**Fig. 2.** Satisfaction Transfer Function

## 2.4 Combining Importance and Satisfaction

Neither importance nor satisfaction alone provides a valid profile for the business drivers. This is because NPS asks for reasons to recommend; if something is definitely not a reason to recommend it won't appear in a positive comment. It still might appear in a negative comment; thus the satisfaction profile should be taken into account for the total weight of business drivers. This is done by combining the importance profile with the *Satisfaction Gap* rather than with the satisfaction profile. The satisfaction gap is weighting negative statements exponentially and thus stretching the importance profile. The formula is component-wise:

$$Combined\ Profile = \boldsymbol{x_I} + e^{-a\boldsymbol{x_S}} \tag{1}$$

Negative satisfaction adds more weight to the importance profile $\boldsymbol{x_I}$ by turning negative satisfaction $\boldsymbol{x_S}$ into additional profile weight for business drivers. The normalization parameter $a$ in (1) must be defined such that the highest satisfaction profile component adds nothing to the combined profile, thus formula (1) corrects missing importance for missing business drivers in a product or service. The satisfaction profile must impact importance only in case of overwhelming dissatisfaction.

***Business Drivers - Profile***
***Combined from NPS Survey***

| | Business Drivers | *Management Adjustments* | *Combined Profile* | Σ | *Importance* | *Satisfaction Gap* |
|---|---|---|---|---|---|---|
| | | | | | *1* | *1* |
| **C Capability** | C1 Technical Usability | * 1.0 | *0.29* | *0.47* | *0.34* | *0.13* |
| | C2 Service Integration | * 1.0 | *0.47* | *0.76* | *0.12* | *0.64* |
| | C3 Mobile Platforms | * 1.0 | *0.24* | *0.40* | *0.21* | *0.19* |
| **P Processes** | P1 Deployment & Licensing | * 1.0 | *0.51* | *0.82* | *0.11* | *0.71* |
| | P2 Process Excellence | * 1.0 | *0.48* | *0.78* | *0.76* | *0.02* |
| **Q Quality** | Q1 Time to Market | * 1.0 | *0.28* | *0.46* | *0.26* | *0.20* |
| | Q2 Fitness for Purpose | * 1.0 | *0.27* | *0.45* | *0.41* | *0.03* |

**Fig. 3.** Combining Importance and Satisfaction

The formula (1) adds weight to those business drivers that customers are not satisfied with. In this case, the drivers P1: Deployment & Licensing and C2: Service Integration gain considerable weight (**Fig. 3**); P2: Process Excellence and Q2: Fitness for Purpose almost none. The resulting business driver combined profile shows the priorities needed for future products to become successful. Note that importance alone explains why the customer recommends, but non-importance doesn't mean that such topics are not relevant to customers, especially not if they show dissatisfaction.

## 3 The "House of Quality" – The 2nd Transfer Function

Our supplier of customer communication software systems developed a new tool allowing organizations to interactively connect to its customers by providing two-way personalized business communication messages between the organization and its customers, integrated with *Customer Relationship Management* (CRM). Sample users of this product may include banks doing wealth management allowing private investors to initiate and approve stock transactions and other investments through a network of mobile apps or slower, more traditional communication means such as phone calls. Other users of the software may use the same product to connect with their regular patrons to book vacations and wellness weekends, or opera houses may use it for their registered visitors when booking seats in performances. The product has a wide range of applicability and serves very different industries. It seems to extend the product portfolio of the software house in a very sensible way.

The business drivers that have been the controls of the NPS analysis become now the intended response for the second transfer function, called ***T***. The transfer function ***T*** looks like the well-known House of Quality (HoQ) in QFD – however, VoC is replaced by the business drivers' profile and VoE by user requirements. The matrix cell elements of ***T*** are measurable also; you can assess how much a particular user requirement supports one of the business drivers. This is a product feature measurement, often done as a prediction during a QFD workshop by experts. During development or for the finished product, the HoQ matrix is measured using the Buglione-Trudel Matrix, see section 3.5.

### 3.1 User Requirements as Controls

The initial capabilities required by these stakeholders are listed in **Table 3**:

**Table 3.** Selected User Requirements (Capabilities) for Interactive Customer Communication

| *Capability Requestor* | | *User Requirements* |
|---|---|---|
| *RC1* | *C-Level Executive* | *Support brand recognition consistently in the product*<br>*Increase customer loyalty*<br>*Differentiate company through personalized service* |
| *RC2* | *Financial Manager* | *Leverage best channels to control delivery costs*<br>*Benefit from new revenue streams*<br>*On demand production replaces stock inventory* |
| *RC3* | *ICT Operator* | *Only one deployment for all platforms*<br>*Flexible open interface*<br>*Automated installation and maintenance processes* |
| *RC4* | *Marketing Professional* | *Exploit opportunities to sell similar products*<br>*Include e-Marketing, mobile marketing*<br>*Central marketing assets management* |
| *RC5* | *Compliance Officer* | *Approval workflow for marketing messages*<br>*Auditable approval cycle*<br>*Monitor and track editing access* |

### 3.2 A Failed Product

Unfortunately, the first version of that product, meeting requirements RC1 to RC5, did not fulfill the expectations of customers, and flopped. Why becomes apparent when looking at the business drivers' profiles detected by the NPS survey. Trying to match the product features with the targets set by the business drivers, as shown in the HoQ in **Fig. 4**, the convergence gap of 0.38 shows limited support only for the required business driver.

This example from software product management shall explain why Six Sigma professionals use transfer functions to help product managers making the right decisions. Others can be found in (Fehlmann & Kranich, Using Six Sigma Transfer Function for Analysing Customer‟s Voice, 2012), (Fehlmann & Kranich, Transfer Functions, Eigenvectors and QFD in Concert, 2011), (Fehlmann T. , Statistical Process Control for Software Development – Six Sigma for Software revisited, 2006), (Akao, 1990), and (Denney, 2005).

### 3.3 Detecting Missing Requirements

The missing required capabilities originate from missing stakeholders. The missing requirements are those of RC6: Technology Officer and RC7: Information Officer, as listed in **Table 4**:

**Table 4.** Missing Capabilities for Interactive Customer Communication

| *Capability Requestor* | | *User Requirements* |
|---|---|---|
| *RC6* | *Technology Officer* | *Service Oriented Architecture - SaaS*<br>*Ability to plug-in Bring-Your-Own-Device*<br>*Industry Standards - focus on HTML 5* |
| *RC7* | *Information Officer* | *Customer Identity Management*<br>*Recognizing customers in all media*<br>*Track communications in all media* |

Adding required capabilities from those additional stakeholders to the House of Quality resolves the problem, as shown below. Transfer functions cannot detect missing requirements be itself, they only validate whether the analysis is correct or not. In reality, once the gap becomes apparent, filling it is rather straightforward for people knowing the domain subject.

While **Fig. 4** shows measured impact in each cell, the two new rows in **Fig. 5** reflect the product improvement design. Thus their cell values show the QFD workshop scale 0-1-3-9 without intermediary values because they are not yet measurable with the Buglione-Trudel matrix. Thus requirements focus shifts from RC4: Marketing Professional to RC7: Information Officer.

The QFD matrix provides work instructions what the software developers shall do to improve the product. Cell entries show the impact of work on required capabilities per business driver. The control profiles in **Fig. 4** and **Fig. 5** consequently indicate the total impact of product development, or product improvement, relatively to each capability requestor. Note that adding requirements from the two new requestors in **Fig. 5** does not simply increase the weight of the C3: Mobile Platforms business

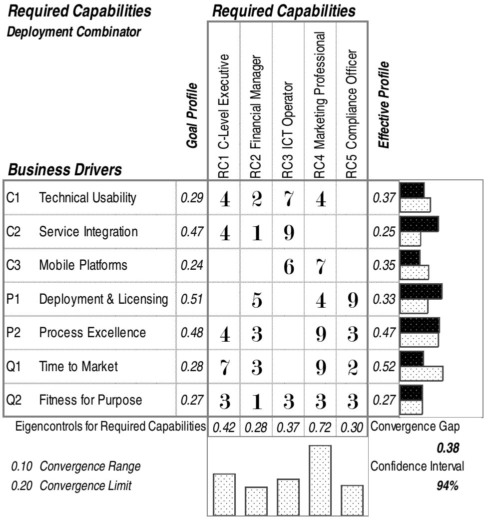



**Fig. 4.** Initial HoQ Detecting Missing Capabilities

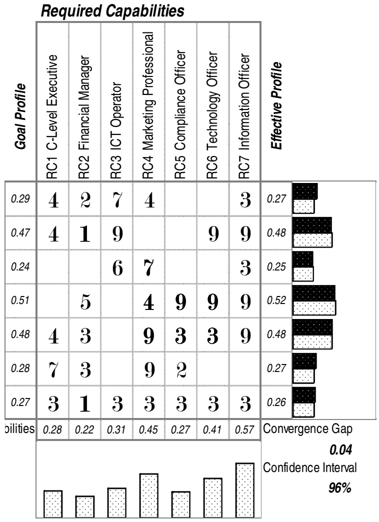



**Fig. 5.** Final HoQ Completed

driver. The product capability focus gains on C2: Service Integration and loses on Q1: Time to Market. Indeed, this is a novel insight on what is needed to make the new product succeed. It means: the intuitive approach, namely simply adding support for C3: Mobile Platforms is not the right strategy. In order to successful expand the reach to Mobile, C2: Service Integration is the key.

### 3.4 User Requirements as Controls

User requirements must be transferred to *Use Cases* (Ambler, 2004), or *User Stories* (Cohn, 2005), whatever is appropriate for the actually applied development methodology to make the implemented product meet customer's expectation. When writing down use cases, it becomes clear that they imply quite a number of functional and non-functional technical requirements. *Functional User* Requirements (FUR), such as the need to access recorded profiles and the SIM card are mixed with *Non-Functional Requirements* (nFURs), such as making the recognition, identification and authentication process trustworthy, transparent and highly reliable. Both FURs and nFURs have significant impact on software development. We call these the *Work Tickets* inferred from a user requirement. Work tickets are common constructs in software development, be it agile or plan-driven.

While identifying the FURs and the corresponding data movements is relatively straightforward – typically it's done when sizing the product according the standard ISO/IEC 19761 COSMIC using sequence diagramming (see (Fehlmann & Kranich, COSMIC Functional Sizing based on UML Sequence Diagrams, 2011), following

Ambler (Ambler, 2004) and Abran (Abran, 2009)) – there are more choices and decisions needed for selecting the nFURs. In an agile environment, the development team is included into this decision, and it can adapt to changing requirements and new priorities. More on lean software development techniques based on the Buglione-Trudel matrix can be found in (Fehlmann T. , Measuring and Estimating Ongoing Agile Projects in Real-Time, 2011) and (Fehlmann T. , Agile Software Projects with Six Sigma, 2011).

For this case study we concentrate on the interesting point, how requirements from RC7: Information Officer support C2: Service Integration. This is the last cell in the second matrix row. The user requirements in **Table 4** mention "Customer Identity Management", "Recognizing customers in all media", and "Track communications in all media". Which Use Cases do implement these requirements?

There are several. Among them the most important might be stated as follows (in the Grant Rule format for User Stories (Buglione & Trudel, 2010), following (Rule, 2010)): "Me as an Information Officer, I want to recognize customers who contact us by phone, e-mail, Facebook, Twitter or else such that I can retrieve them and record their activity on the Corporate CRM". Obviously, such a requirement implies quite a number of FURs and nFURs, among them the service interfaces needed to access phone numbers, SIM card information, e-Mail addresses of senders, credentials in Facebook and Twitter, and more. These service interfaces are FURs that require data movements and can easily be counted and accessed by the ISO/IEC 19761 functional sizing standard "COSMIC" (Abran, 2009). However, none of those FURs has particular impact on business drivers. Functional size, i.e., the number of data movements needed, drives these work tickets (large circle in **Fig. 6**).

The more complicated aspect of this use case is that customers need to agree to such information management. The implied nFURs and FURs might include a customer cockpit (see the small circle in **Fig. 6**) that allows users to connect corporate customer accounts with social media, e-Mail addresses and phone numbers, if they agree to be known to the corporation providing services. Such functionality has impact on the business driver C2: Service Integration. Customers will benefit from integration, but need to agree on it. Functionality does not drive these work tickets, rather usability, although the user cockpit also requires some functionality. The main concern is making complicated service integration topics understandable and acceptable for end customers. This is not only a matter of software development and ergonomic design but as well of psychology and legal user rights, and definitely not captured in functional size measurements.

### 3.5 The Buglione-Trudel Matrix

The Buglione-Trudel matrix is an extended form of the QFD matrix which records all completed and planned use cases (or user stories) as controls, allocating work tickets in matrix cells of the transfer function, and calculates the convergence gap against the business driver's profile. The concept originated from workshop discussions at IWSM/MetriKon/Mensura 2010 in Stuttgart (see (Buglione & Trudel, 2010)).

The rows of the matrix (the $\boldsymbol{y}$-axis) contain the business drivers; each row represents one driver. Below is an unlabeled bottom row, not related to any particular business driver. The columns ($\boldsymbol{x}$-axis) represent the user stories that the developers are expected to implement. Each column contains all work tickets that together make up the use case. The matrix cells contain the work tickets that contribute to a specific business driver. Work tickets that have no specific contribution – like most FURs – are recorded in the bottom row. For instance, accessing user identification in service interfaces has no specific contribution to any of the seven identified business drivers. It's simply needed functionality.

While the FURs carry a functional size, the nFURs typically have no associated functional size. However, all work tickets carry an effort needed for implementation and testing. The Buglione-Trudel matrix is also a tool for estimating development effort (Fehlmann T. , Measuring and Estimating Ongoing Agile Projects in Real-Time, 2011). FURs can be estimated using the ISBSG benchmark database (Hill, 2010).

Thus, two basic metrics are needed to control lean software development: 1) cost & effort, 2) impact on business drivers. Both can be measured; however, impact is a matter of human perception and thus measurement is by getting agreement among experts, sponsors, solution architects, and developers. In practice, the scale of 0-1-3-6 for no, low, medium, or high impact is used.

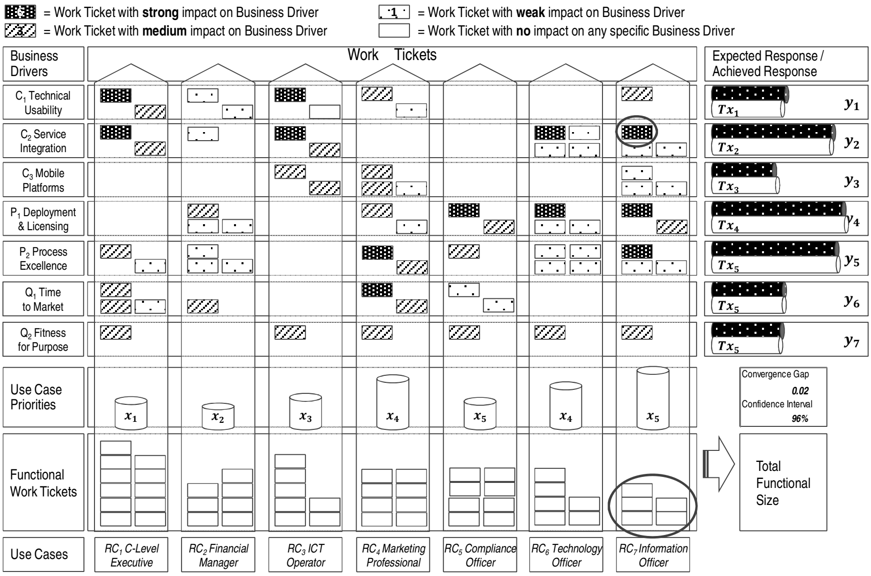



**Fig. 6.** Complete Buglione-Trudel Matrix for Interactive Customer Communication

### 3.6 Linking Developer's Decisions to Business Drivers

Similar to the use of the transfer function when validating NPS analysis, the software development team can validate their development effort. Depending what nFUR they select for implementation, the impact on the business driver profile can immediately be visualized.

**Buglione-Trudel Matrix**
**Deployment Combinator**

| Business Drivers | | Goal Profile | RC1 C-Level Executive | RC2 Financial Manager | RC3 ICT Operator | RC4 Marketing Professional | RC5 Compliance Officer | RC6 Technology Officer | RC7 Information Officer | Effective Profile |
|---|---|---|---|---|---|---|---|---|---|---|
| C1 | Technical Usability | 0.29 | 4 | 2 | 7 | 4 | | | 3 | 0.27 |
| C2 | Service Integration | 0.47 | 4 | 1 | 9 | | | 9 | 8 | 0.46 |
| C3 | Mobile Platforms | 0.24 | | | 6 | 7 | | | 3 | 0.26 |
| P1 | Deployment & Licensing | 0.51 | | 5 | | 4 | 9 | 8 | 9 | 0.51 |
| P2 | Process Excellence | 0.48 | 4 | 3 | | 9 | 3 | 4 | 8 | 0.48 |
| Q1 | Time to Market | 0.28 | 7 | 3 | | 9 | 2 | | | 0.28 |
| Q2 | Fitness for Purpose | 0.27 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 0.27 |
| Eigencontrols for Required Capabilities | | | 0.29 | 0.23 | 0.31 | 0.47 | 0.28 | 0.41 | 0.55 | Convergence Gap 0.02 |

*Required Capabilities*

0.10 *Convergence Range*
0.20 *Convergence Limit*

Confidence Interval **96%**

**Fig. 7.** QFD Matrix for Tracking Priorities

Development teams can control achievements in real-time, by tracking the response $\boldsymbol{Tx}$ of the software development process $\boldsymbol{T}$, where $\boldsymbol{x} = \langle x_1, x_2, \dots, x_n \rangle$ is the use case profile, from delivering the expected contribution profile $\boldsymbol{y} = \langle y_1, y_2, \dots, y_n \rangle$ to the business drivers. The difference is the Euclidian distance between two tors $\|\boldsymbol{y} - \boldsymbol{Tx}\|$, called the Convergence Gap. $\boldsymbol{Tx}$ will not match $\boldsymbol{y}$ exactly; however, the gap measures whether the software development project meets customer expectations. The QFD matrix in **Fig. 7** shows the achieved software product profile, measured by the impact of work tickets in the Buglione-Trudel matrix **Fig. 6**.

## 3.7 Keeping Moving Targets under Control

Thanks to the Six Sigma measurement approach, variation can be kept under control. It is immediately seen when changes to the work plan affect the convergence gap. It is possible to immediately consider necessary corrections to the plan, the features, and the qualities being implemented in the software product. Change is no longer a threat but becomes an opportunity for delivering the software product right the first time. Agile is instrumental for Lean Six Sigma software development (Poppendieck, 2007).

Moreover, if some business drivers are not covered by the planned work tickets, i.e., the convergence gap is high and some business driver lacks support, the team visually perceives the missing pieces and stimulates creativity how to close the gap. It is straightforward to explain to product sponsors, or other stakeholders, why such additional work must be done, and how it adds value, since its impact must be discussed and agreed while updating the Buglione-Trudel matrix. This process makes the software development process both lean and agile (Fehlmann T. , Agile Software Projects with Six Sigma, 2011).

## 4 Conclusions

It has been shown how transfer functions in the form of the Buglione-Trudel matrix **Fig. 6** are handy tools for harvesting knowledge from development teams during development, and allow in the form of traditional QFD matrices analyzing NPS surveys for successful product strategies. Whoever masters the flow of such information into the development process has an immediate advantage upon competition, allowing them to embrace new technologies as they emerge.

The techniques used are well known: QFD is around for some thirty years, DfSS and NPS both for some ten years or more; its work in concert must yet be trained, and basic knowledge about how to use and calculate transfer functions seems still missing in the software community, despite attempts even by very traditional institutions like the SEI to promote Six Sigma for Software. Experience how to develop software products based on QFD is available and known.

However, integrating live customer survey information into product development processes is an important step forward.

## References

[1] Reichheld, F.: The Ultimate Question: Driving Good Profits and True Growth. Harvard Business School Press, Boston (2007)
[2] Fehlman, T.M., Kranich, E.: Social Media Metrics for Embedded Software. In: Tagungsband MetriKon, Stuttgart, Germany (2012)
[3] Fehlman, T., Kranich, E.: Using Six Sigma Transfer Function for Analysing Customer‟s Voice. In: Proceedings of the 4th International Conference on Lean Six Sigma, Glasgow, UK (2012)
[4] Hu, M., Antony, J.: Enhancing Design Decision-Making through Development of Proper Transfer Function in Design for Six Sigma Framework. International Journal of Six Sigma and Competitive Advantage (3), 33–55 (2007)
[5] Fehlman, T.M., Kranich, E.: Transfer Functions, Eigenvectors and QFD in Concert. In: Proceedings of the ISQFD 2011, Stuttgart, Germany (2011)
[6] Gallardo, P.F.: Google's Secret and Linear Algebra. EMS Newsletter 63, 10–15 (2007)
[7] Kressner, D.: Language Hierarchies and Interfaces. Lecture Notes in Computational Science and Engineering, vol. 46 (2005)
[8] Fehlman, T.: Statistical Process Control for Software Development – Six Sigma for Software revisited. In: EuroSPI 2006 Industrial Proceedings, Joenssu, FI (2006)
[9] Vandermerwe, S., Rada, J.: Servitization of business: Adding value by adding services. European Management Journal 6(4), 314–324 (1988)
[10] López, T.S., Ranasinghe, D.C., Harrison, M., McFarlane, D.: Using Smart Objects to build the Internet of Things. IEEE Internet Computing (to appear, 2013)
[11] Fehlman, T.M.: The Impact of Linear Algebra on QFD. International Journal of Quality & Reliability Management 21(9), 83–96 (2004)
[12] Akao, Y. (ed.): Quality Function Deployment - Integrating Customer Requirements into Product Design. Productivity Press, Portland (1990)
[13] Denney, R.: Succeeding with Use Cases – Working Smart to Deliver Quality. Booch–Jacobson–Rumbaugh. Addison-Wesley, New York (2005)

[14] Ambler, S.W.: The Object Primer. Agile Model–Driven Development With UML 2.0, 3rd edn. Cambridge University Press, New York (2004)
[15] Cohn, M.: Agile estimating and planning. Prentice Hall, New Jersey (2005)
[16] Fehlman, T.M., Kranich, E.: COSMIC Functional Sizing based on UML Sequence Diagrams. In: MetriKon, Kaiserslautern (2011)
[17] Abran, A.: The COSMIC Functional Size Measurement Method – Version 3.0.1 – Measurement Manual (2009)
[18] Fehlman, T.: Measuring and Estimating Ongoing Agile Projects in Real-Time. In: Proceedings of the UKSMA Conference, London, UK (2011)
[19] Fehlman, T.: Agile Software Projects with Six Sigma. In: Proceedings of the 3rd European Research Conference on Lean Six Sigma, Glasgow, UK (2011)
[20] Buglione, L., Trudel, S.: Guideline for sizing agile projects with COSMIC. In: Proceedings of the IWSM / MetriKon / Mensura, Stuttgart, Germany (2010)
[21] Rule, G.: Sizing User Stories with the COSMIC FSM method," (2010), `http://www.smsexemplar.com/wp-content/uploads/20100408-COSMICstories-article-v0c1.pdf` (accessed April 3, 2013)
[22] Hill, P. (ed.): Practical Software Project Estimation, 3rd edn. McGraw-Hill (2010)
[23] Poppendieck, M.T.: Implementing Lean Software Development. Addison-Wesley, New Yor (2007)

# Framework to Assist Healthcare Delivery Organisations and Medical Device Manufacturers Establish Security Assurance for Networked Medical Devices

Anita Finnegan, Fergal McCaffery, and Gerry Coleman

Regulated Software Research Centre, Dundalk Institute of Technology & Lero, Dundalk, Co Louth, Ireland
{anita.finnegan,fergal.mccaffery,gerry.coleman}@dkit.ie

**Abstract.** This paper introduces an assurance framework for networked medical device development. This work is being conducted to address the ever-increasing concerns of medical device security with a specific focus on medical devices to be incorporated into IT networks. The framework utilises a Process Assessment Model and a Process Reference Model to address system development lifecycle processes, security assurance processes and a focused risk management process. There is currently no governance for the development of secure medical devices in place and so, this work sets out to resolve this problem by increasing the awareness of medical device security risks, threats and vulnerabilities among Medical Device Manufacturers, IT vendors and Healthcare Delivery Organisations.

**Keywords:** Security Assurance, Networked Medical Device Security, Process Assessment Model, Process Reference Model, Security Capabilities.

## 1 Introduction

Security of medical devices is a very serious and concerning topic among the medical device domain at present so much so that it has been elevated with the involvement of US Government bodies. One reason for this concern is due to advancements in the design of medical devices in recent years. The introduction of software and then the introduction of interoperable and networked medical devices have presented significant benefits for Healthcare Delivery Organisations (HDOs) and for patient care. The design and functionality of these devices have changed tremendously in the last number of years. However, the development processes have remained unchanged and consideration for new types of risks for such devices with communication capabilities has not yet been adequately built into the development life cycle. This work sets out to change this and to overcome gaps in the development life cycle where security requirements need to be prioritised. This work introduces a Process Assessment Model (PAM) that incorporates the system development life cycle processes and builds upon this to add further assurance for these processes. It then incorporates a very focused security risk management process with a specific set of security controls, requirements and capabilities for consideration.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 313–322, 2013.

ISO/IEC 15504-2 [1] is an international standard that is often used in the IT and software industry to establish an organisations ability to achieve a particular process or set of processes. It provides a measurement framework for process capabilities and defines the requirements for performing the assessment. In utilising ISO/IEC 15504 the three major outputs are the Process Reference Model (PRMs), PAM and a capability measurement of the assessed processes. Existing generic Software Process Improvement (SPI) models are available which include the Capability Maturity Model Integration (CMMI®) [2] and ISO 15504-6:2006 [3] (SPICE) however these models were not developed to provide sufficient coverage of all areas necessary to assure the security of medical devices being incorporated into an IT network [4]. We achieve this through the development and implementation of an enhanced Process Reference Model (PRM), a Process Assessment Model (PAM) (including a Process Measurement Framework in compliance with IEC/ISO 15504-2 [1]) for the assurance of Medical Device Manufacturers (MDMs) development processes. It is intended that this will impact MDMs in their design decisions during the development of networked medical devices. In developing this framework, another key objective is to strengthen the relationship between MDMs and HDOs with involvement of HDO IT administration staff during the planning stage. This communication will assist MDMs better understand the environment, the intended use and the users of the medical device and, through a predefined set of security capabilities, the HDO will be able to better communicate the security requirements for a particular medical device.

This research aims to address security in networked medical devices and to build an awareness of the types of security vulnerabilities and threats that can negatively impact the safety of patients through the development of a focused security risk management process. Section 1.1 discusses the background to this problem, the reason for this work, and the approach taken. Section two describes process assurance and discusses key standards. Section three concludes the paper and details the expected impact this research will have upon the medical device industry (including the HDOs, MDMs) and in terms of regulatory compliance assessments.

### 1.1 Background

Medical device design innovations over the last number of years have provided significant benefits for patient care and healthcare providers. An increased use of software has allowed MDMs to add sophisticated functionality to devices. More recently medical devices include functionality to communicate via healthcare IT networks, wirelessly, across the Internet and from device to device. Networked medical devices can now provide patients with around-the-clock care outside the healthcare environment. Resource demand for HDOs to administer this care is also significantly reduced. HDOs utilize a wide range of networked devices from hard-wired monitoring devices such as diagnostic equipment (CT scanners) to implanted medical devices such as defibrillators. Clearly the benefits of networking these devices are significant but, in using such technology, a new set of risks arise associated with their use. These are security risks, threats and vulnerabilities. In the last 12 months there have been many published reports highlighting the vulnerabilities of networked medical devices. One report issued by the Department of Homeland

Security [5] highlights common threats associated with each type of device (implantable, external and portable medical devices). As this technology is relatively new, there is fear within the healthcare industry that the security of medical devices is insufficient and has not been thoroughly addressed in terms of research and design. More concerning is that malicious attackers have not yet fully exploited these devices but they do possess the potential to do so. This became evident through a number of controlled hacking demonstrations where security researchers proved the vulnerability of medical devices. One such incident was at the 2011 Black Hat Security Conference in Las Vegas where, a diabetic security researcher, Jerome Radcliffe, hacked his own insulin pump. This enabled him to increase and decrease the dosage levels without a warning that either, the pump had been tampered with or that the dosage levels may be harmful to him. More recently, researchers from Cylance, a stealth security firm based in Irvine, California, hacked into Philips XPER medical management system and allowed them to take control of other pieces of connected equipment [6]. This raised a lot of concern within the medical device domain and led to the interjection of the US government, which prompted a US Government Accountability Office (GAO) inquiry into the FDA's assessment of medical devices in terms of security. The outcome of this was a report published in August 2012 [7] detailing the lack of consideration for both intentional and non-intentional security vulnerabilities during the FDA's PMA and 510k approval processes. This paper outlines work that addresses security issues for medical devices to be incorporated into an IT network. The remainder of section 1 presents an overview of this research and also the approach to address This round of checking takes place about two weeks after the files have been sent to the Editorial by the Contact Volume Editor, i.e. roughly seven weeks before the start of the conference for conference proceedings, or seven weeks before the volume leaves the printer's, for post-proceedings. If SPS does not receive a reply from a particular contact author, within the timeframe given, then it is presumed that the author has found no errors in the paper. The tight publication schedule of LNCS does not allow SPS to send reminders or search for alternative email addresses on the Internet.

### 1.2 Framework Development – The Approach

The first step in this approach was to select a suitable PRM to build the PAM upon. A system life cycle process standard was most suitable as a foundation for the PAM as it addresses the life cycle of a system (including hardware and software), in 25 processes, from concept through to retirement. In order to place emphasis on security, it was felt that further assurance of particular development processes was required so the PAM was tailored to include additional processes, activities and tasks from another standard. This standard specifically addresses assurance in the system life cycle based on a selected critical property of a system (i.e. dependability, safety, security etc.).

As one of the main objectives of this work is to provide MDMs with a focused security risk management process we have facilitated this by furthering enhancing the PAM to include a list of security controls to be addressed during the development life cycle of the system. In order to achieve this, a security standard review has been performed. A complete set of controls deemed relevant to these types of medical devices were devised and validated through the use of expert opinion, interested parties within the FDA and the International medical device standards committee (i.e. IEC SC62A JWG7). The outcome of this exercise is a technical report presenting these security controls. This will be raised as a new work item in May 2013 at the IEC SC62A JWG7 International standards meeting. In addition to this another technical report will be published to provide guidance to MDMs for the implementation of the PAM. Upon the preliminary completion of this framework, it will be trialed within MDMs and HDOs within both the EU and the US.

Figure 1 details the overview of this framework for addressing security in the development life cycle stages for networked medical devices.

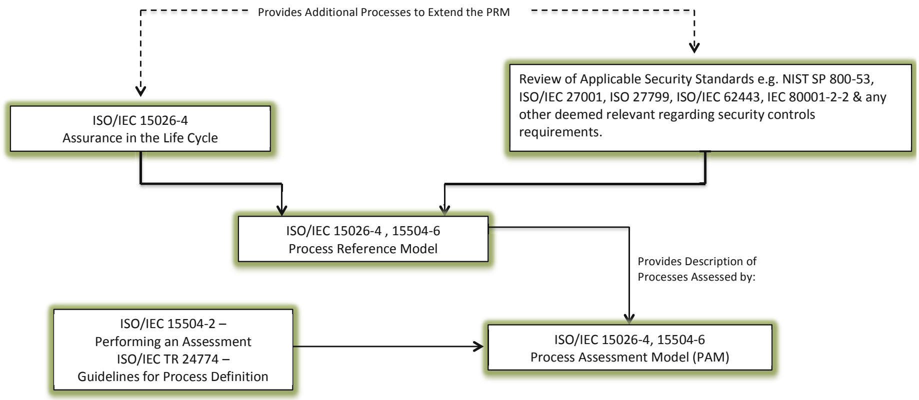



**Fig. 1.** Process Assurance Overview

# 2 Security Process Assurance

## 2.1 ISO/IEC 15504 – Process Assessment Model

The International standard for Software Process Improvement and Capability determination (ISO/IEC 15504) will be utilized to establish the development process capability level. Compliance with IEC/ISO 15504 results in the following outputs; a PRM and a PAM (including an aligned Measurement Framework). The PAM contains two dimensions, which are the Process Dimension and the Capability Dimension. The Process Dimension is developed from an external PRM that presents the processes for assessment in terms of their 'Purpose' and 'Outcome'. The PRM helps support process analysis and design activities as it provides a set of descriptions of the processes to be assessed. The PAM expands the PRM with the use of a set of Performance Indicators called Base Practices and Work Products. The Performance Indicators vary from process to process. Work Products are both, inputs to a process and also the outputs produced by a process. The Work Product Performance

Indicators are the results of performing the process and are used to review the effectiveness of each process. Base Practices are the actions taken to transform the inputs into outputs addressing the purpose of the process. They describe 'what' should be done in order to address the process but do not detail 'how' it should be done. The Base Practices are the basic required activities that specifically address the process purpose. Combined evidence of Work Practice characteristics and the performance of Base Practices provide the objective evidence of achievement of the 'Process Purpose'.

ISO/IEC 15504-2 [1] sets out a Capability dimension that utilizes six Capability Levels from Level 0, 'Incomplete' to Level 5, 'Optimizing'. ISO/IEC 15504-2 defines the measurement framework based upon a set of 9 Process Attributes associated with Levels one through to five. These Process Attributes represent measurable characteristics required to manage and improve each process. The extent of achievement of each attribute is defined on a rating scale. In ISO/IEC 15504-6, these Process Attributes include Generic Practices and Generic Work Products that belong to a set of Process Capability Indicators. These indicators provide the means of achievement of the capability addressed by each of the Process Attributes within each of the associated Capability Levels.

For the solution, the most suitable PRM is defined in ISO/IEC 15288 – *Systems Engineering – System Life Cycle Processes* [8] and forms the foundation for the PAM. ISO/IEC 15288 provides a process framework that covers the entire life cycle of systems from cradle to retirement. A system development life cycle standard is most applicable to networked medical devices as these devices may contain one or more of the following: *"Software, hardware, humans, processes (e.g. review processes), procedures (e.g. operator instructions), facilities and natural occurring entities (e.g. water, organisms, minerals)".*

Due to the fact that ISO/IEC 15504-6 [3] uses ISO/IEC 15288 as the external PRM, this was then selected as a suitable foundation for the PAM. ISO/IEC 15504-6 details an exemplar PAM that also includes the process attributes that are compliant with ISO/IEC 15504-2. The Process Dimension utilizes the processes as defined in ISO/IEC 15288 and divides these into four groups which are the Agreement, Enterprise, Project and Technical processes. While the foundation PRM and the PAM framework addresses the entire system life cycle it has been extended for the inclusion of additional processes from ISO/IEC 15026-4 [9]. These processes are included as a measure to address security assurance of networked medical devices. This is discussed in the following section.

### 2.2 Building Additional Assurance into the PAM

Due to advancements in medical device designs and the fact that it is now proven that networked and interoperable medical devices are open to malicious attack, additional steps are required during the development life cycle to address security. An emphasis on security is required and has been achieved through the inclusion of processes in the PRM from ISO/IEC 15026-4 – *Systems and Software Engineering – Systems and Software assurance – Assurance in the Life Cycle*. ISO/IEC 15026-4 is mainly utilized where additional assurance for a critical property, such as dependability, safety or security, is required for a system or software. The standard is used as an add-on to an already existing life cycle process standard (such as ISO/IEC 15288).

**Table 1.** IEC/TR 80001-2-2 Capabilities

| | Security Capability | Code |
|---|---|---|
| 1 | Automatic Logoff | ALOF |
| 2 | Audit Controls | AUDT |
| 3 | Authorization | AUTH |
| 4 | Configuration of Security Features | CNFS |
| 5 | Cyber Security Product Upgrades | CSUP |
| 6 | Data Backup and Disaster Recovery | DTBK |
| 7 | Emergency Access | EMRG |
| 8 | Health Data De-Identification | DIDT |
| 9 | Health Data Integrity and Authentication | IGAU |
| 10 | Health Data Storage Confidentiality | STCF |
| 11 | Malware Detection/Protection | MLDP |
| 12 | Node Authentication | NAUT |
| 13 | Person Authentication | PAUT |
| 14 | Physical Locks on Device | PLOK |
| 15 | Security Guides | SGUD |
| 16 | System and Application Hardening | SAHD |
| 17 | 3rd Party Components in Product Lifecycle Roadmaps | RDMP |
| 18 | Transmission Confidentiality | TXCF |
| 19 | Transmission Integrity | TXIG |
| 20 | Unique User ID | UUID |

ISO/IEC 15026-4 is an international standard recently published that provides a process framework (Systems Assurance Process View) for software or a system that requires assurance for a particular aspect. This is usually when additional or careful attention is required for a particular system; otherwise known as a critical property. Critical properties are usually associated with substantial risk concerning safety, dependability, and reliability or, as we have adapted, security. The standard presents a set of add-on processes, activities and tasks with guidance and recommendations.

These processes, activities and tasks are intended to build upon the Agreement, Project and Technical processes as set out in ISO/IEC 15288. Therefore, conformance to this standard is achieved through the demonstration of these additional processes as well as conformance with the Agreement, Project and Technical processes of ISO/IEC 15288. For this reason, demonstration of additional assurance specifically addressing security, through the use of this standard, is suited for integration with the Process Assessment Model as set out in ISO/IEC 15504-6. The expected outcomes incorporating processes from IEC/ISO 15026-4 are [9]:

1. A subset of requirements for the achievement of critical properties is defined.
2. Assurance claims, their justification, and the body of information showing the achievement of the assurance claims for the critical properties are established as an element of the system.
3. A strategy for achieving these assurance claims and showing their achievement is defined.
4. The extent of achievement of the assurance claims is communicated to affected stakeholder.

## 3 Security Process Assurance

As we have developed this framework to specifically address security as the system critical property we have enhanced the PAM to focus on the Risk Management Processes where we introduce new considerations to be utilized during risk management activities (Process Reference PRJ.5 from ISO/IEC 15504-6). This paper discusses the security risk management process only and so this is additional to the normal practices for project and product risk management. This subsection looks at security standards and the development of a set of security controls for assuring the security of medical devices that will be validated and approved by medical device security experts in the domain and the FDA.

### 3.1 IEC/TR 80001-2-2

IEC/TR 80001-2-2 - Application of risk management for IT-networks incorporating medical devices - Guidance for the communication of medical device security needs, risks and controls [10] is a technical report which sets out to promote the communication of security controls, needs and risks of medical devices to be incorporated into IT networks between MDMs, IT vendors and HDOs. In this technical report there are a total of 20 security capabilities (Table 1) presented. These security capabilities provide a base template for a HDO to communicate their security requirements for a given medical device based on their needs. Prior to the acquisition of a medical device, HDO IT administrators may use this technical report to assist MDMs in establishing the HDO requirements. The benefit in adapting this approach is that the HDOs then become more aware of their requirements in order to securely incorporate a medical device into their network. It assists MDMs to better understand the intended use and environment in which the medical device will be utilized. However, the security requirements as indicated by the HDO are for guidance purposes only. The MDM will continue to carry out the usual risk analysis steps and upon completion of this will communicate back and agree with the HDO the necessary security capabilities for the product. This technical report will form the foundation for the security risk management process. The 20 security capabilities defined in IEC/TR 80001-2-2 will be included in the risk management process. A set of sub requirements, called Security Capability Requirements (SCRs) for each security capability will be required. These sub requirements present alternatives for implementation of a particular security capability. The security capabilities and their SCRs are intended to act as a template for communicating high level security requirements between the HDOs and MDMs. SCRs for each of the 20 security capabilities in IEC/TR 80001-2-2 have been developed and will be validated through untilising the opinion of expert users, security researchers and also interested personal within the FDA. An example of a set of sub requirements for security capability Automatic Logoff is show in Table 2.

**Table 2.** Security Capability Requirements ALOF ALOF

| Implementation Identifier | Capability |
|---|---|
| ALOF.01 | A screensaver starts automatically 5 minutes after last keystroke/mouse movement operation |
| ALOF.02 | The screensaver clears all displayed health data from the screen. |
| ALOF.03 | The screensaver does not log-off the user / does not terminate the session. |
| ALOF.04 | User has to log-in after occurrence of the screensaver |
| ALOF.05 | The user-session terminates automatically 60 minutes after last keystroke/mouse movement/touchscreen operation. |

ISO/IEC 15504-6, Process PRJ.5 - Risk Management Process, the process purpose is to identify and assess threats and monitor the risks throughout the life cycle. The PAM further builds on this with the inclusion of the Base Practice 'PRJ.4.BP.2: Identify Risks' as a performance indicator. The MDM will conduct the risk assessment, considering the type of networked medical device, the design, its operational environment, the user and the users' needs (as communicated by the HDO). For each of these risks, the following Base Practices must be performed:

- PRJ.4.BP.3 Determine the Risk Occurrence Probability
- PRJ.4.BP.4 Evaluate the Risk Consequence
- PRJ.4.BP.5 Prioritize Risks
- PRJ.4.BP.6 Select Risk Treatment Strategies

The Base Practice PRJ.4.BP.6, Select Risk Treatment Strategy will detail the implementation of the SCRs for each security capability (such as Automatic Log Off, Unique User ID etc.) as communicated and agreed between the HDO and the MDM.

In addition, to the inclusion of the security capabilities presented in IEC/TR 80001-2-2, work has been carried out to survey an array of security standards and best practices. The standards reviewed were ISO/IEC 27001 [11], ISO/IEC 27799 [12], ISO 15408 [13], IEC 62443-3-3 [14] and NIST SP 800-53 [15]. Each of these standards and guidance documents similarly highlight security classes and controls with many repeating controls existing between standards. A security control matrix has been developed to map the controls across each standard and to identify cross over controls. An exhaustive list of security controls from all security standards has been compiled for review in terms of their relevance to networked medical devices. With this complete list of security controls from the above standards, a mapping has been done to link the security capabilities from IEC/TR 80001-2-2 to their attributing security control(s). This will assist with the development of guidance documents for suitable security controls for networked medical devices. In addition to this, a gap analysis is being conducted in order to identify further capabilities/controls that should be included in IEC/TR 80001-2-2. This will be achieved through the use of expert opinion (i.e. expert users from industry and the FDA). The validated security controls, plus the existing IEC/TR 80001-2-2 security capabilities, will form the foundation for the security risk management process. A Technical Report will be published in the coming months detailing this security matrix gap analysis with the anticipation that IEC/TR 80001-2-2 will be revised based on this.

## 4 Conclusions and Future Work

This paper presents a framework for the assurance of networked medical devices in terms of security. The solution combines an array of international standards, guidance documents and processes to create a step-by-step process for MDMs. MDMs will follow this during development to decrease the risk of potential security vulnerabilities associated with the use of networked medical devices. As a PAM forms the foundation of this framework, with an associated measurement framework, it provides great benefits to the FDA and for external assessors in establishing the efficiency, thoroughness and quality of processes used to develop networked medical devices. This also benefits HDO's with supplier selection activities. The approach discussed in this paper focuses on development process assurance with the aim of positively impacting the overall security capability of networked medical devices. The remainder of this section describes the expected outputs from both the process and product assurance components of the approach. The output for the process assurance component is:

1. The development of a PAM based on the international standard ISO/IEC 15504-6 model that has been specifically developed for the international system life cycle process standard, ISO/IEC 15288. This PAM will be extended to include additional processes based on security being the critical property in line with yet another international standard for security assurance in the life cycle, ISO/IEC 15026-4.
2. A published technical report detailing the application and use of this extended PAM.
3. A validated set of applicable and meaningful security controls to be adopted and included in the Risk Management process of the PAM.
4. The publication of a technical report detailing the security controls required for consideration in using this approach. This is fully supported by the FDA and a Standard Committee Conveyor. It is expected that this be prioritised as a new work item within one of the Standard Committee Joint Working Groups. The expectation is the development of an international standard on the basis of this.

This framework will be trialed with MDMs and HDOs in both Europe and the US. Medical device security assurance driven development is a new concept and so future work will be to further build upon this to develop product specific SCRs following the trialing of this with MDMs. Currently there is no method to specifically address security assurance for the development processes for networked medical devices. This is the primary focus of this research and so it is expected that the output of this research will positively impact the medical device domain in both the EU and the US by building awareness of security vulnerabilities, threats and related risks between the HDO and the MDM [4].

**Acknowledgements.** This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 10/CE/I1855.

## References

1. ISO/IEC, 15504-2: 2003 Software Engineering - Process Assessment - Performing an Assessment (2003)
2. SEI, CMMI-DEV, CMMI for Development (2010)
3. ISO/IEC, 15504-6:2008 Information technology — Process assessment — An exemplar system life cycle process assessment model (2008)
4. Finnegan, A., McCaffery, F., Coleman, G.: Development of a process assessment model for assessing security of IT networks incorporating medical devices against ISO/IEC 15026-4. In: Healthinf 2013, Barcelona, Spain, pp. 250–255 (2013)
5. DHS, Attack Surface: Healthcare and Public Heath Sector (2012)
6. Rashid, F.Y.: Researchers Uncover Privilege Escalation Bug in Philips Medical Devices (2013), `http://www.securityweek.com`
7. GAO, Medical Devices, FDA Should Expland Its Consideration of Information Security for Certain Types of Devices (2012)
8. ISO/IEC, 15288 - Systems engineering — System life cycle processes (2008)
9. ISO/IEC, 15026-4: Systems and Software Engineering - Systems and Software Assurance - Assurance in the Life Cycle (2012)
10. IEC, TR 80001-2-2 - Application of risk management for IT-networks incorporating medical devices - Guidance for the disclosure and communication of medical device security needs, risks and control, International Electrotechnical Committee (2011)
11. ISO/IEC, 27001 Information Technology - Security Techniques - Information Security Management Systems - Requirements (2005)
12. ISO, EN ISO 27799:2008 Health informatics. Information security management in health using ISO/IEC 27002 (2008)
13. ISO/IEC, 15408-1 Information Technology - Security Techniques - Evaluation Criteria for IT Security, in Introduction and General Model (2009)
14. IEC, 62443-3-3 – Security for industrial automation and control systems - Network and system security – System security requirements and security assurance levels Introductory Note (2011)
15. NIST, 800-53 Recommended Security Controls for Federal Information Systems and Organisations, U.S.D.o. Commerce, Editor (2009)

# Implementing Functional Safety Standards – Experiences from the Trials about Required Knowledge and Competencies (SafEUr)

Richard Messnarz[1], Christian Kreiner[2], Ovi Bachmann[3], Andreas Riel[4], Klaudia Dussa-Zieger[5], Risto Nevalainen[6], and Serge Tichkiewitch[4]

[1] ISCN LTD, Ireland
rmess@iscn.com
[2] Graz University of Technology, Institute for Technical Informatics, Graz, Austria
christian.kreiner@tugraz.at
[3] SIBAC, Mittelbiberach, Germany
info@sibac.de
[4] EMIRAcle c/o Grenoble Institute of Technology GSCOP UMR5272, Grenoble, France
{andreas.riel,serge.tichkiewitch}@grenoble-inp.fr
[5] Methodpark, Erlangen, Germany
Klaudia.Dussa-Zieger@methodpark.de
[6] Spinet Oy and FISMA, Finland
risto.nevalainen@falconleader.fi

**Abstract.** In the EU project SafEUr (518632-LLP-1-2011-1-AT-LEONARDO-LMP) the partnership developed a skill set with learning objectives, training materials, and tools to teach and coach the implementation of IEC 61508 and ISO 26262. Automotive, Medical, and Nuclear industry gave inputs to the project. A group of above 20 multinational companies (SOQRATES www.soqrates.de) which also are active in automotive industry (some of them represent the largest suppliers in Automotive industry) organised reviews and trial courses with safety managers. This led to a defined set of skills and tools we expect from functional safety managers and functional safety engineers. In this paper we describe the results of SafEUr, the feedback we received from the collaboration with leading automotive industry and the next steps in 2013 to launch this schema with official certificates from end of 2013 onwards.

**Keywords:** Functional Safety Manager, Functional Safety Engineer, Integrated Safety Design and Technical Safety Concept, ECQA, Certification.

## 1 Introduction to SafEUr

Functional safety of modern products and industry systems containing embedded systems has become a first priority in several industrial sectors. The IEC61508 group of standards require companies to have in place "Functional Safety Management". Domain specialized standards like ISO 26262 [1] for the passenger cars complement IEC 61508. The objective of SafEUr was to create a European-wide accredited training

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 323–332, 2013.

and certification program for Functional Safety Managers, based on a skill card which is compliant to the European Qualification Framework. SafEUr [2] delivers modern e-learning based and vocational training, which is based on practical case studies and best industry practices. This training will be complemented by a world-wide unique web-based integration platform for industry and academia in the domain of Embedded Systems. Certified SafEUr trainers are available all across Europe, assuring a major impact and sustainability of this ECQA job role. Results we achieved in the project include:

- Skills set with 15 learning elements / training units
- A pool of test questions
- Training courses and coaching experiences in collaboration with leading automotive industry such as Continental Automotive, ZF Friedrichshafen AG, KTM Sport Motorcycle GmbH., etc.
- Training courses integrated in University Education (Grenoble Institute of Technology, France, and Graz University of Technology, Austria)
- A pool of certified trainers and coaches (in progress)
- A Europe wide Job Role Committee to maintain the profession

Based on SafEUr the EuroSPI community started the Build Up of an Experience Exchange Community in form of a series of international workshops attached with EuroSPI. The Certified Functional Safety Manager [2] follows the ECQA quality procedures and sets up a Europe wide schema in collaboration with ECQA, thus assuring the compliance with European quality standards in training.

The ECQA (European Certification and Qualification Association) [3] has set up defined guidelines and procedures for (see www.ecqa.org, about ECQA, Guidelines):

- Standards about how to define skills sets
- Standards about how to design tests and test questions
- Standards about learning material development
- Standards about certification
- Standards about accreditation of training bodies

To base the Functional Safety approach on a practical set of case studies, a set of success factors has been defined with the support of European leading companies [4], [5], [6] to be considered when applying “Functional Safety”. Also, this industry group [7] has developed an integrated SPICE (ISO 15504) [8] and safety (IEC 61508, ISO 26262) assessment approach. These companies are also members of the functional safety working group of SOQRATES initiative. In addition partners from ISO 15504 Part 10 working party are invited to integrate their approach of an extended safety assessment.

## 2 Functional Safety Manager Skills Set and Knowledge Areas

The skills set has been reviewed at an international workshop at EuroSPI 2012 [2] and further reviewed by an expert team and applied in an industry workshop with leading automotive industry in Feb. 2013. The industry feedback showed that some of our elements are too technical for safety managers. In the industry they use 2 roles, a functional safety manager and a functional safety engineer and they asked the team to split the two roles in the approach. In general the feedback was positive because all the contents supported by real automotive examples and best practices used in lead projects.

| SafEUr Skill Set and Training Units | | |
|---|---|---|
| Content Units of the training and skill card | Functional Safety Manager | Functional safety Engineer |
| Unit 1 - Introduction to Functional Safety Management (0,5 days) | | |
| U1.E1 International Standards and Norms | FSM | |
| U1.E2 Product Life Cycle | FSM | |
| U1.E3 Terminology | FSM | |
| Uni 2 - Management of Functional Safety (1,5 days) | | |
| U2.E1 Safety management on organisational and project level | FSM | |
| U2.E2 Safety Requirements and Safety Case Definition | FSM | |
| U2.E3 Overview of Required Engineering and V&V Methods | FSM | |
| U2.E4 Establish and Maintain Safety Planning | FSM | |
| Unit 3 - Engineering aspects of Functional Safety (2,5 days) | | |
| U3.E1 System Hazard Analysis and Safety Concept | FSM | FSE |
| U3.E2 Integrating Safety in System Design & Test | Only Overview | FSE |
| U3.E3 Integrating Safety in Hardware Design & Test | Only Overview | FSE |
| U3.E4 Integrating Safety in Software Design & Test | Only Overview | FSE |
| Unit 4 - Safety on Product Level | | |
| U4.E1 Reliability design on product and system level | | FSE |
| U4.E2 Safety in the Production,Operation and Maintenance | FSM | |
| Unit 5 - Legal aspects of Safety | | |
| U5.E1 Legal aspects and Liabilities | FSM | |
| U5.E2 Qualification and Certification (previous U2.E5) | FSM | |

**Fig. 1.** SafEUr Skill Set with Functional Safety Manager and Engineer Scope

The role of the functional safety manager relates to the safety planning, the safety life cycle, the safety case and the prove of coverage of the safety case, the coverage of all selected methods required in the method tables of ISO 26262, and the legal aspects and the qualification of the product.

The role of the functional safety engineer relates to the technical work of deriving a technical safety concept from the functional safety concept, moderating an FMEDA and defining a set of diagnose functions to be part of the monitoring functionality, the hardware design (decisions about hardware redundancy), design of the HSI (Hardware/Software Interface), and the use of test methods to achieve a 100% test coverage of the fault injections and diagnose functionality. Both roles closely collaborate in a functional safety team according to the integrated engineering design approach that is characteristic for modern product development [9]. Also the industrial experience clearly outlined that functional safety is not a topic you can assign to one responsible person. A technical safety concept is usually created by a team of software, hardware and system level experts and moderated by a systems architect collaborating with the functional safety engineers. Also an FMEDA is usually done in a multidisciplinary team and the same applies for a hazard and risk analysis.

This means that the functional safety manager is a role which is played a few times in a company, while the role of a safety engineer (and the knowledge of it) can be assigned to even a whole team. For the SafEUr project this results in a concept where two roles are covered by one skills set, and depending on the roles, different skills elements and training units can be selected.

## 3 SafEUr Best Practices Approach – Learning by Real Examples

Another main feature of SafEUr is that we explain the very complex theory of the functional safety standard based on real case automotive examples. We use examples of ASIL-D classified items in gear box design, in ABS brake design, and steering wheel system design. In the courses and coaching we then ask the attendees to apply that on their systems ("learning by doing" approach) and discuss the result in the team. This results in a number of fruitful discussions and a real knowledge transfer.

The trial courses so far already yielded as broad spectrum of functional safety example items elaborated by course attendees, like active suspension, lighting system, power window, hydrogen tank, drive-by-wire, as well as battery management and "electronic differential" systems for an electric race car.

Below we would like to give some illustrations of the level of knowledge which is expected for functional safety managers and functional safety engineers. This helps to understand the depth of knowledge transferred for the units and elements outlined in Fig. 1. One of the most important steps at the beginning of the safety life cycle is the item definition and the hazard and risk analysis. This results in an ASIL classification of the hazard and the formulation of an overall safety goal.

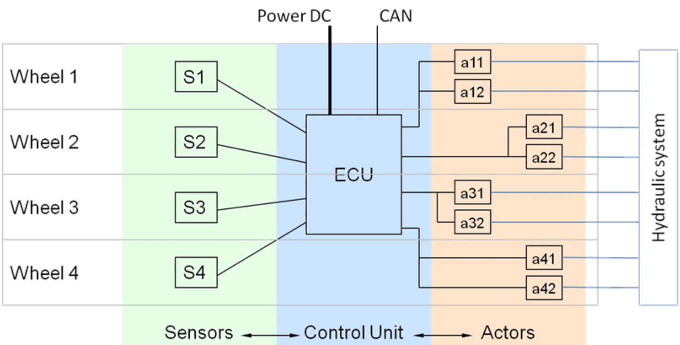



**Fig. 2.** SafEUr – Selected Example of an Item Definition (Example ABS Break System)

| | B) Hazard identification | | A) Situation (Worst Case) | C) Classification | | |
|---|---|---|---|---|---|---|
| Fkt. | Failure | Consequence of failure | | Severity | Exposure | Controllability |
| ABS | MIN | No brake force at the wheels | Fully occupied vehicle; higher speed; obstacle ahead (e.g. traffic jam)<br>Or:<br>Average speed; city traffic; pedestrians ahead (e.g. crosswalk) | S3<br>Without the possibility to slow down a crash is possible. Depending of the speed, passengers in the car as well as traffic participants could be badly injured or killed. | E4<br>Braking at medium or higher speed is an every-day situation. | C3<br>A car without operating braking system is for most drivers not controllable. The average driver and other traffic participants are normally not able to avoid harm in this situation. |
| | MAX | Wheels blocking / wheel slip | | | | |

**Fig. 3.** SafEUr – Selected Example of a Hazard and Risk Analysis (Example ABS Brake System)

| | B) Hazard | | A) Situation | C) Classification | | | D) Determination of ASIL and safety goals | |
|---|---|---|---|---|---|---|---|---|
| Fkt | Failure | Conseq. | | S | E | C | ASIL | Safety Goal |
| ABS | MIN | No brake force | ... | S3 | E4 | C3 | ASIL D | No hydraulic pressure (by fault) must be avoided |

**Fig. 4.** SafEUr – Selected Example of a Safety Goal (Example ABS Brake System)

| Severity class | Probability class | Controllability class | | |
|---|---|---|---|---|
| | | C1 | C2 | C3 |
| S1 | E1 | QM | QM | QM |
| | E2 | QM | QM | QM |
| | E3 | QM | QM | A |
| | E4 | QM | A | B |
| S2 | E1 | QM | QM | QM |
| | E2 | QM | QM | A |
| | E3 | QM | A | B |
| | E4 | A | B | C |
| S3 | E1 | QM | QM | A |
| | E2 | QM | A | B |
| | E3 | A | B | C |
| | E4 | B | C | D |

**Fig. 5.** ISO 26262 – Determination of an ASIL level

The Anti-lock braking system (ABS) allows the driver to maintain steering control in situations like heavy braking or on slippery surfaces by preventing significant wheel slip.

The system constantly monitors the rotational speed of each wheel. When it detects a wheel rotating significantly slower than the others (a condition indicative of impending wheel lock) it actuates the valves within the brake hydraulics to reduce hydraulic pressure to the brake at the affected wheel, thus reducing the braking force on that wheel. The wheel then turns faster; when the wheel is turning significantly faster than the others, brake hydraulic pressure is increased so the braking force is reapplied and the wheel slows. This process is repeated continuously, and can be detected by the driver via brake pedal pulsation. The Figs. 3 and 4 show an implementation example of the methods of ISO 26262 illustrated in the Figs. 5 and 6.

Classes of severity

| Class | S0 | S1 | S2 | S3 |
|---|---|---|---|---|
| Description | No injuries | Light and moderate injuries | Severe and life-threatening injuries (survival probable) | Life-threatening injuries (survival uncertain), fatal injuries |

Classes of probability of exposure regarding

| Class | E0 | E1 | E2 | E3 | E4 |
|---|---|---|---|---|---|
| Description | Incredible | Very low probability | Low probability | Medium probability | High probability |

Classes of controllability

| Class | C0 | C1 | C2 | C3 |
|---|---|---|---|---|
| Description | Controllable in general | Simply controllable | Normally controllable | Difficult to control or uncontrollable |

**Fig. 6.** ISO 26262 – Classification schema used in Figs. 3 and 4

The risk analysis results in a so called safety goal (see Fig. 4) and an ASIL-classification. In this example case the hazard is identified as ASIL D. The safety goal mentioned there is: “No hydraulic pressure by an electronic fault must be avoided!”. Such an analysis is now the starting point of work for the functional safety manager and the functional safety engineer. Then, the systems architect, a multidisciplinary team of experts from different departments and the functional safety engineer analyse the system and create an item definition (see Fig. 1) outlining the elements which can cause such a hazardous situation.

Especially the so-called safety critical signal flow is analysed (from sensors through the ECU to the actuators). Safety critical signals are described in the HSI (hardware software interface). In this case e.g. we must be able to trust the speed calculation of the car. So we would carefully analyse the speed sensors and realise that the speed can be calculated from 4 single inputs as an average speed. If one of the sensors fails we can still use 3 sensors and if two of the sensors fail we might not trust the average speed any more. Another issue is that we must trust that the rotational speed sensor on each wheel is measuring correctly, otherwise the ECU would make false decisions for the opening or closing of valves on that wheel.

At this stage the standard would impose a so called decomposition. An ASIL-D imposes the selection of a hardware at a very low FIT rate (1 FIT = the probability of a failure per hour is $10^{-9}$) and a high diagnostic coverage (a lot of diagnose and independent monitoring SW development). This can get very expensive and you might realise that a rotational speed sensor does not fulfil the ASIL-D error rates in general, and there is no realistic scenario to realise this reliability without extensive diagnosis. The standard proposes a redundancy in such cases. So you can (splitting ASIL-D in two ASIL-B) use two rotational speed sensors at ASIL-B quality (lower error rate demands) at each wheel. However, the standard also allows other types of redundancy. In the industry, for instance, a model simulation is used as a parallel model to monitor the speed vector at each wheel concluding that the rotational speed sensor is still working.

Based on such technical analysis the functional safety engineer (in collaboration with experts from different departments) creates a functional safety concept, including safe states, safety functions, diagnose functionality and monitoring, etc. Here are some examples:

- A default safe state in the ABS is the switching off of the ABS ECU and having the traditional brake system as a backup. However, a system can have many safe states. There could be e.g. a limp home mode when we can only trust 3 of the 4 wheel speed sensors and thus coordinate with the motor control unit a maximum rpm supported (resulting in e.g. a speed limit).
- A functional safety concept would define e.g. that a trusted speed for the overall vehicle must be calculated with an ASIL-D quality. Or it can demand that a mathematical model is used as a parallel monitoring to check whether the speed vector for a specific wheel speed sensor is behaving correctly, otherwise we would not trust the value any more.

While the functional safety engineer does his technical work the functional safety manager would establish safety plan and a major part in the safety plan is that the standard proposes specific methods to be applied in the design. Most of the methods proposed have a direct impact on the design implemented in the technical safety concept (see Fig. 7).

| Methods and measures for error detection at **software architecture level** | | A-SIL | | | | How is this implemented in the project |
|---|---|---|---|---|---|---|
| | | A | B | C | D | |
| 1a | Range checks of input and output data | ++ | ++ | ++ | ++ | eqivalence class testing for signals |
| 1b | Plausibility check | + | + | + | ++ | A safety critical signal is read by two different channels and the values are compared, if the signal cannot be trusted a safe state is actuated |
| 1d | Detection of data errors | + | + | + | + | All safety critical variables will be stored with a checksum, and because of ASIL D in the independent safety monitoring application the calculation is done with a bit complement to identify bit changes in operation |
| 1c | External monitoring facility | - | + | + | ++ | Due to ASIL-D a separate CPU with an external monitoring application is implemented. |
| 1e | Control flow monitoring | - | + | ++ | ++ | A counter is used to remember the start and end of a specific module which is called in a task and a monitoring system checks the counter sequence |
| 1f | Diverse software design | - | - | + | ++ | not used, we already calculate with the bit complement in the external monitoring unit. |

| Explanations |
|---|
| - .. Not recommended |
| + ... Recommended |
| ++ .. Highly Recommended |

The method selection is argued in the method selection as part of the safety plan

**Fig. 7.** ISO 26262 – Method Table Example 1

The method tables list methods per ASIL level which are highly recommended (++). The functional safety manager must go through each table and for each method depending on the ASIL level clarify its usage. If a specific method is not used, this must be explained. Beside the safety plan (as a Gantt plan) the analysis of all method tables forms an appendix to the safety plan.

This method table has an influence on the work of the functional safety engineer and what level of monitoring, plausibility checks, range checks etc. will be required.

In the SafEUr examples we also look deeply into the hardware and software design, including hardware FIT rate analysis and software architectural design (e.g. E-Gas model) and software diversity. One of the SafEUr goals in the engineering part is to point out the necessity of iterations while stepping down from the Hazard&Risk Analysis, via Functional and Technical Safety Concepts to Hardware and Software Design as well as to make aware of the mutual influence between hardware and software design and verification.

Beyond implementing safety in the corporate organisation and the engineering process, the SafEUr syllabus also deals with methods for taking into account the reliability of the mechanical systems and subsystems, as well as the required safety control mechanisms in production and maintenance. Often not addressed in comparable trainings, both these subjects are vital to implementing safety on a complete system level, as major architectural design decisions depend on component availabilities, and safety-critical subsystems have to be treated in special ways in production and End-of-Line tests.

## 4 SafEUr – European Vision

The SafEUr results are on trial by major Automotive Tier 1 companies at the moment. The results will lead to a Release 3 development in autumn 2013. Interested companies can join an online training in summer 2013: certified trainers will guide and support trainees via an e-learning platform where all the training material will be available, and exercises be submitted and discussed on-line in groups.

In 2013 at EuroSPI 2013, the safety community will be extended to include also major medical device industry and the safety design strategies in this industry domain. This way the experience exchange between Automotive and Medical device industry might lead to further best practices to implement functional safety standards.

SafEUr can also lead to innovation. The keynote at EuroSPI 2012 (KTM Motorsport) applied the functional safety concepts on their new bikes and invented a lot of new functionality by that approach. E.g. when analysing the light control system and the safety state ("there is always light") the system and software design was adapted so that if a lamp is faulty it automatically switches to a different light (high beam, low beam, parking light, day light). The probability that the lamp of all 4 types of light fails is then very low fulfilling the ASIL requirement. However, all this has an impact on product design because it must be possible to diagnose each light separately, to switch the light on by a separate bridge on the ECU (otherwise a failure can switch off all 4 at once), etc.

We also will continue in the safety working party of German companies (www.soqrates.de) and further elaborate best practices and share knowledge about what is the right way of implementation. Key stakeholders of the French automotive industry will also be involved.

SafEUr follows the ECQA [3] (European Certification and Qualification association) standards. This means that from autumn 2013, all the exams across Europe will be standardised, an international job role committee will be formed to maintain the defined skills set and knowledge structure. The ECQA will then act as the certification body across Europe and other continents.

## 5 Conclusion

SafEUr is a real and unique industry-driven European initiative establishing a certified practice-oriented training program in Functional Safety certified by a European Certification Organisation. SafEUr therefore fills a large gap that still exists between a rapidly growing number and variety of functional safety standards, and their efficient and effective implementation in modern products and systems, as well as the enabling of engineering organisations and projects. Its initial focus is clearly on automotive applications, and thus on ISO 26262, however, its scope will be widened up with its increasing deployment in different industry sectors. Currently, EU-funded on-site and on-line pilot trainings both in industry and academia help assure the high level of quality and relevance of the complete program.

**Acknowledgements.** The SafEUr project is financially supported by the European Commission in the Leonardo da Vinci part of the Lifelong Learning Programme under the project number 518632-LLP-1-2011-1-AT-LEONARDO-LMP. This publication reflects the views only of the authors, and the European Commission cannot be held responsible for any use which may be made of the information contained therein. We are grateful to the experts who have contributed to the SOQRATES Design AK and Safety AK: O. Bachmann (SIBAC), S. Habel, L. Ross, I. Sokic, R. Dreves (Continental Automotive), F. König, A. Koundoussi, H. Galle (ZF), A. Much (Elektrobit), H. Zauchner, A. Kaufmann (KTM Motorsport), L. Borgmann (HELLA), G. Spork (Magna Powertrain), M. Haimerl (IMBUS), K. Dussa-Zieger (Methodpark), A. Riel (EMIRAcle/Grenoble INP), J. Unterreitmayer (SQS), C. Kreiner (TU Graz), D. Ekert, R. Messnarz (ISCN).

## References

1. International Organization for Standardization (ISO): ISO 26262. Road vehicles – Functional safety – Parts 1–9 (2011)
2. Riel, A., Bachmann, V.O., Dussa-Zieger, K., Kreiner, C., Messnarz, R., Nevalainen, R., Sechser, B., Tichkiewitch, S.: EU project safEUr – competence requirements for functional safety managers. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 253–265. Springer, Heidelberg (2012)

3. Messnarz, R., Sicilia, M.A., Reiner, M.: Europe wide Industry Certification Using Standard Procedures based on ISO 17024. In: Proceedings of the TAEE Conference in Vigo Spain. IEEE (June 2012)
4. Messnarz, R., König, F., Bachmann, V.O.: Experiences with Trial Assessments Combining Automotive SPICE and Functional Safety Standards. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 266–275. Springer, Heidelberg (2012)
5. SOQRATES Safety Team, Messnarz, R., Ross, H-L., Habel, S., König, F., Koundoussi, A., Unterrreitmayer, J., Ekert, D.: Integrated Automotive SPICE and safety assessments. Wiley SPIP 14(5), 279–288 (2009)
6. Bachmann, O., Messner, B., Messnarz, R.: Adapting the FMEA for Safety Critical Design Processes. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 290–297. Springer, Heidelberg (2011)
7. SOQRATES Initiative, `http://www.soqrates.de` (last accessed on April 07, 2013)
8. Automotive SPICE, an international standard used in Automotive industry, `http://www.automotive-spice.com` (last accessed on April 07, 2013)
9. Riel, A., Tichkiewitch, S., Messnarz, R.: Qualification and Certification for the Competitive Edge in Integrated Design. CIRP Journal of Manufacturing Science and Technology 2(4), 279–289 (2010)

# Automotive Knowledge Alliance AQUA – Integrating Automotive SPICE, Six Sigma, and Functional Safety

Christian Kreiner[1], Richard Messnarz[2], Andreas Riel[3], Damjan Ekert[2], Michael Langgner[4], Dick Theisens[5], and Michael Reiner[6]

[1] Graz University of Technology, Austria
christian.kreiner@tugraz.at
[2] ISCN LTD/GesmbH, Ireland and Austria
rmess@iscn.com
[3] EMIRACLE Association, Belgium
[4] Automotive Cluster Austria, Austria
[5] Symbol BV, Netherlands
[6] European Certification and Qualification Association, Europe

**Abstract.** This paper discusses (based on the EU project AQUA) how the core elements of three complementary approaches and standards can be integrated into one compact skill set with training and best practices to be applied. In this project experts from Automotive SPICE (ISO 15504), Functional Safety (ISO 26262) and Lean Six Sigma collaborate. In a first analysis the experts identified an architecture of core elements where all three approaches fit together and where a holistic view about improvement is needed. The Automotive Clusters from Austria and Slovenia are trial partners and will roll out this knowledge in pilot courses to the industry. Other Automotive Clusters showed interest and will join the trial phase.

**Keywords:** Automotive SPICE, Functional Safety, Lean Six Sigma, Integrated View.

## 1 Introduction

Electronics and software control 70% of modern cars' functionality; studies predict 90% and more tomorrow. The induced system complexity makes it increasingly difficult for automotive companies to master interdisciplinary, horizontal issues such as quality, reliability, and functional safety.

Moreover, the ISO 26262 reference standard for road vehicles has been released only very recently. Consequently, existing knowledge is rare, and highly specialised on teaching the standard rather than its practical implementation. This is where competition is happening in automotive worldwide, and where Europe can create a competitive advantage.

In the Automotive Cluster Austria they currently discuss "Can we still manage the complexity of software and electronics in cars?" [1], and come to the conclusion that such integrated automotive and safety engineering best practices are needed.

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 333–344, 2013.

Key Notes about Functional Safety at EuroSPI 2012 illustrated that functional safety is increasingly important for the success on the market:

The EuroSPI 2012 key note from the KTM quality head stated: “It is important to show a way of effective integration of the process and the methods of functional safety for a medium-sized business based on pilot projects. The principle of these projects is to acquire expert knowledge via practical execution of the work products and simultaneous training.”

The EuroSPI 2012 key note from a Magna program manager of a highly safety critical product line states that “for Tier 1 suppliers of mechatronic systems it is inevitable today to comply with standards like Automotive SPICE and ISO 26262 (Functional Safety for Road Vehicles). This can lead to substantial on-top costs and a lot of additional effort if especially requirements management is not implemented in a smart way."

A group of industry partners from Automotive and medical device industry joined a workshop series at EuroSPI and collaborate in task forces since 2003 which was kick-off financed for one year by the Bavarian software initiative. This group published a number of papers about their integration of Automotive SPICE and Functional Safety in an integrated approach [2], [3], [4].

The Lean Six Sigma Academy published papers about EuroSPI emphasising the implementation of Lean Six Sigma in Europe applying the Toyota success story in Europe. They presented different levels of six sigma experts (yellow, orange, green, black belt) and contributed examples of success stories.

Automotive SPICE, Functional Safety standards, and Lean Six Sigma in a way form the quality backbone of the automotive industry. Only such common standards enable highly integrated supply chains as we find them in automotive industry. For a participating company this means competence and ability in all these areas is a priority.

Also the Automotive Clusters reported that - while there is a limited number of Tier 1 companies in the market - there are hundreds of Tier 2 and Tier 3 small and medium sized companies. They do not have the time to invest in each of the three approaches separately and they need an integrated compact view which can be implemented (as much as possible related to real practice).

The EU project AQUA proposes such a compact integration of core elements and will create training which is delivered through the Automotive Clusters.

## 2 Modular Integration Strategy

We illustrate the AQUA architecture in Fig.1 below. In AQUA a base layer of core modules will be established which allow an integrated and complementary view about the three approaches, including Automotive SPICE, Functional Safety, and Lean Six Sigma. Integrated means, the base layer modules extract and teach common paradigms and principles - “the essence” - from the latter, and the link layer expresses the mapping or translation to Automotive SPICE, Functional Safety, and Lean Six

Sigma. In this way, the complexity of learning and mastering all three standards can be significantly reduced.

From the core modules AQUA also develops a linking layer which references the parts addressed in the existing and established approaches, such as Automotive SPICE, Functional Safety, and Lean Six Sigma.

Also the AQUA project maps the key words which the partners in the Automotive clusters need to address onto the core modules which we propose in AQUA.

The Automotive Clusters stated that these core modules must contain enough best practice examples so that based on the core modules an implementation in the projects can be done. A selected set of experts in the company will be recommended to the full courses (see link to existing and established approaches and courses (= layer of existing courses on the market).

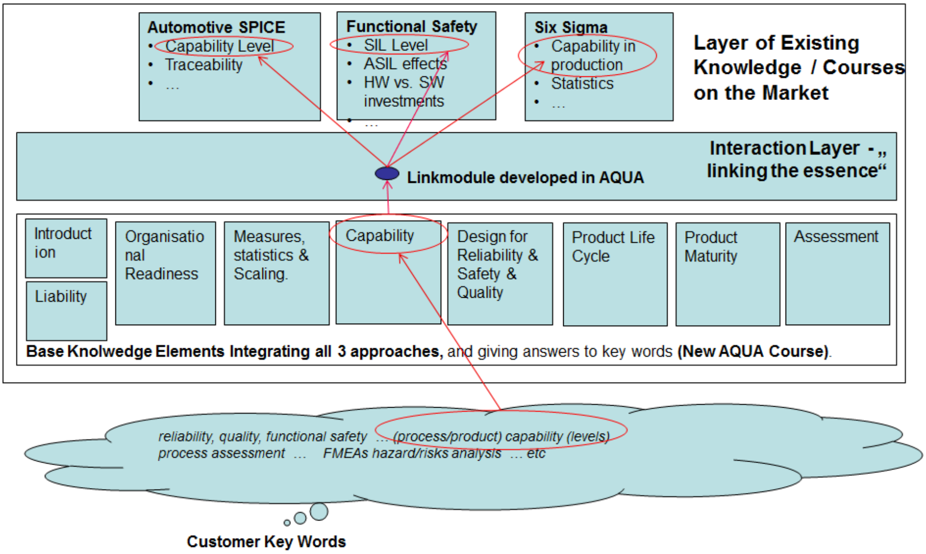



**Fig. 1.** Integrated Base Modules Concept of AQUA and Linking Strategy

The functionality of this architecture in Fig.1 can be illustrated based on a key word based “signal” flow (see the arrows in Fig.1).

If you take the key word “capability” it has three different meanings although it is used by all three approaches. Thus in a core module the concept of capability is explained from the three perspectives. In Automotive SPICE [5] the capability levels are derived from process capability levels based on ISO 15504 (the capability of an engineering process such as ENG.5 Software Design). In ISO 26262 the Safety Integrity Levels ASIL-A to ASIL-D are originally derived from IEC 61508 and represent a specific redundant hardware design and hardware FIT rate (1 FIT is equal to a probability of 10-9 that an error occurs in an hour) and corresponding diagnostic coverage by software to avoid that failures of the electronic lead to hazardous situations for the driver. So this SIL is a kind of product maturity level. And in Six

Sigma the capability relates to the production capability which means that by statistical quality control the number of faults in introduction is reduced to achieve 6-sigma deviation. This is also needed in Automotive because the contracts in Automotive mention ppms (Parts per Million Errors) which need to be achieved and contracts contain less than 100 ppm.

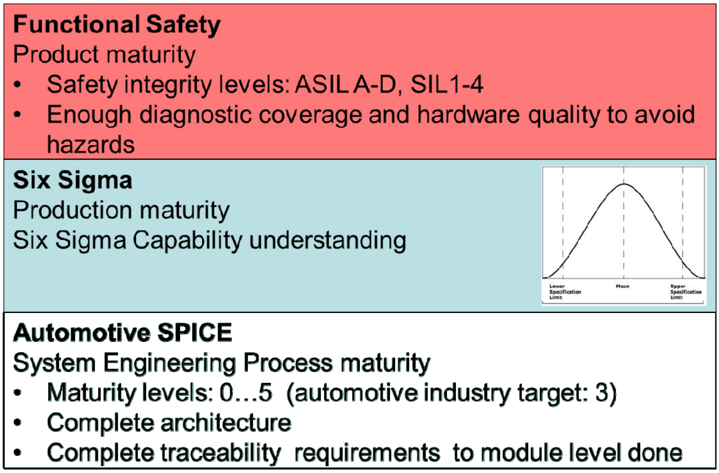



**Fig. 2.** Example Skill Element – AQUA Module – U2.E2 – Capability

As outlined in Fig.2, AQUA will create a three dimensional view on key terms, broadening the mind set of Automotive companies to understand and implement capability from three angles: process capability, product capability, as well as production capability. It is obvious that it would be best to cover all three aspects at once in a holistic quality and engineering approach.

## 3 Overview of the Proposed AQUA Core Modules

The following module architecture for AQUA (see the base layer in the architecture in Fig.1 above) has been elaborated in a kick off workshop with experts from Automotive SPICE, Six Sigma, and functional safety.

Each module fulfills the following criteria:

- The module addresses key words which the Automotive Clusters addressed as a requirement for the members
- The module integrates the view of the three approaches in one holistic concept for process capability, product capability, and production capability
- The module can be linked to specific content in the established certification and course programs of Automotive SPICE, Functional Safety, and Lean Six Sigma

Also the modules are mapped onto the skills set definition standards of ECQA [7] (European Certification and Qualification Association). ECQA established Europe wide standards for skills definitions, skills assessment, online teaching and Europe wide exams following standard procedures. In ECQA the competencies are structured in so called skills sets (see Fig. 3 and Fig.4)

A **DOMAIN** contains
**JOB ROLES**, which contain
**UNITS**, which contain
**ELEMENTS**, which contain
**PERFORMANCE CRITERIA**, which must be proven by
**EVIDENCES**.

**Fig. 3.** Standard Skills Set Architecture (ECQA)

- Introduction
  - U1.E1: Overview of Standards and Norms
  - *U1.E2: Liability*
  - U1.E3: Organisational readiness (&continuous improvement)
- Measurement
  - U2.E1 Measures/Statistics/Scaling
  - U2.E2 Capability in 3 dimensions (metrics)
- Engineering
  - U3.E1: Design for Reliability & Safety & Quality (Structure)
  - U3.E2: Design for Reliability & Safety & Quality (Tools)
- Product lifecycle
  - U4.E1: Life cycle (V-model (VDA 6.1), traceabilty, methods, sfty LC, APQP)
  - U4.E2: Product maturity (traceabilty, DC+V&V methods, APQP DVP&P (18)
- U5.E1: Assessment

**Fig. 4.** AQUA Skills Set Architecture (ECQA compliant)

Each AQUA module has been assigned to a specific unit and skills element based on an overall skill set architecture. For each module three complementary views of knowledge are considered, looking at the topic based on Automotive SPICE, functional safety, and Six Sigma.

Here is a list of the core modules and the main content topics.

All three approaches help to avoid costs related to liabilities / penalties (see Fig.7). While functional safety addresses liabilities due to injuries and casualties due to hazardous errors of the electronics, the Six Sigma helps to avoid penalties due to not reaching specific ppm rates or start of production milestones. Automotive SPICE assures that you can demonstrate capability needed to get the contracts with customers. Failing in one of these three perspectives can cause a lot of costs for the company.

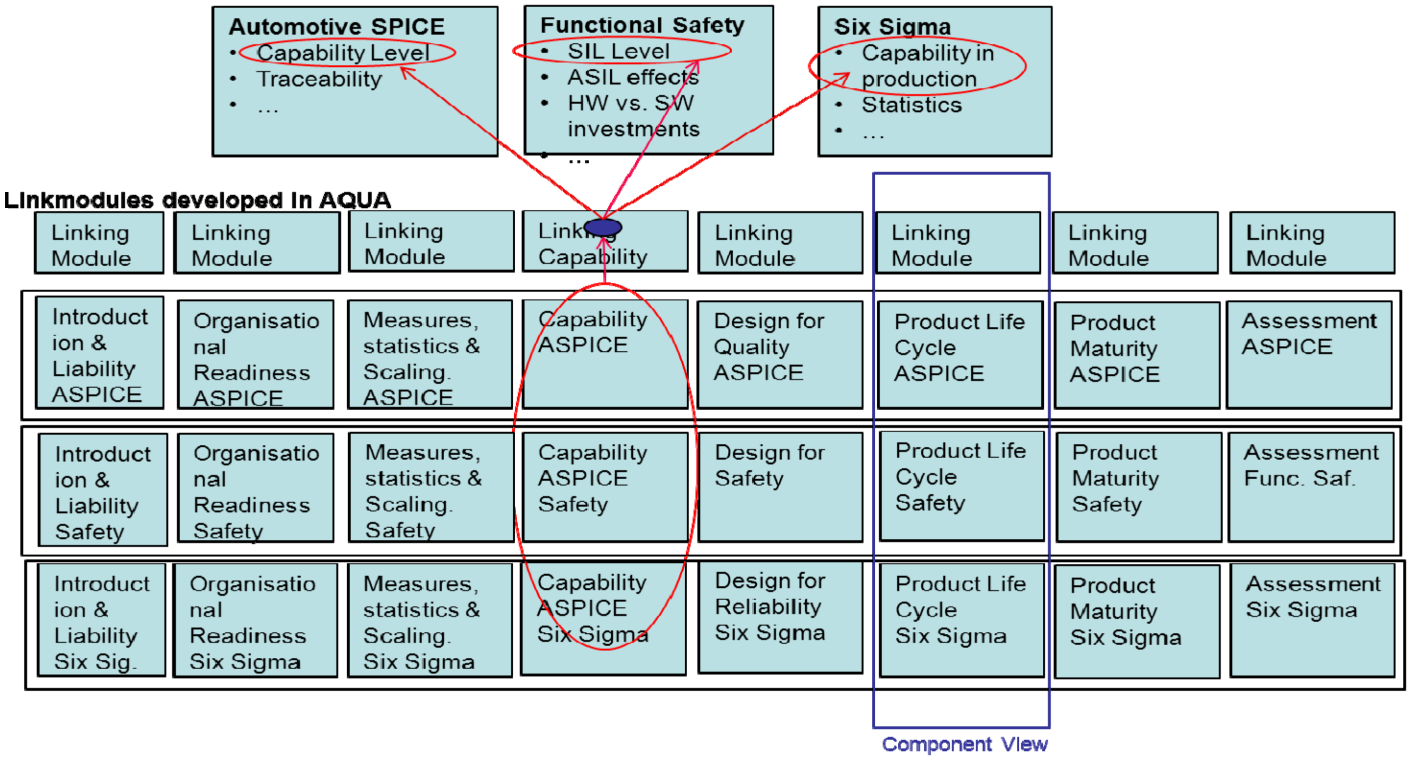



**Fig. 5.** AQUA course structure: each component (an ECQA element) consists of core modules from ASPICE, Safety, Six Sigma, plus a linking/consolidation module

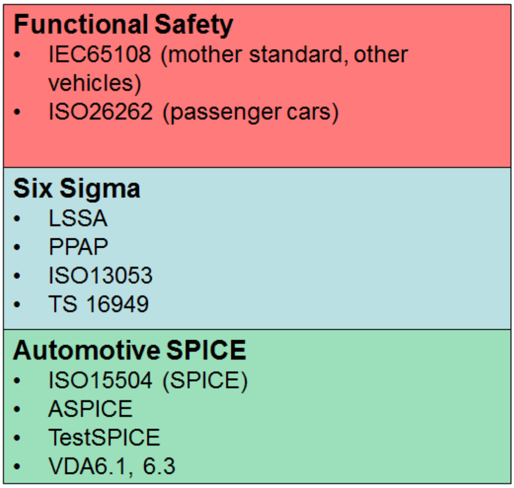



**Fig. 6.** Integrated View – U1.E1 Introduction/ Overview of Standards

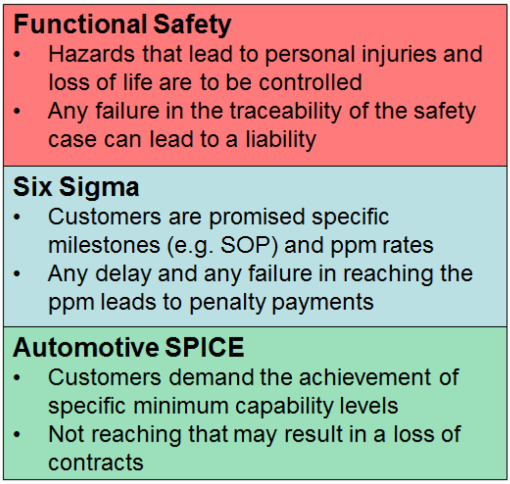



**Fig. 7.** Integrated View – U1.E2 Introduction/ Liability

Each of the three approaches (see Fig. 8) has a specific understanding of a required organizational structure. In Six Sigma improvement projects are managed for improving the production capability helping to avoid cost of errors, A Green Belt manages improvement projects, a Black Belt managers the team of Green belts and organizes larger scale improvement projects. Yellow Belts are staff which understand statistical basics and apply the measures in the production (help to gather the data). In

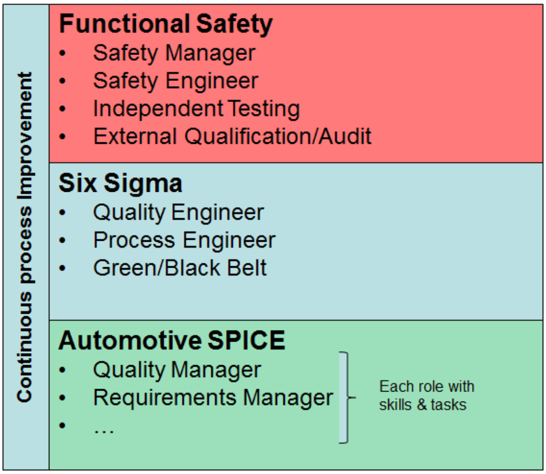



**Fig. 8.** Integrated View – U1.E3 Introduction/ Organisational Readiness

functional safety management there are safety managers who are responsible for the safety plan, the safety life cycle, and managing the safety case (starting from a hazard and risk analysis). And the functional safety engineers are experienced architects who are responsible for the functional safety requirements, the technical safety concept, etc. In Automotive SPICE there are different management, supporting and engineering processes and each process requires specific roles (e.g. a requirements manager, a configuration manager, a software tester, etc.).

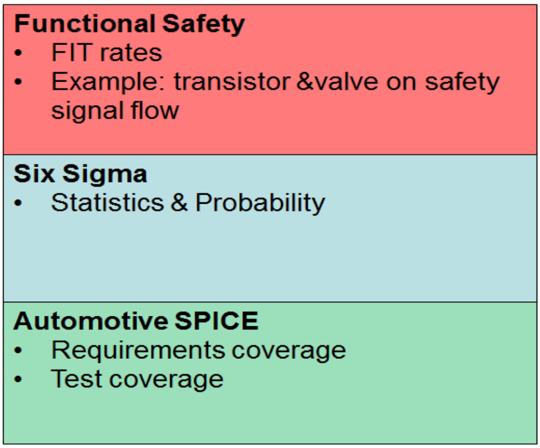



**Fig. 9.** Integrated View – U2.E1 Measures / Statistics / Scaling

All three approaches demand the use of statistical measures (see Fig.9), but all three use metrics on a different level. In Automotive SPICE the metrics are used to prove that the engineering is complete, e.g. coverage of all requirements in the specification, coverage of all requirements in the test, coverage of all requirements in the design, etc. In functional safety the hardware metrics are used to evaluate the probability per hour that the hardware fails (FIT=$10^{-9}$ probability that the hardware fails per hour) and the required coverage of hazardous error by software diagnostics (in percent). And in Six Sigma the probability theory and statistical process control are used to predict the number of errors in production.

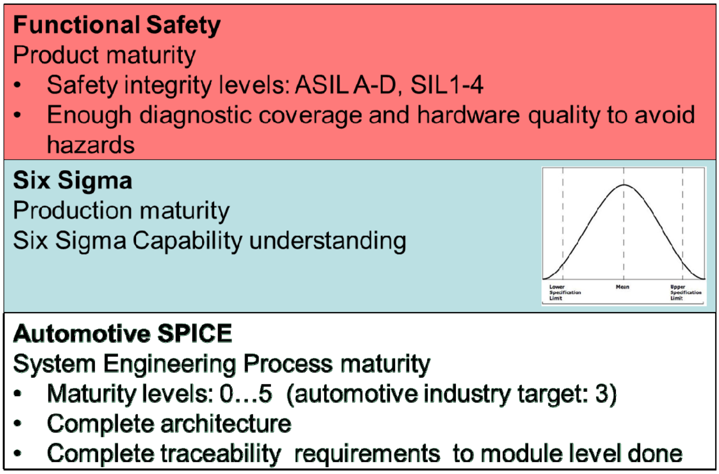



**Fig. 10.** Integrated View – U2.E2 Capability in 3 dimensions

All three approaches (see Fig. 11) demand a specific set of processes and methods to be implemented. In functional safety the safety critical path (from sensors to electronics / software to actuators) is analyses in an item definition and depending on the safety integrity level the redundancy of hardware and diagnostic capability of software is derived. Tools used are H&R (hazard & risk analysis), FMEDA (Failure Modes, Effects and Diagnostic Coverage Analysis), Signal Flow Design, Technical Safety Concept, HSI (Hardware Software Interface), etc. In Automotive SPICE assessment tools are used to determine capability levels and derive improvement actions. The Automotive SPICE model emphasizes the use of tools to establish traceability throughout the engineering (e.g. traceability of requirements to design). In Six Sigma tools are used to preventively identify weaknesses (e.g. DOE – Design of Experiments, e.g. DFMEA – Design FMMEA), and a cluster of statistical tools to control the production, as well as tools to analyse the production flow / optimization (CTQ flow).

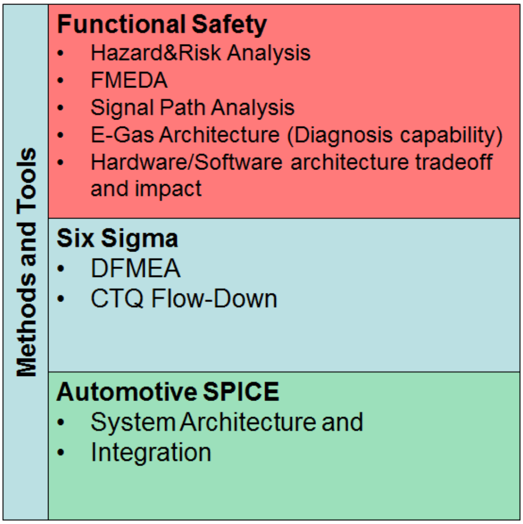



**Fig. 11.** Integrated View – U3.E1 Design for Reliability & Safety & Quality ("Design for 0 Errors") & U3.E2 Tools

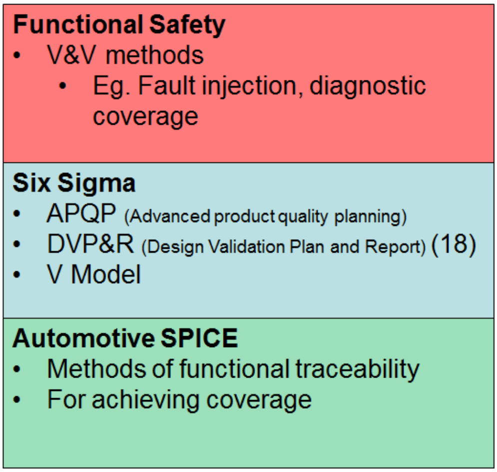



**Fig. 12.** Integrated View – U4.E1 Product Lifecycle

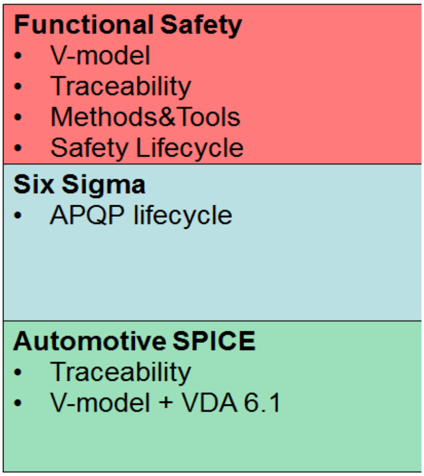



**Fig. 13.** Integrated View – U4.E2 Product Maturity

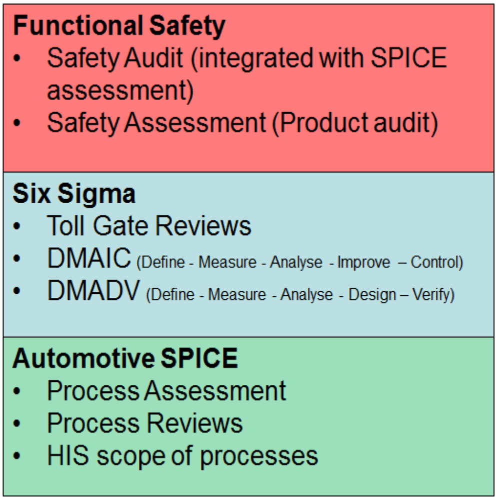



**Fig. 14.** Integrated View – U5.E1 Assessment

## 4 Conclusion and Outlook

The Automotive Knowledge Alliance (AQUA) was formed to bring together Automotive SPICE (ISO 15504), Functional Safety (ISO 26262) and Lean Six Sigma under one skills framework. The architecture of this modular framework is symmetric, that means, skills, methods, and practices from all three areas are not only offered in isolated skill sets and courses, but also in a synoptic way organized and linked by common concepts across all three areas. From that we expect better integrated understanding and practicing Automotive SPICE, Functional Safety, and

Lean Six Sigma within one organization, as well as more effective learning by offering a new paradigm of course structuring.

Furthermore a Europe-wide recognized certification scheme will be offered via the European Certification and Qualification Association (ECQA).

The modular AQUA architecture will be reviewed by industrial partners and based on this feedback the structure will be refined. The development of knowledge modules as course modules starts in the mid of 2013 and by end of 2013 a first version of elaborated materials will be available for interested partners of Automotive Clusters across Europe.

**Acknowledgement.** The AQUA project is financially supported by the European Commission in the Leonardo da Vinci part of the Lifelong Learning Programme under the project number EAC-2012-0635.

This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

## References

[1] Automotive Cluster Austria. AC Quarterly Magazine (February 2012)

[2] Riel, A., Bachmann, V.O., Dussa-Zieger, K., Kreiner, C., Messnarz, R., Nevalainen, R., Sechser, B., Tichkiewitch, S.: EU Project SafEUr – Competence Requirements for Functional Safety Managers. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 253–265. Springer, Heidelberg (2012)

[3] Messnarz, R., König, F., Bachmann, V.O.: Experiences with Trial Assessments Combining Automotive SPICE and Functional Safety Standards. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 266–275. Springer, Heidelberg (2012)

[4] SOQRATES Safety Team, Messnarz, R., Ross, H.-L., Habel, S., König, F., Koundoussi, A., Unterrreitmayer, J., Ekert, D.: Integrated Automotive SPICE and safety assessments. Wiley SPIP 14(5), 279–288 (2009)

[5] Automotive SPICE, an international standard used in Automotive industry, `http://www.automotive-spice.com`

[6] Theisens, D.: How Green is your Black Belt. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 257–267. Springer, Heidelberg (2010)

[7] Messnarz, R., Sicilia, M.A., Reiner, M.: Europe wide Industry Certification Using Standard Procedures based on ISO 17024. In: Proceedings of the TAEE Conference in Vigo Spain. IEEE (June 2012)

[8] ISO 26262, Road vehicles — Functional safety

[9] SOQRATES Initiative, `http://www.soqrates.de`

[10] HIS, `http://www.his-automotive.de`

# Experience with an Integrated Risk Management Process in the Medical Regulatory Environment

Botond Tényi, Adrien Csík, Ibolya Monoki, and Ferenc Tegzes

B. Braun Medical Kft., Budapest, Hungary
B. Braun Avitum AG, Active Medical Device R&D, Location Budapest
{botond.tenyi,adrien.csik,ibolya.monoki,
ferenc.tegzes}@bbraun.com

**Abstract.** In the medical domain, manufacturers are required to implement a Risk Management Process by multiple standards. ISO 14971 provides a framework and taxonomy for *medical device* risk management process but does not provide details or explanations of its requirements. The IEC 60601 standard family for medical electrical equipment defines the major hazards as an input for the Risk Management Process. Further process standards deal with software and use-related risks in particular (IEC 62304, ISO 62366). It is a challenging task to fulfil all these requirements in one integrated process and provide a comprehensive documentation (Risk Management File) to achieve compliance. We have set up a Risk Management (RM) process for our two different active medical devices. We will share our implementation which handles the Hardware, Software and Use Risk in an integrated way on the functional level. Further particular software-related RM tasks are handled in the software development life-cycle separately. Additionally, we will present our tool chain, which provides evidence of carrying out RM tasks throughout the whole development life-cycle, including connections to the requirement management and effectiveness verification.

**Keywords:** risk management, medical software development, usability engineering.

## 1 Introduction

Ensuring a high level of safety is a key goal in the medical domain, similarly to other safety critical domains. Most of the manufacturers have a long term product experience regarding safety issues. On the other hand, introduction of new technologies, medical therapies or new companies on this field require a systematic method to analyse the possible hazards in connection with the medical application. To ensure an equal safety level within the whole domain, the standard committees of different disciplines have setup requirements and continuously improve them by harmonising to each other and to the actual technological level. Despite such harmonisation effort,

F. McCaffery, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2013, CCIS 364, pp. 345–353, 2013.

these standards are separate ones and they have different focus, e.g. for Programmable Electrical Medical Systems (PEMS):

- Risk management (RM) process
- Development process (system – PEMS)
- Software development process
- Usability engineering (UE) process
- Identifying general hazards
- Identifying product group related particular functional hazards.

There is however no regulation, which integrates all these requirements and gives a standard method for their implementation. For software RM processes a Technical Report (IEC/TR 80002) is provided, detailing the recommended methods but still not in an integrated fashion. So the task remains for the manufacturers to setup their own integrated RM process, which defines the interfaces among the processes and provides a comprehensive documentation (Risk Management File) to achieve compliance with each relevant safety-related standard.

This paper presents an implementation of the Integrated Risk Management Process in our company. It highlights the phases where the different processes are connected, but it does not cover the complete RM process required by ISO 14971 (evaluation of overall residual risk acceptability, risk management report, production and post production information phases are not detailed).

## 2 Regulatory Compliance

### 2.1 Process Standards

ISO 14971 provides a framework and taxonomy for medical device RM process. It provides a questionnaire supporting the identification of qualitative and quantitative characteristics of the product and the relevant hazards, i.e. the potential sources of possible harms. The manufacturer shall create a Risk Management File (RMF) containing records required by this standard. However it does not provide details or explanations of its requirements.

IEC 60601-1 is the general standard of basic safety and essential performance. It takes special considerations in connection with PEMS devices defining the development life-cycle process in relation to the RM process. It gives an example structure for the RMF, as well.

IEC 62304 provides a framework of software life cycle processes necessary for the safe design and maintenance of medical device software. It requires the software RM process to be compliant with ISO 14971, with special attention to the software life cycle processes, where the level of detail depends on the safety classification of a software item. The prerequisite of these activities is a solid architecture design. This standard defines the required additional documentation into the RMF. However this standard does not provide detailed software RM methods and does not explain completely how software RM is applied within the (overall) medical device risk management.

Recommendations for the software engineering techniques and methods are described in the IEC/TR 80002 (Guidance on the application of ISO 14971 to medical device software). Examples and explanations can be found for the software RM activities through the whole software development life-cycle.

IEC 62366 standard deals with usability engineering process for all medical devices and use-related risks in particular. This standard gives the connection points between the RM process and the Usability Engineering (UE) process.

- UE process gives input for the following RM process steps:

- Identification of characteristics related to safety (users, use environment, user activities),
- Analysis of use-related hazards and hazardous situations,
- Identification of design changes that might result in new hazards or hazardous situations,
- Evaluation of residual risk (because usability goals have not been met).

- RM process gives input to UE process:

- At identification of Primary Operating Functions related to safety.

UE-related documentation shall be stored in the Usability Engineering File, which can be part of the Risk Management File.

### 2.2 Product Standards

IEC 60601-1 is the general standard of basic safety and essential performance. It defines the basic hazards and their hazardous situations as an input to the RM process, and contains 153 requirements with direct RM File links.

The particular standards (IEC 60601-2-x) of specific active medical devices define additional and more detailed functional safety requirements and relevant essential performance data. In some cases they override the general (IEC 60601-1) and the corresponding collateral (IEC 60601-1-x) standards.

## 3 Integrated Risk Management Process

The practice of the integrated Risk Management (RM) process was established through several pilot projects. The risk acceptance criteria reflecting the safety policy of the company are declared and illustrated in a risk matrix form, with severity and probability dimensions. The *severity level* definitions are qualitative (“fuzzy”), whereas the *probability of occurrence* definitions are quantitative. However for the easier and more reproducible result, practical examples based on statistical data are assigned to each level, to be used as a rule of thumb (e.g. probability of an electronic failure, or failure of an independent protective system with self-test).

Another important input for the RM process is the so called Hazardous Situation List. This list is created for each medical device group based on the relevant hazardous situations defined in the IEC 60601 standard family (general, collateral and particular standards) with the help of medical experts. Among other details, it contains the following information:

- Hazardous situations with unique IDs,
- Possible harms caused by the hazardous situation,
- Severity of the harm with reference to relevant literature,
- Probability of occurrence of the harm with the defined severity level.

A well-established Hazardous Situation List based on clinical data and previous experience with the medical device group is the fundamental basis of a reproducible risk assessment. It is worth to highlight the definition of the probability of occurrence of the harm (see P2 in Figure 1), which represents the medical probability reaching the harm with the specified severity in case of the given hazardous situation. Separating the medical and technical probabilities makes risk estimation easier for the risk management team with engineering background.

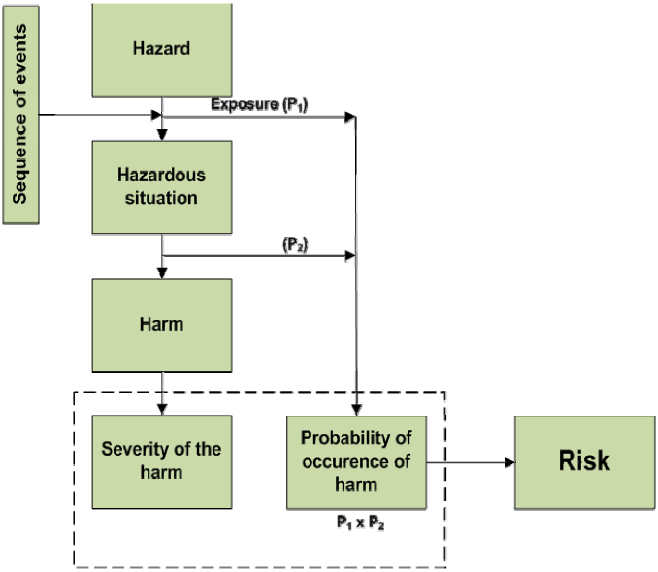



**Fig. 1.** Risk calculation

### 3.1 Risk Management Plan

The first step of the project specific RM process is the preparation of a Risk Management Plan. Besides defining the standard RM activities it describes the SW RM and usability engineering aspects or it refers to the relevant process plans.

### 3.2 Risk Assessment Process

The software and use RM cannot and should not be performed in isolation from the overall medical device RM. So the process handles the system hardware, software and use-related risks in an integrated way, but well separating them to be able to clearly prove the compliance with the separate standards. Furthermore RM activities cannot be effectively performed unless they are integrated into the product development and software development life-cycles and into the usability engineering process. In our process, the scope of the Risk Assessment (RA) is extended (compared to the definition in ISO 14971). It covers not only risk analysis and risk evaluation, but also risk control (see Figure 2). The reason is that the results of these phases are documented in the same document, but there is no generic term for that in the standard.

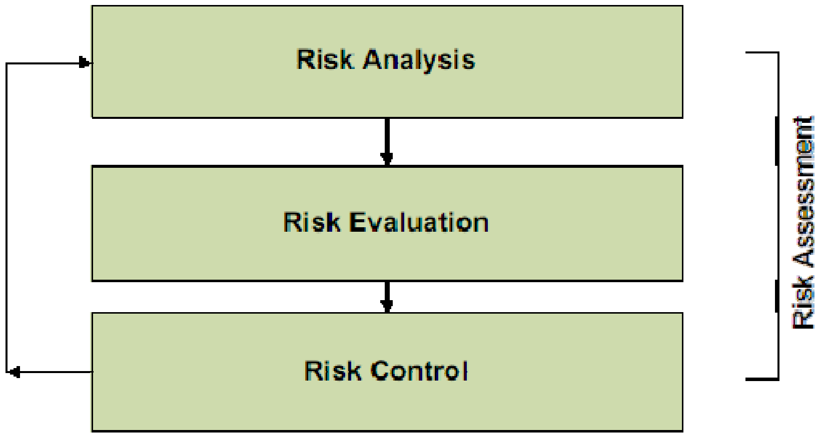



**Fig. 2.** Risk Assessment steps

**Create the functions list:**
The system hardware and software-related risks are analysed through the major functionalities. This is the same organizational structure as in the system requirements specification, providing a natural link to the highest level of the requirement management system. This is the common point of the system development life cycle and RM process.

The use-related risks are analysed through the user activities collected in a usage process specification, which provides the connection between the usability engineering and RM processes.

**Identify the major components related to the functions based on the system architecture.**

**Identify the foreseeable sequence of events related to the components and the caused hazardous situations (Risk Analysis):**

The final standard way of documentation is top-down, i.e. all related foreseeable sequence of events are linked to a hazardous situation. However, for system hardware and software risk analysis the most frequently applied method is the bottom-up FMECA (Failure Mode, Effect and Criticality Analysis). It means that at first all the possible normal operational and failure events or sequences are analysed and then the consequences, i.e. hazardous situations, are identified and arranged in the proper order. The traceability to the Hazardous Situation List is ensured by unique hazardous situation IDs.

Software failures are analysed top-down on a system level assuming the complete failure of a function caused by the software.

For the use-related risk analysis the SWIFT (Structured What-IF checklist Technique) method is used, i.e. what is the consequence in case of a use error during an activity.

**Evaluate the risk related to the foreseeable sequence of event (Risk Estimation and Evaluation). The risk is estimated and evaluated as follows (see Figure 1):**

- Severity: according to the Hazardous Situation List definition,
- Probability = P1 x P2:
  - P1: according to the risk matrix probability definition or examples in the Risk Management Plan.
  - P2: according to the Hazardous Situation List definition,
- Decision: according to the risk matrix.

In case of software risks the P1 probability is set to 1 as a standard definition.

**Define risk control measures (RCM) (Risk Control):**
The proper RCMs are defined according to the Option analysis requirement in ISO 14971. Each RCM is identified according to the selected option and SW measures are also specifically identified.

**Evaluate residual risk after implementation of RCM (Residual Risk Evaluation):**
Our standard approach is that the RCMs are reducing the probability and not the severity. However the determination of the probability of the occurrence is always a challenging task.

In case of protective measures, practical examples for the probability levels can be defined depending on the standard hardware safety schemes (e.g. a double channel system with an independent hardware monitoring subsystem) based on general statistical data.

If the protective measure is implemented fully or partially in software, statistical data cannot be used for the probability estimation, so another method is required. In this case practical examples for the probability levels can be defined depending on the safety classification and the related SW life-cycle model of the software item.

If the RCM is only an information for safety, i.e. the safety depends on the user, the practical examples for the probability levels are defined depending on the way the user is informed (e.g. via device display, Instructions for Use).

**Implement RCMs:**
The risk control measures (RCM) are entered into the requirements management system. Traceability is ensured via the RCM identifiers. This is an important connection point to the system (PEMS) development process ensuring the proper implementation of the RCMs taking over the "responsibility" from the RM process.

During the requirement break-down (from system to components and further to SW units) and the design activities, further interconnections are necessary between the system development and RM process, such as:

- Possible newer hazardous situations shall be analyzed, and
- Design shall also be reviewed (e.g. with an FMEA) not to degrade the protection level of the RCMs.

Further software-related interconnections will be described in chapter 3.3.

Use-related functions with residual risk after implementation of RCM (subset of Primary Operating Functions) are entered into the usability engineering process.

**Verify implementation of RCMs:**
Since RCMs become part of the requirement management system, their verification is ensured by the normal development process verifications on component and system levels. This way no specific verification activity is required within the RM process.

**Verify effectiveness of RCMs:**
Such verification activity is RM process specific. RCM implementation verification results are reviewed, whether they provide proper evidences for stating the effectiveness of the RCMs for each foreseeable sequence of events. If the evidence is not sufficient, the effectiveness shall be tested additionally e.g. with the simulation of the foreseeable sequence of events or via the usability verification and validation.

Risk Management File (RMF) is an extensive set of direct RM and indirect design control documents in connection with the RM process. Finally the content of our RMF has been defined based on the various standards and our development documentation structure. RMF is realized within the Design History File documentation database via an attribute ensuring a specific view for this file.

### 3.3 Software Design Risk Management

As stated above, software RM activities cannot be effectively performed unless they are integrated into the software development life-cycle. Furthermore according to IEC 62304 the level of detail of the corresponding activities depends on the safety classification of the software components and units (defined during the software architecture design). With an appropriate system and software safety architecture the safety-related and non-safety-related software components and units can clearly be separated. So risk assessment can be performed mainly top-down on system level for the failure of software functions. In such a way detailed bottom-up software design related risk assessment is required only in such cases, where:

- the proper segregation (without common mode failure) cannot be easily proven or
- if the independence of a software risk control measure in a double channel system is questionable.

This analysis or review can be done with an FMEA method.

To prevent further coding and functional hazards the following SW unit level testing activities are applied in the SW life-cycle:

- Static Code Analysis according to Misra C.
- Unit testing on object and task level.
- Code Review, which after the previous steps rather can focus on the functionality.

The next step is the SW unit integration testing, which links back to the requirement management system by testing the software requirements relevant to the software unit integrated into the SW component. In this paper the further software development phases and other software life-cycle processes (maintenance, problem resolution, configuration management) are not discussed.

### 3.4 Usability Engineering Process

One of the main goals of the usability engineering process is to specify usability objectives for the so called Primary Operating Functions in the Usability Specification. These functions are defined according to the usage process activity analysis, as frequently used functions and based on the RM process, as safety-related and user-dependent functions. Safety-related functions are user activities having at least one foreseeable sequence of event which exceeds the acceptable region in the risk assessment after evaluating measures. User-dependent functions include a user action, as a RCM, which cannot be associated with any hardware or software measures, and it has at least one foreseeable sequence of event that exceeds the acceptable region after evaluating measures. Since the measures for the safety-related and user-dependent functions are in connection with the user, their effectiveness can only be verified by usability verification and validation. The scale for usability objectives were determined by the probability of the corresponding foreseeable sequence of event in the risk assessment in such a way, that the success rate shall be higher in case of higher risk probability.

## 4 Possible Tool Chain

Our tool chain is continuously extended and refined along the process. It has already provided an applicable evidence of carrying out RM tasks throughout the whole development life-cycle including connections to the requirement management, software life-cycle, effectiveness verification and usability engineering processes. An SQL based risk management tool is used for performing the Risk Assessment. This database provides an easy and automatic way to transfer the result into our requirement management system, where all the system and component level documentation (specifications) and their verification are managed.

For controlling the implementation of the SW requirements (also the risk control measures) an internally developed database is used as change management tool. It is used to assign the target functionality of a SW unit (object, task or screen) for the next release. SW unit level documentation (SW detailed design, unit verification, unit design review) is handled in this database. The link between the requirement management system and this database is ensured by the database record IDs.

For SW unit level testing a central static code analyzer is used, which creates feedback to the developers based on the daily test runs. Additionally a local code analyzer is also partially introduced to give the possibility for the developers to get an immediate feedback about their implementation. Unit testing is performed using an internally developed unit test framework on object level, thus reducing the number of units to test.

## 5 Conclusion

We have a positive experience with the above described Integrated Risk Management process, although few areas are not applied yet. Our aim is to continuously extend, refine and further automate the process. An important goal for the future is to monitor the RCM implementation coverage automatically, based on data in the requirement management system.

## References


1. ISO 14971:2007 Medical devices - Application of risk management to medical devices
2. IEC 60601-1:2005 Medical electrical equipment - Part1: General requirements for basic safety and essential performance
3. IEC 62304:2006 Medical device software - Software life-cycle processes
4. IEC 62366:2007 Medical devices - Application of usability engineering to medical devices
5. IEC/TR 80002-1:2009 Medical device software - Part1: Guidance on the application of ISO 14971 to medical device software

# Author Index