Dietmar Winkler
Rory V. O'Connor
Richard Messnarz (Eds.)

Communications in Computer and Information Science 301

Systems, Software and Services Process Improvement

19th European Conference, EuroSPI 2012
Vienna, Austria, June 2012
Proceedings

Springer

Communications
in Computer and Information Science 301

Dietmar Winkler Rory V. O'Connor
Richard Messnarz (Eds.)

Systems, Software and Services Process Improvement

19th European Conference, EuroSPI 2012
Vienna, Austria, June 25-27, 2012
Proceedings

Springer

Volume Editors

Dietmar Winkler
Institute of Software Technology
and Interactive Systems
Vienna University of Technology, Austria
E-mail: dietmar.winkler@tuwien.ac.at

Rory V. O'Connor
Lero, The Irish Software Engineering
Research Centre, School of Computing
Dublin City University, Ireland
E-mail: roconnor@computing.dcu.ie

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

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-31198-7 e-ISBN 978-3-642-31199-4
DOI 10.1007/978-3-642-31199-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

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 19th EuroSPI Conference, held during June 25–27, 2012, in Vienna, Austria.

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 packaged SPI knowledge in job role training and established a European certification association (www.ecqa.org) to transport this knowledge Europe wide using standardized certification and exam 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, and in Roskilde (Denmark) in 2011.

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 professionsrelated with SPI and management. This is supported by European certificatesand 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 availableto 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 and Business Factors
- Section II: SPI Lifecycle and Models
- Section III: SPI Assessment and Quality
- Section IV: SPI Processed and Standards
- Section V: SPI in SMEs
- Section VI: SPI and Implementation
- Section VII: Selected Key Notes and Workshop Papers

Section I presents studies "On SPI and Business Factors." Clarke and O'Connor examine the role of SPI in business success in software SMEs and provide recommendations for future SPI studies. Sussy et al. present the critical success factors to take into account in the deployment process, and a method of process deployment to be used in software projects; they also highlight the importance of having an effective deployment strategy to adopt, use and institutionalize the process.Yilmaz and O'Connor consider software development as an economic activity, where goods and services can be modelled as a resource-constrained task-allocation problem and introduce a market-based mechanism to overcome task-allocation issues in a software development process.

Section II presents three papers on "SPI Lifecycle and Models." Lacheiner and Ramlerpresent an approach to detect process conformance violations that reveal deviations between planned and executed software engineering processes, an approach based on process rules that complement the process documentation.Heidenberg et al. propose a model for business value that is intended to make explicit different factors that constitute the concept of business value in agile and lean software development. In the final paper of this section,Heikkinenand Jantti address the research questionof which methods and practices are related to continuous service improvement in IT service management and they describe an improved version of the CSI model that provides a more detailed and practical view of CSI activities.

Section III presents papers related to "SPI Assessment and Quality." In the first of three papers, Woronowicz et al. present an introduction to the innoSPICE Model, the assessment methodology, and provide an initial analysis of the innoSPICE assessments performed so far.Nikitina and Kajko-Mattssonevaluate the conditions necessary for succeeding with the SPI implementations and sustaining their results by providing an SPI health checklist. Finally, Lami et al. discuss the sustainability of software processes by defining a core set of processes that represent the activities to be performed in order to introduce and integrate the greenness culture in a software-developing organization.

Section IV explores “SPI Processes and Standards.” Mercedes de la Cámara et al. present the research context and the results of finding out how PRINCE2 meets the expectations of IT governance and management according to ISO 38500 and ISO 20000 standards in order to achieve the success of IT projects. Larrucea et al. focus on providing a harmonized framework not only covering practice-based process models but also covering product characteristics, and they reconcile this framework for safety critical systems.Finally, in the third paper, García-Mireles et al. present the result of mapping models based on both (process and product) quality perspectives and a mapping of ISO/IEC 25010 onto CMMI-DEV and ISO/IEC 12207.

Section V presents three related papers on the topic of “SPI in SMEs.” Miler and Wesołowskapresent a method focused on improvements of task management using process models and the results of applying the method in two case studies with SMEs. Renato Ferraz Machado et al. propose a maturity model for IT service management, called MM-GSTI, which is compliant to ISO/IEC 20000 and CMMI-SVC, whose goal is to help service providers in the implementation of improvements for the management of IT services. Boucher et al. address the shortcomings of ISO/IEC 29110 and discuss profiles in an integrated and configurable workflow with illustrations on the requirements engineering activity.

Section VI discusses “SPI and Implementation” issues. Toroi et al. present how functional defect analysis can be applied for software process improvement purposes. Van Stijn et al. present a template for such a structuring method, based on UML use case descriptions and method engineering techniques, together with a case study of two large improvements within a small Dutch software company. Finally, Jenerset al. discuss the integration of multiple reference models based on automated concept extraction.

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

Three invited papers concerning “Creating Environments Supporting Innovation and Improvement” illustrate that SPI is inherently linked to innovation and that innovation requires constant change. Peisl and Schmied discuss how to innovate the current innovation principles and how this impacts the future of SPI thinking. Riel, Neumann et al. describe how a concept of open innovation and an environment supporting idea creation can lead to improvement and innovation in leading European industry.Kerstin Siakas et al. discuss which competencies in the field of valorization and exploitation are needed to successfully roll out a innovation and improvement (EU project VALO).

Two invited papers concerning “Standards and Experiences with the Implementation of Functional Safety” illustrate that SPI in systems development (e.g., automotive industry, aerospace industry, and medical industry, etc.) needs to consider safety design-related competencies and needs to integrate ISO 15504 assessments with functional safety standards such as IEC 61508 and ISO 26262. Andreas Riel et al. describe results from a European research project that develops the core competencies required to cover the functional safety standards and incorporate them into the SPI programs in firms. Messnarz, Bachmann et al. describe the results from first trial assessments combining ISO 15504, Automotive SPICE and IEC 61508, ISO 26262.

Three invited papers concerning “Business Process Innovation and Improvement” address new approaches on how to apply SPI principles at a corporate business level. While in the past SPI was a software and system development-related field, it has become more and more a business development topic with world-wide recognition at a business standard level. AncaDraghici et al. describe experiences in university and industry with applying business process modelling principles developed in the EU project CertiBPM. Ivanyos et al. describe how international financial standards are used for SPICE compliant assessments, process modelling and system support improving the business processes and governance of companies (EU project GOSPEL). NájeraVillar and Brändle describe experiences with key factors for organizational learning at a corporate level and how terminology management strategies support such improvements.

Three invited papers illustrate SPI in small and medium enterprises (SMEs) from a project management perspective.Lepmets and McBride consider the value of an organization’s strategic goal for small and agile settings, while Calvo-Manzano and Caballero show how a very small enterprise has tailored Scrum according to its own needs. Finally, O’Connor and Laporte discuss the role and structure of project management in the emerging ISO/IEC 29110 standard.

June 2012

Dietmar Winkler
Rory V. O’Connor
Richard Messnarz

Recommended Further Reading

In [1] the proceedings of three EuroSPI^2 conferences were integrated into one book, which was edited by 30 experts in Europe. The proceedings of EuroSPI^2 2005 to 2011 inclusive have been published by Springer in [2], [3], [4], [5], [6] [7] and [8], 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, p. 213. Springer, Heidelberg (2005)
3. Richardson, I., Runeson, P., Messnarz, R. (eds.): EuroSPI 2006. LNCS, vol. 4257, pp. 11–13. Springer, Heidelberg (2006)
4. Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.): EuroSPI 2007. LNCS, vol. 4764, pp. 1–6. Springer, Heidelberg (2007)
5. O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.): Software Process Improvement. CCIS, vol. 16. Springer, Heidelberg (2008)
6. O'Connor, R.V., Baddoo, N., Gallego, C., RejasMuslera, R., Smolander, K., Messnarz, R. (eds.): Software Process Improvement. CCIS, vol. 42. Springer, Heidelberg (2009)
7. Riel, A., O'Connor, R.V., Tichkiewitch, S., Messnarz, R. (eds.): Software, System, and Service Process Improvement. CCIS, vol. 99. Springer, Heidelberg (2010)
8. O'Connor, R., Pries-Heje, J., Messnarz, R.: Systems, Software and Services Process Improvement. CCIS, vol. 172. Springer, Heidelberg (2011)

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 11 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. EuroSPI^2 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
Guenter Ruhe	University of Calgary, Australia
Ivo Vondrak	Technical University of Ostrava, Czech Republic
Jan Pries-Heje	Roskilde Universitet, Denmark
Javier Garíca-Guzman	Carlos III University of Madrid, Spain

Jose Antonio Calvo-Manzano	Universidad Politecnica de Madrid, Spain
Jürgen Münch	University of Helsinki, Finland
Kai Stapel	Leibniz Universität Hannover, Germany
Keith Phalp	Bournemouth University, UK
Kerstin Siakas	Alexander Technological Educational Institute of Thessaloniki, Greece
Luigi Buglione	Engineering Ingegneria Informatica, Italy
Marion Lepmets	CRP Henri Tudor, Luxembourg
Markku Oivo	University of Oulu, Finland
Michael Reiner	IMC Fachhochschule Krems, Austria
MiklósBiró	Software Competence Center Hagenberg, Austria
Patricia McQuaid	California Polytechnic State University, USA
Paul Clarke	Lero, Irish Software Engineering Research Centre, 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
TimoMäkinen	Tampere University of Technology, Finland
Timo Varkoi	Tampere University of Technology, Finland
Tom Peisl	HochschuleMünchen, Germany
Tony Gorschek	Blekinge Institute of Technology, Sweden
Torgeir Dingsøyr	SINTEF ICT, Norway
Valentine Casey	Dundalk Institute of Technology, Ireland
Vincenzo Ambriola	Universitá di Pisa, Italy

General Chair

Richard Messnarz

Scientific Chairs

Dietmar Winkler
Rory V. O'Connor

All three Chairs, the General and the Scientific Chairs, have quite a complementary and interesting profile. Dr. Messnarz works in close collaboration with Austrian research institutions (universities of applied sciences) and large German automotive companies. Dietmar Winkler is a key researcher in the Christian Doppler Laboratory "Software Engineering Integration for Flexible Automation Systems" at the Institute of Software Technology and Interactive Systems at Vienna University of Technology. His research interests focus on software processes, SPI, quality assurance and quality management, and empirical software engineering. Dr. Rory O'Connor is a seniorlecturer in Dublin City University and a senior researcher with Lero, the Irish Software Engineering Centre.

His main research interests center on software processes and SPI in relation to small and very small organizations.

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^2 to cover a variety of different markets, experiences, and approaches.

Acknowledgements

Some contributions published in this book have been funded with support from the European Commission.

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

Table of Contents

SPI and Business Factors

SPI Lifecycle and Models

SPI Assessment and Quality

SPI Processes and Standards

SPI in SMEs

SPI and Implementation

Selected Key Notes and Workshop Papers

Creating Environments Supporting Innovation and Improvement

Standards and Experiences with the Implementation of Functional Safety

Business Process Management (CertiBPM)

SPI in SMEs – A Project Management Perspective

Business Success in Software SMEs: Recommendations for Future SPI Studies

Paul Clarke[1] and Rory V. O'Connor[2,3]

[1] Lero Graduate School in Software Engineering, Dublin City University, Ireland
pclarke@computing.dcu.ie
[2] Dublin City University, Ireland
[3] Lero, The Irish Software Engineering Research Centre
roconnor@computing.dcu.ie

Abstract. There is presently insufficient data regarding the relationship between software process improvement (SPI) and business success, a fact which may reduce process prioritisation in software development in practice. To assist future studies examining the relationship between SPI and business success, we developed a new two-phased approach to examining business success. The first phase involves the elicitation of business objectives for the forthcoming year, with the second phase determining the extent of achievement of the recorded objectives. At EuroSPI 2011, we described the two-phased approach in detail and reported on the findings from deploying the first phase of the examination to software developing small- and medium-sized enterprises (software SMEs). In this follow-up paper, we report on the findings from the second phase of the investigation in the participating software SMEs, formulating an additional important new recommendation for future studies.

Keywords: Software Process Improvement, Business Success, Software SMEs.

1 Introduction

Business processes are the routines or activities that firms adopt in order to conduct their business [1], with various empirical studies demonstrating that business process management has a positive effect on business success [2], [3]. Within software development organisations, the software development process is a large and complex component of the overall business process, and therefore, it follows that software process management should also have a positive effect on business success. One of the principle vehicles of software process management is the domain that is commonly referred to as software process improvement (SPI).

Earlier research has demonstrated that software companies can benefit from SPI programs, including financial benefits, such as return on investment (ROI) [4]. While Van Solingen [4] examined large software development organisations, other research demonstrated that software SMEs can also derive benefits from SPI, including improvements in quality, schedule and budget adherence [5], [6]. However, none of the earlier research has focused on examining the influence of SPI on business

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 1–12, 2012.

success, hence it has been reported that there is a lack of direct evidence of the business benefits of SPI [7], resulting in some software companies choosing to implement SPI in response to negative business events alone [8]. Therefore, there is a need to conduct studies that investigate the relationship between SPI and business success – and members of the software process and SPI communities would expect that such studies would highlight the important role of SPI in creating competitive advantages and thus in supporting business success. Future studies examining the relationship between business success and SPI will require a robust, thorough and reliable method for making determinations in relation to business success.

In an earlier published work, we identified a new approach to examining business success in software development organisations [9]. Our approach recommended using a two-phased engagement with companies when making determinations in relation to the degree of business success. The first phase identifies the objectives for a forthcoming period (say for example, 1 year), with the second phase returning to the organisation at the end of the period and examining the extent to which the recorded objectives were achieved. In our earlier published paper, we reported on our experience of applying the first phase to seventeen software SMEs, making a number of recommendations for later studies. In this paper, we report on the findings from the second phase of the business success inquiry, extending our recommendations for later studies seeking to examine business success in software development organisations.

The remainder of this paper is structured as follows: Section 2 presents a brief review of the two-phased technique for examining business success in software companies, along with details of the application of the first phase to participating software SMEs. In Section 3, we describe the second phase of the business success investigation, identifying objectives with the highest and lowest degree of achievement. Finally, in Section 4, we present a summary and a conclusion.

2 Study Background

This section provides an overview of the business success literature, along with a brief review of the two-phased business success examination. In addition, this section presents a summary of the results from the implementation of the first phase of the examination in software SMEs (more comprehensive details are available in [9]).

2.1 Business Success in Software Development Companies

The domain of business success, sometimes referred to as business performance, is multi-faceted. Historically, businesses took the view that only financial measures of business success were of importance [10]. Such financial measures include profitability, ROI, and productivity [11-14]. However, the pursuit of profit is not the only purpose that a company must address [15] and a number of other important non-financial measures of business success also exist [16], [17]. Such non-financial measures include customer satisfaction and business process management. Collectively, the financial and non-financial aspects of business success are addressed in multi-dimensional business performance measurement frameworks [18]. A number of multi-dimensional business performance measurement frameworks have been developed [19-22], with the Balanced Scorecard (BSC) [23] approach being the most popular [24]. Although the BSC has been criticised as being impractical for use in small companies [25-27], it has also been noted that SMEs can obtain benefits by using the components of the scorecard as a frame of reference for implementing business success investigations [28]. The creators of the Holistic Scorecard (HSC) [29] extended the BSC to include specific business success considerations for software development companies (refer to Figure 1).

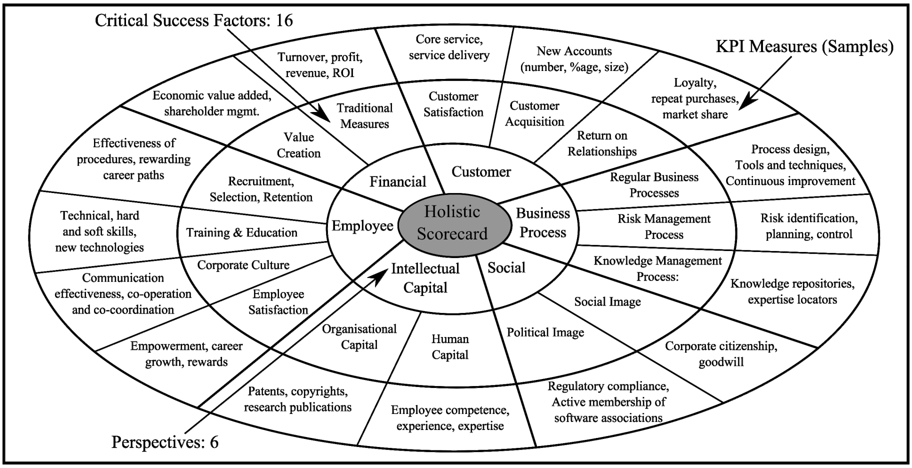


Fig. 1. Holistic Scorecard Overview

2.2 Examining Business Success in Software SMEs Using the HSC

A comprehensive survey instrument was developed using the HSC as a reference framework. In order to minimise the effect of biased or false recollection, a two-phased investigation was designed (refer to Figure 2). In Phase 1 of the investigation, the business objectives are elicited for the forthcoming period. In Phase 2 of the investigation, the extent of achievement of the business objectives is determined.

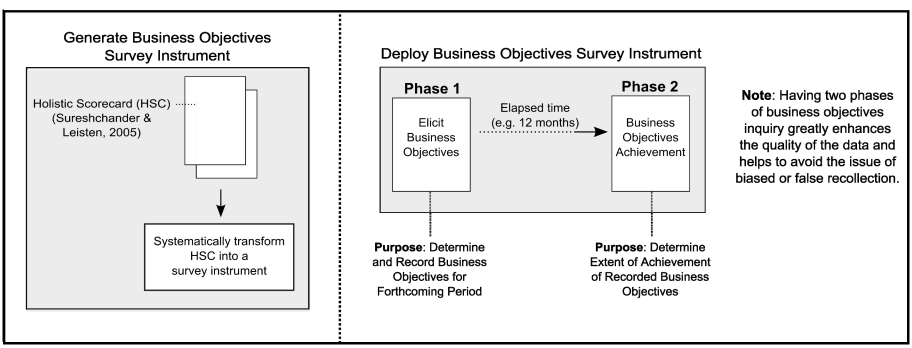


Fig. 2. Using the HSC in a two-phased business success investigation

2.3 Phase 1 – Business Objectives Elicitation in Software SMEs

The first phase of the survey instrument was deployed to 17 SMEs in February-June 2010. An analysis of the data permitted the development of a hierarchy of business objectives for software SMEs (refer to Figure 3), plus two recommendations for future studies:

Recommendation 1. If a future study of business objectives in SMEs were to use the HSC (or the HSC-based survey instrument produced by this research), the researchers could consider removing or consolidating the objectives that are in the lowest tier of the hierarchy in figure 3.

Recommendation 2. Future research into the business objectives in software companies should include questions relating to objectives in the areas of (1) financial liquidity (sometimes termed cash flow); (2) off-shoring or outsourcing some aspects of the development work; (3) mergers and acquisitions (M&A).

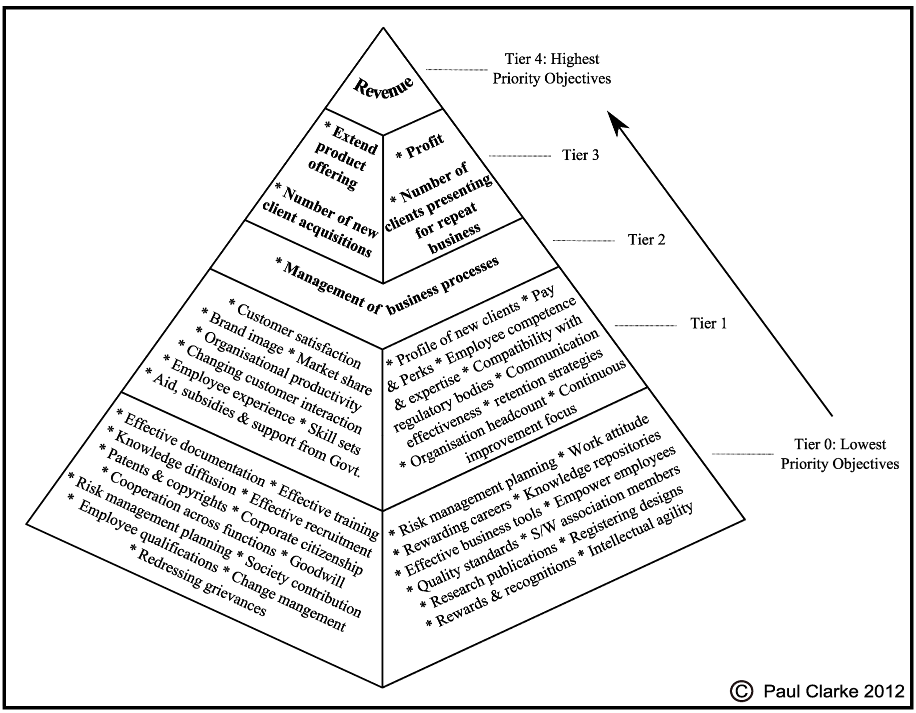


Fig. 3. Hierarchy of HSC Business Objectives for Software SMEs

In the following section, we report on the findings from Phase 2 of the business objectives examination, identifying the extent of achievement of business objectives in software SMEs. Furthermore, we make an additional new recommendation for future studies of business success in software companies.

3 Phase 2 – Extent of Achievement of Business Objectives in Software SMEs

In this section, we report on the additional contribution of this paper: the results of Phase 2 of the business success examination in the participating software SMEs. In the period February to June 2011, we returned to the participating organisations, this time discharging the second phase of the investigation – to examine the extent of achievement of the recorded objectives. Two of the participating organisations were unable to participate in the second phase of the investigation, citing business pressures as an obstacle to setting aside time for the interview. Therefore, fifteen companies participated in the second phase of the investigation, with the interview time being approximately 45 minutes per company.

All of the fifteen organisations satisfied the European Commission definition of an SME [30]. Within each of the participating organisations, a suitable participant was identified; most commonly, the interviewee held the job title of Managing Director (though other Director-level job titles were also involved). The participating software SMEs varied in terms of the headcount: 3 of the participating companies had less than 10 staff; 4 companies had between 10 and 19 staff; the remaining 8 companies had between 20 and 129 staff. Predominately, the participating organisations were primarily located in Ireland. However, in some cases, the organisations were mostly located outside of Ireland, in places such as the US and Chile. The participating companies operated in diverse business domains. Four of the organisations developed web-based software, with another four organisations developing software for the telecommunications domain. The remainder of the organisations operated in a variety of different sectors, including, content management, data mediation, and embedded software.

Where possible, the interviews were conducted face-to-face with telephone interviews being employed in a small number of cases (for example, where the interviewee was based in a remote location). Irrespective of whether the interview was conducted face-to-face or via telephone, the interview was (with the consent of the interviewee) recorded and later, the interview recording was carefully examined to ensure that the responses of the interviewee were accurately and completely documented in electronic form. For each of the objectives recorded in Phase 1 of inquiry, the participant was asked to identify the extent of achievement of the objective on a four-point Likert scale (refer to Table 1).

Table 1. Achievement Rating Scale for Business Objectives

Achievement Value	Achievement Interpretation
0	Not achieved to any extent
1	Partially achieved
2	Mostly achieved
3	Totally achieved

The data collected in Phase 2 was carefully analysed using both spreadsheets and voice recording analysis. This analysis reveals that there are a number of areas where software SMEs are strong at achieving their business objectives, while there are other areas where software SMEs are not as successful at achieving their objectives. An overview of the results of our analysis is presented in Figure 4, with the following sections dedicated to evaluating the results of our analysis.[1]

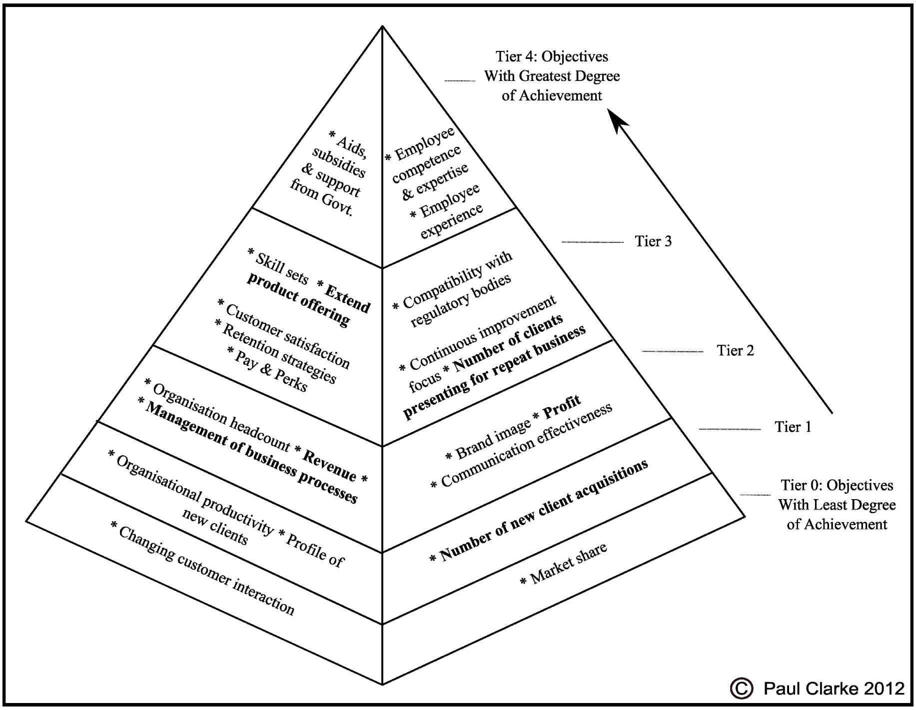


Fig. 4. Hierarchy of achievement of HSC Business Objectives for Software SMEs[2]

3.1 Objectives with Greatest Degree of Achievement

Having conducted a careful and thorough analysis of the data, we have identified eleven objectives that software SMEs tend to be most successful in achieving. The highest degree of achievement was in respect of the objectives in relation to obtaining aids, subsidies and support from government. This finding was slightly surprising, as in the first phase of the investigation, some of the participating organisations had highlighted that they often had difficulty in obtaining financial support and assistance from government. Perhaps the success in this area is related to the prevailing business conditions at present. With many organisations struggling against a headwind of challenging broader economic conditions, it is likely the case that software SMEs simply have to be as successful as possible in obtaining the maximum possible support from governments.

[1] Since Phase 1 of the study highlighted that software SMEs tend not to have objectives on Tier 0 of the hierarchy of objectives (Figure 3), the analysis herein focuses on Tiers 1-4.

[2] Note that objectives that are in **bold** in Figure 4 are from the top three tiers of Figure 3.

Our analyses also find that software SMEs are quite strong at achieving their objectives in relation to employees, specifically in relation to the competence, expertise and experience of employees. Again, this could be related to the need for software development companies to obtain the maximum possible value from employees in what are very challenging economic conditions. However, a deeper analysis of this finding demonstrates that in the first instance, the participating organisations did not have particularly strong objectives in relation to employee competence, expertise and experience – the initial objective was related to sustaining a position whereby employees improved their competence, expertise and experience as a natural outcome of working in a fast moving and dynamic SME environment. It is therefore the case that the initial objectives in the area of employees were not strong and were largely concerned with natural outcomes. We encountered a similar outcome with employee skill sets, though the extent of achievement for skill sets was somewhat lower than that recorded for competence, expertise and experience. This type of finding was not anticipated when we originally discharged the first phase of the investigation and as a result, we make the following recommendation for future studies:

Recommendation 3. If a future study of business objectives in software companies were to use the HSC (or the HSC-based survey instrument produced by this research), the researchers should take care to ensure that employee-related objectives in the areas of competency, expertise, experience and skill sets are distinct objectives beyond the increasing competency, expertise and experience that are accrued as part of routine working arrangements.

Our analyses also demonstrate that software SMEs are relatively successful when it comes to achieving business objectives in the area of compliance with regulatory bodies. In the case of the participating organisations, a number of the individual companies operated in business domains wherein regulatory compliance was a pre-requisite for business – for example, certain telecommunications and confidential data processing systems. Therefore, it is not altogether surprising to discover that the participating SMEs report that in general they have been successful in terms of satisfying the regulatory bodies associated with their business domain. We also found that software SMEs are successful when it comes to extending their product offering. Many of the participating organisations had explicit new features and capabilities that we identified as objectives from the first phase – and in most cases, the participating organisations were successful in implementing the features or products. Of all the high priority objectives identified in the first phase (those objectives on Tiers 3 and 4 of the hierarchy in Figure 3), the participating companies were most successful in terms of implementing new product base or in enhancing existing product base.

We also found that among the participating organisations, companies were reasonably successful in terms of their objectives in relation to customer satisfaction levels and gaining repeat business from existing clients. These two objectives would appear to be related, since if a client is satisfied, they are also more likely to present for repeat business. In relation to employee retention strategies and pay and perks, we found that the participating companies were generally achieving the objectives that they had identified during the initial investigation phase.

3.2 Objectives with Lowest Degree of Achievement

While the participating companies were most successful in areas such as obtaining aids, subsidies and support from government, and in terms of extension to product offerings and generating repeat business from existing clients, there are a number of other areas in which the objectives were not as successfully met. Notably, the companies were not quite as successful when it came to meeting revenue and profit targets, or in terms of objectives in relation to the business process. The broad view that we can establish from the analysis is that software SMEs work hard at retaining and extending business with existing clients, but that other aspects of their business objectives become much more difficult to realise. In addition to areas such as revenue and profit, we also found that the participating SMEs are less successful again when it comes to hitting targets for new client acquisitions. These observations highlight some interesting aspects of the software SME sector.

Firstly, none of the participating organisations are listed on a stock exchange and therefore, they are not subject to the predictability of revenue and profit targets that are generally demanded by the markets. As a result, aggressive revenue and profit targets may be set by small company owners – since there is no immediate negative funding impact from failure to achieve objectives. Or perhaps it is also the case that there is not a great deal of oversight of the original financial objectives with a view to tempering them against the whims of the principle agent, the owner. We must also highlight that small software development companies are often involved in market creation and innovative product development – the results from which can be difficult to predict in advance.

Secondly, small businesses are like any other general type of business in one key respect – in that it tends to be less difficult and less costly to obtain new business from existing clients than it is to secure entirely new clients [31]. Once a relationship is established and trust is in place, it is more likely that a customer will be prepared to do business with an established supplier with existing delivery experience. This particular issue could be exacerbated in software SMEs that are trying to convince potential new clients of the benefits of their innovative new product – they may first have to work hard to create the market for the new product, something that is acknowledged as being a significant challenge [32]. We also found in the study that among the participating companies, there was a low degree of success in terms of achieving objectives concerning the profile of new clients. This, we believe, further emphasises the dependence that software SMEs can have on existing clients, and the difficulty that they can experience when trying to attract new customers, especially if the profile of the desired customer base is divergent from the existing customer base. Therefore, small companies may be much more exposed to the demands of a few key clients – something that is further evidenced by the relatively low degree of success that the participating organisations had in terms of changing the nature of customer interaction. One final observation in relation to the difficulty of securing new clients is the fact that the participating organisations were overwhelmingly falling short of the market share objectives that were captured during the first phase engagement.

Our analyses also revealed that the participating organisations were not entirely successful in terms of achieving their organizational productivity objectives. However, while a significant number of organisations expressed objectives in relation to organizational productivity in the first phase of the inquiry, for the most part these objectives were of the lowest order of priority. Therefore, the achievement (or non-achievement) of organisational productivity objectives would not appear to be significant in the overall scheme of business success for the participating companies. A summarised view of the findings from the data analysis is presented in Figure 4.

4 Summary and Conclusions

As highlighted in the introduction for this paper, there is presently a need to examine the role of SPI in terms of supporting business success. Numerous earlier studies have demonstrated that there are benefits to be gained from SPI activities, including improvements to quality, budget and schedule adherence. Despite these noted benefits, it has also been reported that some software development companies can have a low process priority [7], and that software SMEs appear to conduct SPI in response to negative business events [8].

In order to support future studies examining the influence of SPI on business success, we developed a new approach to examining business success in software development companies. This new approach utilises the HSC [29] as a reference framework, ensuring that the broad spectrum of multi-dimensional business performance considerations (i.e. financial and non-financial) are included in the business success examination. Furthermore, the inclusion of employee and intellectual capital related business performance criteria in the HSC ensures that all of the major software development business success considerations are included in our new approach. From the HSC, we systematically derived a survey instrument that is deployed in two phases – Phase 1 determines the business objectives for the forthcoming period (e.g. 1 year), with Phase 2 later examining the extent of achievement of the recorded objectives. The outcomes from discharging Phase 1 in the field, which were presented at EuroSPI 2011 [9], found that the HSC is a comprehensive and useful framework for examining business success in software SMEs. From this exercise, we determined that software SMEs tend to have high priority business objectives in the areas of revenue, profit, extending product offerings, number of new client acquisitions and the number of existing clients presenting for repeat business.

In this paper, we have reported the findings from discharging Phase 2 of the business success inquiry - determining the extent of achievement of the business objectives. Some of the highest priority objectives, including revenue and profit, were not completely achieved which may suggest that the revenue and profit targets of software SMEs are unrealistic. Equally, the participating companies performed relatively poorly when it came to achieving the objectives for new client acquisitions. However, the participating companies were quite successful in meeting the targets for aids and subsidies from government, something that may have helped to address the shortfalls in revenue and profit targets. We also found that the participating companies were relatively successful in terms of extending their product offerings in line with objectives, and that they performed quite strongly when it came to meeting targets for obtaining repeat business from existing clients.

Overall, we find that the HSC is generally comprehensive in nature and appropriate for studies investigating business success in software companies. However, we do make a number of recommendations for future research into business success in software companies. Firstly, we recommend that future studies in software SMEs can consider focusing their business success inquiries on the top four tiers of the SME business objectives pyramid (refer to Figure 3). Secondly, we suggest that a number of new objectives are included in future business success investigations in software companies, including financial liquidity, off-shoring, outsourcing, and mergers and acquisitions. Thirdly, we recommend that future researchers take care to identify objectives that are a natural outcome of working (for example, improvements in experience, skill sets and competencies). The achievement of such objectives are not particularly strong indicators of business success but more a case of extending employee capabilities through regular working activities. Finally, we also suggest that future studies of business success in software companies include a closing question to elicit objectives that have possibly been overlooked by the HSC – a measure that was originally recommended by the creators of the HSC [29].

The authors consider the two-phased business success investigation to be superior to a single stage examination in software SME settings. The risk of false or biased recollection on the part of the participants is greatly reduced and it supports the formal elicitation of business objectives in an environment where they might not otherwise be recorded. However, the approach cannot claim to completely eliminate the risk of false or biased recollection, since it is possible that participants could be biased in their reporting of the extent of achievement of the recorded objectives. Equally, some of the objectives are subjective in nature – such as examining the extent to which improvements in customer satisfaction were achieved. While considerable care was taken to deploy the survey instruments in a consistent and clear fashion (and the researchers had no reason to doubt the feedback from the participants), false, biased and subjective responses do present as potential threats to the validity of our findings.

In conclusion, we believe that the business success investigation approach identified in this research is well suited to examinations of business success in software companies in general. We also believe that there is a need for future research to utilise this approach in an effort to better understand the role of SPI in supporting business success. Future studies that attempt to examine the influence of SPI on business success will require considerable effort, especially since the generalisability of findings can only be established through multi-organisational examinations. However, there is much to be gained from such research and the establishment of a positive association between SPI and business success – through empirical investigation – could transform the views of practitioners with respect to SPI. Rather than maintaining a low process priority and implementing SPI in a reactive fashion, equipped with empirical evidence of the business benefits of SPI, software companies might choose to be more proactive in terms of managing their software development process.

Acknowledgments. This work is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie). We would also like to thank Prof. Brian Leavy from the Dublin City University Business School from whom we received invaluable advice in relation the general structure of the two-phased business success investigation method.

References

1. Pall, G.: Quality process management. Prentice Hall, NJ (1987)
2. McCormack, K., Johnson, W.: Business process orientation - gaining the E-business competitive advantage. CRC Press, Florida (2001)
3. Skerlavaj, M., Stemberger, M.I., Skrinjar, R., Dimovski, V.: Organizational Learning culture—the Missing Link between Business Process Change and Organizational Performance. International Journal of Production Economics 106(2), 346–367 (2007)
4. Van Solingen, R.: Measuring the ROI of Software Process Improvement. IEEE Software 21(3), 32–38 (2004)
5. Sanders, M. (ed.): The SPIRE handbook. better, faster, cheaper software development in small organisations. Centre for Software Engineering Limited, DCU, Dublin, Ireland (1998)
6. Von Wangenheim, C.G., Weber, S., Hauck, J.C.R., Trentin, G.: Experiences on Establishing Software Processes in Small Companies. Information and Software Technology 48(9), 890–900 (2006)
7. Baddoo, N., Hall, T.: De-Motivators for Software Process Improvement: An Analysis of Practitioners' Views. J. of Sys. & Soft. 66(1), 23–33 (2003)
8. Coleman, G., O'Connor, R.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software 81(5), 772–784 (2008)
9. Clarke, P., O'Connor, R.V.: The Meaning of Success for Software SMEs: An Holistic Scorecard Based Approach. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 72–83. Springer, Heidelberg (2011)
10. Jennings, D.F., Seaman, S.L.: High and Low Levels of Organizational Adaptation: An Empirical Analysis of Strategy, Structure, and Performance. Strategic Management Journal 15(6), 459–475 (1994)
11. Morgan, R.E., Strong, C.A.: Business Performance and Dimensions of Strategic Orientation. Journal of Business Research 56(3), 163–176 (2003)
12. Ghalayini, A.M., Noble, J.S.: The Changing Basis of Performance Measurement. International Journal of Operations & Production Management 16(8), 63–80 (1996)
13. Ansoff, H.I.: Corporate strategy. McGraw-Hill, New York (1965)
14. Reid, G.C., Smith, J.A.: What Makes a New Business Start-Up Successful? Small Business Economics 14(3), 165–182 (2000)
15. Nonaka, I., Toyama, R.: The Theory of the Knowledge-Creating Firm: Subjectivity, Objectivity and Synthesis. Industrial and Corporate Change 14(3), 419–436 (2005)
16. Maidique, M.A., Zirger, B.J.: The New Product Learning Cycle. Research Policy 14(6), 299–313 (1985)
17. Hart, S.: Dimensions of Success in New Product Development: An Exploratory Investigation. J. of Marketing Management 9(1), 23–41 (1993)
18. Bourne, M., Mills, J., Wilcox, M., Neely, A., Platts, K.: Designing, Implementing and Updating Performance Measurement Systems. Int. Journal of Operations & Production Management 20(7), 754–771 (2000)
19. Lynch, R.L., Cross, K.F.: Measure up! yardstick for continuous improvement. Basil Blackwell, Cambridge (1990)
20. Brown, M.G.: Keeping score: Using the right metrics to drive world-class performance. Quality Resources (1996)
21. Kanji, G.K., Sa, P.M.: Kanji's Business Scorecard. Total Quality Management 13(1), 13–27 (2002)

22. Neely, A.D., Adams, C., Kennerley, M.: The performance prism: The scorecard for measuring and managing business success. Prentice Hall, London (2002)
23. Kaplan, R.S., Norton, D.P.: The Balanced Scorecard - Measures that Drive Performance. Harvard Business Review 70(1), 71–79 (1992)
24. Kennerley, M., Neely, A.: Performance measurement frameworks: A review. In: Business Performance Measurement - Theory and Practice. Cambridge University Press, Cambridge (2002)
25. Gautreau, A., Kleiner, B.H.: Recent Trends in Performance Measurement - the BSC Approach. Management Research News 24(3), 153–156 (2001)
26. McKenzie, F., Shilling, M.: Avoiding Performance Measurement Traps: Ensuring Effective Incentive Design and Implementation. Compensation and Benefits Review 30(4), 57–65 (1998)
27. Chow, C.W., Haddad, K.M., Williamson, J.E.: Applying the BSC to Small Companies. Management Accounting 79(2), 21–27 (1997)
28. Andersen, H., Cobbold, I., Lawrie, G.: BSC implementation in SMEs: reflection in literature and practice. In: Proceedings of the fourth SMESME Conference, pp. 103–112. Department of Production, 2GC Limited. Aalborg University, 9220 Aalborg, Denmark (2001)
29. Sureshchandar, G.S., Leisten, R.: Holistic Scorecard: Strategic Performance Measurement and Management in the Software Industry. Measuring Business Excellence 9(2), 12–29 (2005)
30. European Commission: Commission Recommendation of 6 may 2003 Concerning the Definition of Micro, Small and Medium-Sized Enterprises. 2003/361/EC. Journal of the European Union L(124), 36–41 (2003)
31. Osterwalder, A., Pigneur, Y.: An e-business model ontology for modeling e-business. In: Bled Electronic Commerce Conference, pp. 75–91 (2002)
32. Molina-Castillo, F., Munuera-Aleman, J.: The Joint Impact of Quality and Innovativeness on Short-Term New Product Performance. Industrial Marketing Management 38(8), 984–993 (2009)

MEDEPRO: A Method to Deploy Processes Focused on People

Sussy Bayona, Jose A. Calvo-Manzano, Gonzalo Cuevas, and Tomás San Feliu

Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software
Facultad de Informática, Universidad Politécnica de Madrid
28660 Boadilla del Monte, Madrid, España
sbayona@hotmail.com, {joseantonio.calvomanzano, tomas.sanfeliu}@upm.es, gcuevas@fi.upm.es

Abstract. Nowadays, organizations need to respond to customer demands with quality products and services. Methodologies and process models have been developed to help organizations to achieve these objectives. However, despite the efforts of organizations, they still come up against difficulties in their deployment and the processes are not institutionalized. Difficulty in process institutionalization arises due to the fact that most of these efforts focus on technical issues, and people issues are ignored. This paper presents the critical success factors to take into account in the process deployment, and a process deployment method to be used in software projects. These highlight the importance of having an effective process deployment strategy to adopt, use and institutionalize the process. The results of a case study are included.

Keywords: Process deployment, critical success factors, CMMI, change management.

1 Introduction

Nowadays, organizations need to respond to customer demand with quality products and services. Models and standards, such as Capability Maturity Model Integration (CMMI) [1] have been developed to help organizations to achieve their business objectives. This highlights the importance of having an effective process deployment strategy in order to adopt, use, and institutionalize processes. However, the implementation of these models and standards in organizations presents issues that include: (1) improvement efforts are not aligned with business goals, (2) lack of leadership and visible commitment to improvement efforts, (3) the process does not respond to business needs, and (4) efforts to implement technical aspects ignore strategies based on social aspects [2]. According to Niazi [3], the problem of process improvement is not the lack of standards or models, but the lack of a strategy to implement them. Failure to consider the social aspects of a strategy for process deployment threatens the institutionalization of the processes deployed. McDermid and Bennet [4] argue that human factors for software process improvement have been ignored. According to Zahran [5], the inadequacy of proposals on the implementation of process improvement is one of the most common reasons for failure of improvement initiatives.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 13–24, 2012.

Some issues arise when processes are deployed such as: (1) difficulty in identifying the difference between implementation and deployment, (2) human factors are ignored or only focus on technical aspects, (3) process deployment is a change but this change is not managed to minimize the change resistance, (4) staff do not participate in the process definitions, (5) processes are not suitable for the organization's needs, environment or culture, (6) processes deployed are not used, (7) organizations do not have a formal deployment methodology or (8) method used is not focused on human factors. The issues mentioned are already well known both in academic and industrial circles. The goal of this paper is to present a Process deployment Method that focuses on the critical success factors related to people in order to use and adopt the processes than have been deployed.

We conducted a survey in five software development and maintenance centers to find out how communication, training, staff involvement, and change management factors can motivate organizations to adopt the processes deployed. A research about evaluation strategies of Software Process Improvement, identify seven categories that including the survey technique, and statistical analysis that includes descriptive statistics where data are summarized numerically (e. g. mean, median, mode) or graphically (e. g. charts and graphs) [6].

This paper is organized as follows. Section two presents the research work context. Section three describes the proposed method "Method for Process Deployment in CMMI level 3 organizations" called MEDEPRO. A case study is described in section four. Finally, the conclusions are presented in section five.

2 Research Work Context

Process improvement is a program of activities designed to improve the performance and maturity of an organization´s processes, and the results of such a program [5]. *Process deployment* is focused on people. A set of *critical success factors* is associated with deploying the processes successfully. The goal of process deployment is to put into practice the processes stored in the *Process Asset Library (PAL)*. PAL, as a knowledge repository, helps software engineers to learn about development processes [7]. When organizations implement changes, resistance is a natural consequence [8] because the activities are carried out by people, so is necessary to manage change. Next, it is presented a description of process deployment, their critical success factors for process deployment, PAL and change management. Process deployment is centered on people at all levels: individual, team, group, organizational, country, and cultural. The purpose of process deployment is to get people to use the new processes. It is necessary to recognize the difference between *process implementation* and *process deployment*. The concept of deployment goes beyond the single instantiation of an implemented process, to address the effective deployment of a process specification to achieve multiple implementations of the process across an organization [9].

Diverse methodologies and process models have been developed to help organizations to implement improvement initiatives. Some of this guidance has been developed under the imprimatur of governments, major research universities and international organizations such as: IDEAL (is about the way to implement a process improvement program), ISO 15504 (SPICE) or CMMI. However, organizations have problems with the institutionalization of process because the problem of process improvement is not the lack of standards or models, but the lack of a strategy to implement them [3]. This has motivated the interest of researchers from the International Process Research Consortium (IPRC) [9] who have included Process Deployment in a list of research items. One reason is that process deployment is related to the people, and intensive research into the human factor and change management is needed.

Organizations in their current deployment processes : (1) often use the models and standards as strict criteria for awarding contracts and assessing maturity, (2) the process is defined using international models and standards and they are not aligned to the strategic goals of the business, (3) the processes are defined taking into account the activities carried out in the organization, (4) the status preparation for change is not evaluated, and (5) the critical success factors are ignored in the implementation strategy. According to our research work [10] process deployment elements are: the organization, PAL, process, people, process deployment, reference model and change management. Figure 1 shows the Process Deployment elements.

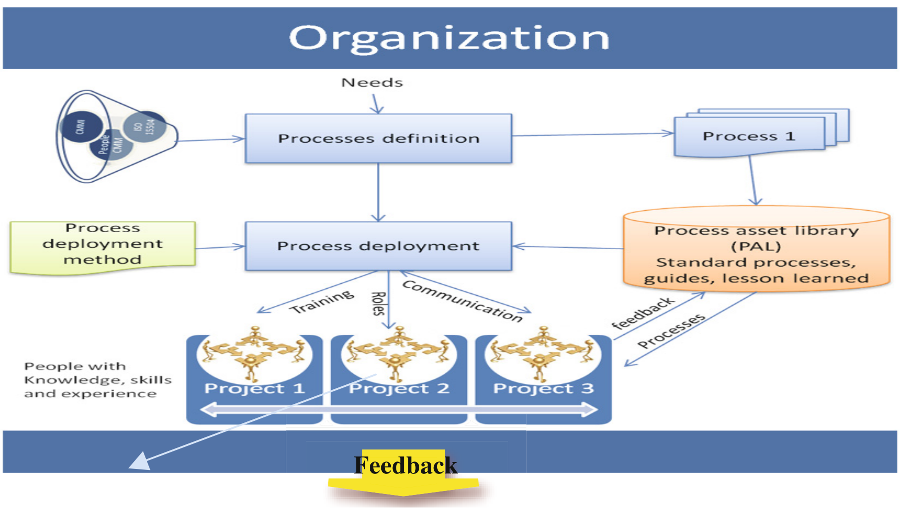


Fig. 1. Process Deployment elements

It is important to know the relationship among process, projects and people. People are involved in software projects in which staff with skills, abilities, and motivation are needed. Software projects use the processes that are contained in the organizational Process Asset Library. PAL contains the processes, models, standards, procedures and adaptation guides, metrics and lessons learned that will be used in projects. These processes require roles to develop the tasks. People that develop software need to know the processes and the adaptation guides. As a result of the deployment process, the lessons learned and the improved processes are documented and included in PAL. PAL is the input of the process deployment. Therefore, it is necessary for the processes to be well defined. If not, the process deployment will fail. There is a set of critical success factors associated with process improvement and process deployment initiatives. A successful strategy for deploying new or updated processes must take these factors into account; otherwise, the goals cannot be achieved. To identify the critical success factors in process improvement a systematic review was conducted. The method used was in accordance with systematic review guidelines [11].

The goals of the systematic review were to indentify (1) the critical success factors associated with the implementation / deployment process of software processes, and (2) the categories used by different authors to classify them. We reviewed the inventory of critical success factors (identified by systematic review) and ranked these factors according to the frequency mentioned in the primary studies (only were considered primary studies). The critical success factors identified for process improvement initiatives are: commitment, alignment with the business strategy and goals, training, communication, resources, skills, staff involvement, improvement management, process definition, monitoring software process improvement process, change management, culture, policies, roles and responsibilities, tools and mentoring [12]. Having identified the critical success factors for the deployment process, we propose a method that integrates these factors to ensure the success of the deployment process. In this paper we focus in change management, training, communication, and staff involvement critical success factors.

2.1 Change Management

Software process improvement is inherently linked with the change [13]. Implementing new or updated processes entails change. This change must be managed to minimize the resistance to change, which is a natural human phenomenon [14] [15]. Resistance to change is a critical success factor in process improvement [16], because the activities are carried out by people. To deploy the processes and manage change, the organization must take into account the change management components such as communication at all levels, staff training, resistance management, coaching. The obvious way to deploy, or promote, the software process is training, but there are several other means available. People are different in their readiness to try new things, for instance new products or processes, and to respond to change. According to Rogers [17], people can be classified into five adopter categories: innovators, early adopters, early majority, late majority, and laggards. Change agents will seek out early adopters to help speed up the process deployment plan and the benefits.

2.2 Training

Training is provided to develop the skills and knowledge needed to implement a software process improvement initiative [18].Sufficient resources, additional time to participate in software process improvement and training will be provided to staff members. Lack of training does not allow people to gain the skills to develop the new tasks. It is necessary establish the difference between learn (it means to acquire or gain skill, knowledge or comprehension) and teach (it means to impart skill, knowledge or comprehension).

2.3 Communication

Communication is a main component to manage the staff ‘s resistance to change. It is necessary to support a bidirectional and effective communication among the different organization levels (management, change leader, staff, and organizational departments). Then, the communication will be clear, effective and timely, bearing in mind why it is necessary to change and the benefits for the staff and the organization [19] [20]. Also, the communication gives opportunities for sharing best practices.

2.4 Staff Involvement

Staff involvement is a range of processes designed to engage the support, understanding and contribution of all employees in an organization and their commitment to its objectives [21]. Driving software process improvement from the bottom-up and promote the involvement of all affected parties actively: High level management and technical staff and maintain motivated all people involved [13]. While, staff participation is defined as a process of employee involvement designed to provide employees with the opportunity to influence and where appropriate, take part in decision making on matters which affect them. The need to generate a culture of process ownership is emphasized, as is the need to value software process improvement as real.

3 MEDEPRO Method

Process Deployment Method (MEDEPRO) is a method whose goal is to incorporate the critical success factors into a process deployment strategy. The method promotes effective and efficient use of deployed processes throughout the organization, incorporating infrastructure, training, communication, change management, adoption and motivation activities to use the processes, lessons learned, and metrics factors to evaluate the deployment process. Here, the focus is with how collectives use processes in their daily work, how to ensure processes get followed, and how to measure relevant aspects of the usage.

MEDEPRO is a method to deploy the processes focused on the critical success factors related to people. The process deployment method has five stages.

1. *Establish the infrastructure*: the purpose is to establish a process deployment team, secure the resources necessary to carry out the activities, and obtain the commitment at high levels of the organization.
2. *Motivate the processes’ use and adoption*: the purpose is to establish the process deployment plan that integrates communication, training, change management plan, metrics plan and planning.
3. *Manage deployment*: the purpose is to deploy the processes using the process deployment plan.
4. *Monitor deployment*: the purpose is to monitor and control effective process deployment using the proposed metrics.
5. *Feedback*: the purpose is to identify the lessons learned to be included in the PAL.

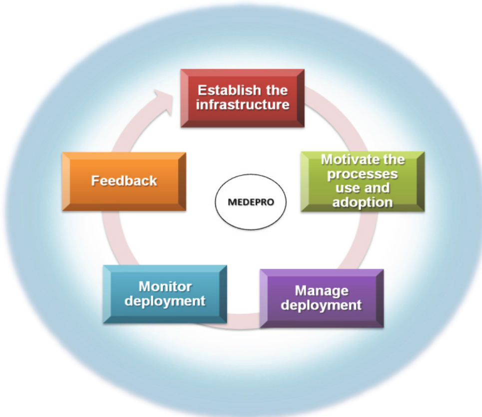


Fig. 2. Stages of the MEDEPRO method that integrates the critical success factors

Figure 2 shows the context and stages of the proposed model. These stages are described below.

3.1 Stage 1: Establish the Infrastructure

The purpose of this stage is to establish and provide the resources necessary to carry out the process deployment activities. This stage includes the following activities:

- Establish a high level group. This group is responsible for the resources allocation to carry out the process deployment plan. Their commitment is recognized by stakeholders.
- Establish a deployment group. This group is responsible for the development of the necessary activities for the process deployment. This group needs staff with skills in communication, training, change management, process models, and quality control.
- Review the processes: The processes that are included in PAL must be reviewed. These processes must be adapted to the organization's needs and must be aligned to the business goals and activities carried out in the organizations under consideration.
- Identify the barriers and risks to process deployment and thereby take corrective actions.
- Identify the owner of the processes to be deployed and review the previously deployed processes.
- Define the process deployment goals aligned to the business objectives. This requires a specific set of clear and feasible goals.

3.2 Stage 2: Motivate the Use and Adoption of the Processes

The purpose is to develop the Process Deployment Plan that integrates the training, communication, change management, and metrics process deployment (plans) among others. The first step is to deploy selected technical and management processes in an organization or project, and the second is to decide which deployment processes should be used to facilitate the adoption of the selected processes. This stage includes the following activities:

- *Determine the impact of the change in the organization* establishing the mechanisms and strategies to eliminate the barriers and identifying the necessary changes in the organization.
- *Develop the communication plan.* The communication plan includes: (1) define the objectives, (2) diagnose the current state of the organization in terms of communication mechanisms, channels, policies, and technology in order to take corrective measures, and (3) design and approve the communication plan.
- *Develop the training plan:* The training plan includes: (1) define the objectives, (2) diagnose the current state of training: how training is prioritized and carried out, (3) design the training plan: each unit develops and maintains a plan to satisfy its training needs, developing the knowledge, skills, and process abilities for the management competency, and (4) approve and communicate the training plan.
- *Develop the metrics plan:* A very important aspect in process deployment is to establish a set of metrics to evaluate the process of process deployment, for example the level of process acceptance and the use of processes by the staff. The plan includes: (1) objectives, (2) identify the metrics, (3) identify the necessary data to collect, (4) design the metrics database, (5) procedure to update the metrics, and (6) design the metrics plan.
- *Develop the management change plan.* The plan includes the following activities:
  - Establish the change vision.
  - Identify change agents: these are the people willing to support the changes and to invest time and effort into convincing their colleagues.
  - Diagnose the current organization's status in relation to the change in order to identify the potential barriers.
  - Analyze people's predisposition to the change in order to establish the strategies to motivate the staff to adopt the processes. We need to identify the early adopters who can assimilate the process quickly as these can serve as role models for the others. It is necessary to establish activities that motivate the staff.
  - Pace of change. It is considered essential that the change be gradual and accepted, not imposed. A flexible pace of change is necessary.
  - Manage the resistance to change and design the management change plan.
- *Process deployment plan:* it is necessary to integrate the partial results of the aforementioned steps. This plan integrates the training, communication, change management, and metrics process deployment plans and others.

3.3 Stage 3: Manage Deployment

The purpose of this stage is careful and quick deployment of the processes using the process deployment plan under the leadership of the deployment group. This stage includes the following activities:

- Deployment of the processes is synonymous with managing change in the organization. However, trying to changing behaviour, values and beliefs is a difficult, slow and tedious process. A strategy to manage change is necessary in order to prepare the staff to adapt to the changes.
- Develop pilot projects to reduce the risk of processes not being tailored to the organization's needs.
- Training the staff according to the needs of the organization to improve their skills, abilities and competences to carry out the activities.
- Communication at all levels.
- Involvement of the staff in the deployment activities.
- Maintain the staff motivated and committed.

3.4 Stage 4: Monitor Deployment

The purpose of this stage is effective process deployment monitoring and control. It establishes the procedures to carry out the adoption process and the efficiency of the different systems component. It is also associated with process engineering and the implementation results. This stage includes the following activities:

- Use the metrics established in the metrics plan.
- Evaluate the process deployment results taking into account how the processes are being used, the complexity of the processes, and the process adoption by the staff.
- Verify that the process deployment is complete, and that there is a relation between the activities carried out and the deployed process, and that the staff are comply with the functionalities established.
- Audit the activities to find possible problems in the adoption and institutionalization of the processes.

3.5 Stage 5: Feedback

The purpose of this stage is to include those improvements that accelerate the deployment process, the identified improvement to the processes, the improvements to the content of the deployment plan, and the lessons learned, all of which must be included in the PAL.

4 Case Study

The MEDEPRO experiment was conducted at five development and maintenance software centers in Spain and South America. A survey with open questions was carried out to the person in charge of process deployment at each center. Surveys can be an effective mean to assess the changes introduced in an improvement effort. The feedback provided by staff can therefore be used to improve the understanding of the effects caused by the introduced changes and to steer future improvements [6]. The survey consisted of three modules: module 1 was related to the organizational aspects, module 2 was related to the process deployed and module 3 was related to process deployment. Each center established its own procedures to implement the processes.

MEDEPRO established two measures, Process Use and Processes adoption, in order to evaluate the results of process deployment.

- P*rocess use* aims to analyze how the process was used. The answer types were: "*In Use*". "*Partial Use*". "*To Modify*" and "*Not Used*".
- *Process adoption* aims to analyze the process adopted by users. The answer types were: "*High*", "*Medium*", and "*Low*".

The measures were obtained from the five centers called C1, C2, C3, C4 and C5. To analyse the activities coverage in each center with respect to MEDEPRO activities, an evaluation method was developed. Statistical analysis techniques was applied to evaluate the effectiveness of software process deployment in terms of process use, process adoption, and staff resistance to change.

4.1 Process Deployment Evaluation at the Centers

In order to measure MEDEPRO's activities carried out at the centers in relation to critical success factors, an analysis was performed. Figure 3 shows MEDEPRO´s activities coverage obtained by center.

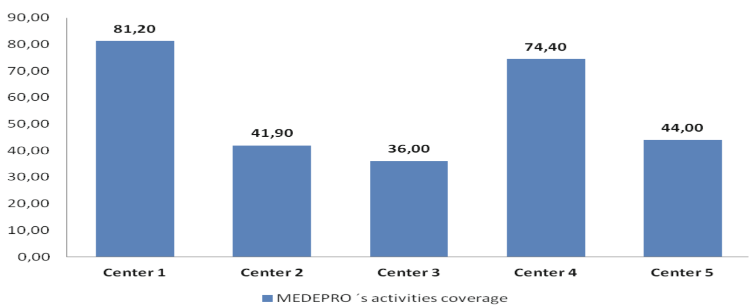


Fig. 3. MEDEPRO´s activities coverage by center

Figure 3 shows that Centers 1 and 4 achieved a MEDEPRO´s activities coverage greater than the other centers (81.20% and 74.40% respectively). The results of the survey show there are great differences among the centers depending on the process deployment strategy.

4.2 Processes Use

The purpose was to analyse the use of the processes deployed in software projects. Figure 5 shows that at Centers 1 and 4 the deployed processes percentage is greater than at the other Centers. At Center 1, the staff that used the processes participated in their definitions. Center 4 tailored the processes from Center 5 in order to get the CMMI certification. Besides, Centers 1 and 4 had a deployment strategy to decrease the resistance to change. At Center 5, although they had defined their processes, they did not establish actions to reduce the resistance to change. As Centers 2 and 3 did not participate in the processes definition, their use was low.

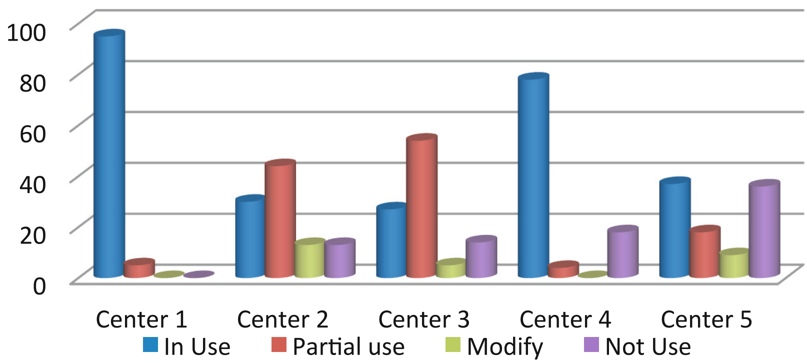


Fig. 4. Use of processes by category and center

4.3 Critical Success Factors

The following critical success factors were analyzed: training, communication, staff involvement, and change management.

- *Communication.* It is a main component to manage resistance to change. Effective communication among the different organization levels was established in a formal communication plan at Centers 1 and 4. Figure 5 shows that Center 1 and Center 4 present greater coverage of communications activities than the other centers.
- *Training:* it is provided to develop the skills and knowledge needed to use the processes deployed. Center 1 and Center 4 developed a formal training plan. Figure 6 shows that Center 1 and Center 4 present greater coverage of training activities than the other centers.

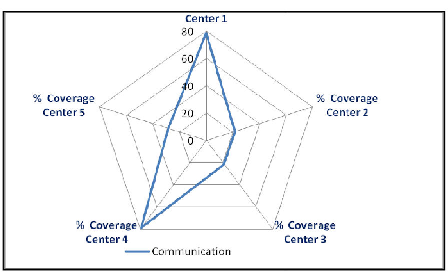


Fig. 5. Communication factor

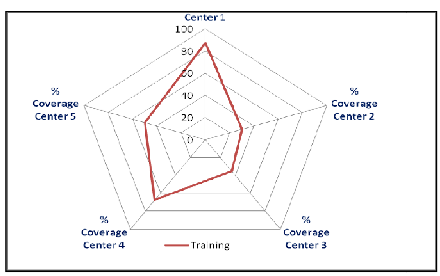


Fig. 6. Training factor

- *Staff involvement:* it is related to promoting the involvement of all affected parties: top level managers and technical staff. The Center 1, Center 3 and Center 4 promoted staff participation in process definition, so the staff was motivated to use the processes. Figure 7 shows that Center 1 and Center 4 present greater coverage of staff participation (activities) than the other centers.
- *Change management:* Process deployment is a change, so a strategy should be developed to motivate the staff towards achieving the objectives. An organization

that has the aptitude for managing change minimizes staff resistance. Figure 8 shows that Center 1 and Center 4 present greater coverage of staff participation (activities) than the other centers. In this case, Center 1 and Center 4 established a change management strategy to minimize the resistance to change. At the centers where staff participation was high, the resistance to use the process was low.

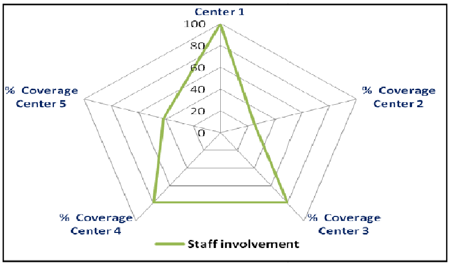


Fig. 7. Staff involvement factor

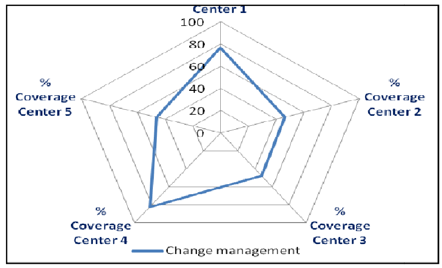


Fig. 8. Change management factor

From the foregoing, we conclude that the factors related to people have an impact on the acceptance of the processes.

5 Conclusions

MEDEPRO is an efficient process deployment method whose goal is to incorporate the critical success factors in a process deployment strategy. The method promotes effective and efficient use of deployed processes throughout the organization, incorporating activities related to infrastructure, training, communication, change management, adoption and motivation to use the processes, metrics to evaluate the deployment process and lessons learned for feedback and to improve the process. MEDEPRO was used to evaluate the process deployment experiences at five centers dedicated to software development and maintenance, and to know status of the processes deployed. The evaluation results show that the centers that achieved high values of activities coverage related to critical success factors such as: training, communication, staff involvement and change management, obtained better results than the other centers. However, the organizations analyzed do not present mechanisms to provide feedback to the deployment group in order to improve processes.

References

1. Chrissis, M., Konrad, M., Shrum, S.: CMMI for Development®: Guidelines for Process Integration and Product Improvement, 3rd edn. (2011)
2. Messnarz, R., Ekert, D., Reiner, M., O'Suilleabhain, G.: Human resources based improvement strategies-the learning factor. Software Process: Improvement and Practice 13(4), 355–362 (2008)

3. Niazi, M., Wilson, D., Zowghi, D.: A framework for assisting the design of effective software process improvement implementation strategies. Journal of Systems and Software 78, 204–222 (2005)
4. McDermid, J., Bennet, K.: Software Engineering research a critical appraisal. IEEE Proceedings on Software Engineering 146(4), 179–186 (1999)
5. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley (1998)
6. 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)
7. Garcıa, J., Amescua, A., Sanchez-Segura, M.-I., Bermon, L.: Design guidelines for software processes knowledge repository development. Inform. Software Technology 53(8), 834–850 (2011)
8. Cameron, Q.: Diagnosing and Changing Organizational Culture. Jossey-Bass (1999)
9. Forrester, E. (ed.): The International Process Research Consortium. A Process Research Framework. Software Engineering Institute (2006)
10. Bayona, S., Calvo, J., Cuevas, G., San Feliu, T., Sanchez, A.: Process deployment in a multi-site CMMI level 3 organization: a case study. In: Lee, R., Kim, H.K. (eds.) Computer and Software Science, pp. 147–156 (2008)
11. Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in software engineering, EBSE Technical Report EBSE-2007-01, Keele University (2007)
12. Bayona, S., Calvo-Manzano, J.A., San Feliu, T.: Critical Success Factors in Software Process Improvement: A Systematic Review. In: Mas, A., et al. (eds.) SPICE 2012. CCIS, vol. 290, pp. 1–12. Springer, Heidelberg (2012)
13. 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 EUROSPI 2011, pp. 7.27–7.37 (2011)
14. Kotter, J.: Leading Change. Harvard Business School Press (1996)
15. Prosci: Prosci's Change Management Methodology (2009), `http://www.change-managemente.com/change-managemen-process.htm`
16. Stelzer, D., Mellis, W.: Success factors of organizational change in software process improvement. Software Process: Improvement and Practice 4(4), 227–250 (1999)
17. Rogers, E.: Diffusion of Innovations. Glencoe Free Press (1962)
18. 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)
19. El Emam, K., Goldenson, D., McCurley, J., Herbsleb, J.: Modeling the likelihood of software process improvement. Empirical Software Engineering 6(3), 207–229 (2001)
20. Kaltio, K., Kinnula, A.: Deploying the defined software process. Software Process: Improvement and Practice 5(1), 65–83 (2000)
21. Hugh, S.: Implementing Best Practices in Human Resource Management, p. 231. CCH Canadian Limited (2003)

A Market Based Approach for Resolving Resource Constrained Task Allocation Problems in a Software Development Process

Murat Yilmaz[1,2] and Rory V. O'Connor[2,3]

[1] Lero Graduate School in Software Engineering, Dublin City University, Ireland
murat.yilmaz@computing.dcu.ie
[2] Dublin City University, Ireland
[3] Lero, The Irish Software Engineering Research Centre
roconnor@computing.dcu.ie

Abstract. We consider software development as an economic activity, where goods and services can be modeled as a resource constrained task allocation problem. This paper introduces a market based mechanism to overcome task allocation issues in a software development process. It proposes a mechanism with a prescribed set of rules, where valuation is based on the behaviors of stakeholders such as biding for a task. A bid process ensures that a stakeholder, who values the resource most, will have it allocated for a limited number of times. To observe the bidders behaviors, we initiate an approach incorporated with a process simulation model. Our preliminary results support the idea that our model is useful for optimizing the value based task allocations, creating a market value for the project assets, and for achieving proper allocation of project resources specifically on large scale software projects.

Keywords: Software Process Improvement, Game Theory, Process Simulation, Mechanism Design, Auction Mechanism, Task Allocations.

1 Introduction

Software development is an organized social setting, which should be equipped with economic methods for producing products in a multi-stakeholder viewpoint. While coping with uncertainties, activities of software development place precious *resources at risk* [1]. Conceptually, software development is also a form of economic activity, whereas its organizational structure should be considered as a social (decision-making) system based on several networks of interactions [2]. In this particular perspective, complexity of software development does stem from the complexity of human interactions and social communication costs [3], and therefore can be investigated as an organizational design problem.

Several empirical observations suggests that many different software projects not only fail due to technical reasons but also fall through organizational or team incompatibilities, and recently there is much interest in the social impacts of a

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 25–36, 2012.

software development process [4]. Furthermore, any intellectual process like software development should take into account that the knowledge used in software practices is tacit, dynamic and most importantly embedded in social relations [5].

Stellman [6] reported that productive team formation is a very vital component of management process. Most importantly however, the challenge here is to constitute a methodology by valuing resources with a decentralized *modus operandi*, which projects the burden of task planing onto the individuals responsible for carrying out specific tasks. To deal with self interested participants who can selfishly consume resources, the concept of mechanism design (MD) - a field of economic theory - has been found useful among community of researchers. For example, it founds an application in the field of computer science as *algorithmic game theory* [7]. While social choice theory claims that it is possible to merge participants' preferences into a single utility (i.e. preference) function, the goal of MD is to optimize these social choices based on the accumulation of individuals' preferences. MD constitutes a collective decision-making process with the assumption that participants will act rationally as defined in game theory. As a software project expands in its strategic settings, it becomes more convenient for management to induce collective decisions as a social choice function to reduce the decision to a single alternative, where several tasks are owned and operated by different parts of the software development organization.

The objective of this paper is to establish a novel approach for analyzing development task-resource allocation problems in the software development process by using a market based mechanism design approach. Our aim is to optimize the task-resource allocations based on the bids of the participants and decentralized market rules. The problem discussed here is constructed in two dimensions; firstly, as a theoretical model, which includes resource allocation rules and their symbolic representations. Secondly, by using simulated pseudo data for an initial test of our model, we develop a process simulation by exploiting kanban as a software development process.

The remainder of the paper is organized as follows. In section 2, we review the literature relating to the use of game theory in the area of software engineering. We identify several cases where MD is used to resolve resource allocation problems in information systems management. In section 3, we propose an auction-based market mechanism, which is constructed for addressing resource allocation issues in software project management. In section 4, we illustrate our model by using a virtual software project. Finally, in section 5, we draw our conclusions with respect to our implementation of the suggested market mechanism.

2 Game Theory in Software Engineering Literature

Several limited attempts have been made to understand software development as a cooperative or a competitive game form. For example, Lagesse [8] build a model based on a cooperative game theory approach with the idea of optimizing task assignment in software engineering efforts. On the other hand, Grechanik and Perry [9] focus on a game theoretic approach as a non-cooperative game, based on the fact that there are a number of potential goal conflicts among the roles of a software development approach. Moreover, Cockburn [10] consider software development as a series of games of invention and communication, where he portrayed the software development as "*economic-cooperative gaming*". His vision is similar to an iterative game in which two goals are competing for a resource. He also suggested that as an emerging area, which he called "mechanics and economics of communication" should be investigated in the near future. Based on the skills of the participants, Cockburn [11] also points out that software development should be considered as a game constrained upon its project resources. Using a approach based on grounded theory, Baskerville *et al.* [12] considered trade-offs and balancing decisions as balancing games that may appear in three different levels (i.e market, portfolio, management), where their nature is to progress dynamically with the demands of a market. Ko *et al.* [13] use a game theoretic approach for improving the reliability of data collected by using a method to improve its accuracy for better quantitative process management, where they also recommend a study for applying game theory in software project management and software process improvement activities. To improve the learning abilities of students Holeman [14] design a software process improvement game, which is a type of board game (designed to instruct CMMI to students) that participants compete for achieving CMMI level 2 on a Monopoly-like game board. Ogland [15] develops an approach for conflicting situations by using game theory and drama theory. He portrays software process improvement (SPI) as a game playable by quality auditors, software engineers, and managers. The goal is to identify how an SPI standard progress through an equilibrium (i.e. a proposed solution concept in a game).

Although game theory can be considered as a new and emerging field, there are a variety of related works outlined the importance of decision-making in software development landscapes. Equipped with the idea of "making everyone a winner", Theory W [16] is an approach based on the concept of risk management in software engineering decisions. To resolve the conflicts among the stakeholders of a project, it also suggests that the role of management somehow acts like a mediator or a negotiator, which seems likely similar to a game theoretic approach. In order to establish *a value based approach* and formalize the design goals of software development, Sullivan *et al.* [17] consider software design as an investment activity, where they applied the concept of real options to evaluate economic outcomes. To improve the effectiveness software architecture decision-making, Vajja and Prabhakar [18] investigate design issues based on the quality attributes, where they can be modeled as a game theoretical problem. Sazawal and Sudan [19] suggest a game model named, as a *basic software evaluation game* seems to be useful for helping software teams on decision-making particularly from an evolutionary perspective on software design decisions. Furthermore, they hypothesize that *lightweight* game theory is more useful for understanding software evolution. Bavota *et al.* [20] investigate the opportunities for using non-cooperative game theory for "extract class refactoring" in a situation such as two players that are competing to build new classes for improving the levels of cohesion.

There are some works in software engineering literature for the application of Prisoner's Dilemma (i.e. a non-cooperative game, based on two persons interactions). For example, Hazzan and Dubinsky [21] investigate the way of cooperation in extreme programming, in particular for pair programming practices. Secondly, a hidden game of Prisoner‘s Dilemma is investigated by Feijs [22] between a programmer and a tester. Thirdly, Oza [23] uses Prisoner‘s Dilemma framework to investigate strategic interactions in a client-vendor relationship in offshore outsourcing projects. Recently, Klein *at al.* [24] draws out attention to the notion of incentive conflicts in a software development both for identifying design characteristics and resource allocation perspectives. To bridge the gap between these conflicts of interests, they suggest that the notion of game theory particularly in terms of mechanism design should be useful for improving incentive compatible, decentralized and dynamic decision-makings in the software development processes.

2.1 Mechanism Design

The notion of MD is about understanding the structure of an organization such as a communication system for improving social decision-making and societal welfare. In MD, a social planner can create organizational structure to induce a planned or desired outcome based on the private information hold by the participant's of an organization. The information provided in this process is useful for modeling organizational procedures, solving economic problems such as allocation of resources, or dealing with problems related with asymmetric information and ultimately for supporting cooperation among the organization [25]. MD should also assist a social planner to model an organization for analyzing how the private information of individuals interacting throughout the organizational rules, which directly affect the expected outcomes. Such a model usually depends on the information of what is the possible action for each participant and their consequences that constitute the allocation decisions as a game theoretical solution.

Zhao *et al.* [26] propose an approach for understanding of Internet security issues as economical factors such as factors govern the actions and interdependence of the participants. To this purpose, they implement an economic mechanism (in this context, a certification mechanism) for reducing the security risks of users over the Internet. The essence of this mechanism depending on the idea to minimize the possibility of sending out malevolent traffic by increasing the responsibility of service providers and promoting the incentives to monitor the suppliers of malware and spam in their networks. The mechanism best works on a certified network concept by which each certified service provider will be able to use the collected information from other providers and held responsible for the traffic that is generated by their users [26].

Stef-Praun and Rego [27] outline a simple mechanism to transfer system wide efficient allocations of resources rather than individual resource allocations in a decentralized market for web services producers and consumers. Authors claim that the proposed mechanism can be realized to fit any structure composed of a large number of self-interested participants (e.g. a dynamic collaborative environment). Friedman and Parkes [28] investigate a customer pricing problem of a wireless networking provider, which may seen in a coffeehouse as a mechanism design problem. They develop a game theoretical model for bandwidth allocation based on a game of incomplete and asymmetric information.

In summary, these findings suggest that the mechanism design theory and its actual implementation for software organization can help for analyzing several economic interactions and designing organizations including markets and auction based market designs.

3 An Auction-Based Market Mechanism

A research focus of software process improvement is to allocate the project resources more efficiently, which is crucial for the software project's overall success. Similar to an economy, the process of software development consist of many independent parties (i.e. stakeholders) with autonomous (sometimes conflicting) objectives. These parties also have their private information (e.g. personal preferences), which should be revealed to improve the socioeconomic success of software development.

In an auction based MD, a market designer (e.g. economist, mediator, manager, etc.) is responsible for regulating the interactions of individuals, who promote social or economic objectives, for example; creating the right incentives for improving participants productivity. An example for such a mechanism is an auction where participants are defined as bidders that are bidding for the resources. These bids, however, may not value their requests truthfully. One way to deal such situations is to implement a *Vickrey auction* (i.e. a second price auction), which is based on a rule that each bidder submit a (sealed) bid and the valuation for a price is chosen as the second highest bid that is also paid by the winner. In theory, an aim is to maximize the efficiency of resource allocation by having a proper valuation.

3.1 Our Approach

Here, we formalize a software project as a market-based auction mechanism where the activities of the project is transformed into tasks. By using a role, *mediator*, these tasks are announced by an auction classification procedure. This rule set automates, which resources will be used for how long. Next, the auction system creates the auction and waits for the highest bidder.

Assume that a task is auctioned among n participants $i = 1, ..., n$, where valuation of an item by an individual is v_i for this item. The preference of a participant is given as a valuation function from an action set a, where $v_i : \mathcal{A} \rightarrow \Re$. Suppose that player i bids are in a vector $b = (b_1, b_2, ..., b_n)$.

The function which is used for delivering the task to a winning participant;

$$f_i(b_1, b_2, ..., b_n) = \begin{cases} 1 & \text{if } b_i > b_j, j = 1, ..., i-1, i+1, ..., n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The utility function of each bidder can be shown as;

$$u_i(b_1, b_2, ..., b_n) = x_i(b_1, b_2, ..., b_n)(v_i - p_i(b_1, b_2, ..., b_n)) \quad (2)$$

Consequently, the participant who values the item most is the winner from the set of participants $\mathcal{P} = \{i_{wins} | i \in I\}$ with the highest declared value and by following second price (Vickrey) auction, (see Listing 1.1 for a Mathematica routine to perform a second price auction). Please note that if there are tied values, a randomize function will be executed to resolve the issue.

Listing 1.1. Second Price Auction (adapted from [29])

```
SecondPrice[Bids_] :=
Module[
 {ii, Players, TiedValues, Winner, WinnerValue},
 Players = Table[ii, {ii, 1, Length[Bids]}];
 g[i_] := Bids[[i]] == Max[Bids];
 TiedValues = Select[Players, g];
 Winner = TiedValues[[Random[Integer,
      {1, Length[TiedValues]}]
     ]
    ];
 LowerSet = Complement[Players, {Winner}];
 WinnerValue = Max[Bids[[LowerSet]]];
 Return [{Winner, WinnerValue}]
 ]
```

3.2 Auction Basics

To realize the *true* economic value of a task, we propose to decompose every divisible parts of project tasks to form a set of auctions. There are two different roles interacting in this paradigm. Firstly, the people who are able to create auctions are called the *auctioneers*. For example, they can be an individual stakeholder or team of employees, who is authorized to construct a relationship between project resources versus potential software tasks. To reveal their true value, they create task based auctions in the market system. Secondly, the participants, who are interacting with the auction mechanism is called the *bidders* (e.g. software developers, testers, analysts, etc.). By using the auction mechanism, our system treats individuals or teams as entities that compete for allocation of project tasks.

In theory, we assume that bidders, who has best bidding plans seek to maximize the sum of their valuations. Furthermore, one important reason for creating such mechanism enables us to micromanage the idle tasks that are not utilized. However, this ability is not possible in many conventional approaches. Two of the auction features we suggest; (i) do not allow participants to enter a number consecutive bids for the same item and (ii) use a type of credit system similar to money or other incentives. Ultimately, this means that we enable auctioneers to create auctions on a time frame with the credits they can spend on auctioned tasks, which is scalable by allocating a suitable budget for required tasks and performance estimations (see Figure 1).

3.3 Rules to the Auction

- Our auction design aims to result in multiple rounds of concurrent bids for each task defined by the market.
- Auctioneer creates an auction, where initially the proposer should have determined the value proposition for every task. However, this value shall fluctuate with respect to the market requirements.
- In the first round, each bidder in the auction system makes a bid on one task that is auctioned. To remain in the auction bidders should keep their status active on the system (e.g. next round in auction).
- A bidder defined by the auction system shall be bidding on at least one task.
- An active bidder either currently holds the top bid on a particular task, or else raises the bid on a task of the bidder's choice by at least the minimum bid increment.
- A bidder who is in the possession of the top bid cannot raise or resign.
- Our auction ensures that the bids should be approved by the auction mechanism.

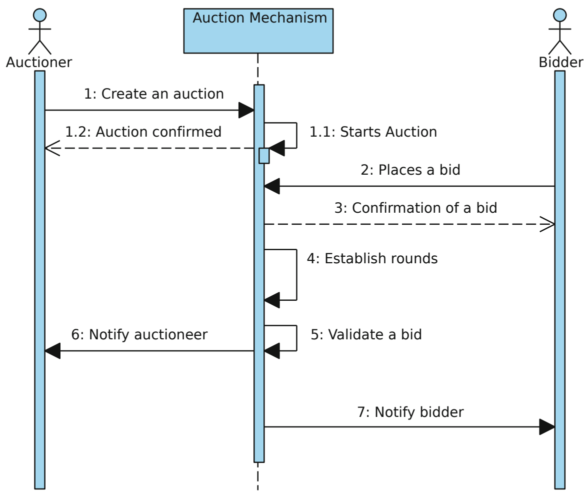


Fig. 1. The process in an auction model based on resource allocation

So as to observe the effect its operational efficiency for convergence of the costs of a resource to a value in a virtual software market, we propose a simulation of our auction based market mechanism, which is implemented in *Mathematica*.

4 A Demonstration of Our Approach

In this section, we illustrate a concrete example of how our approach is applied to a software development process. To simulate auction based market approach, we use the Monte Carlo technique. Consequently, we create a simulation model based on several hypothetical auctions, bidders and auctioneers that are interacting in a virtual (kanban) development process, and generate random variables and related events.

Kanban is a production planning approach that uses a pull system to manage the workflow. There are reasons to choose a kanban scheduling over other software processes. First, it enables us to define "*the software development process in terms of queues and control loops, and manage accordingly*" [30]. Therefore, we find it easy to integrate an market mechanism with kanban workflow. Secondly, as the task allocation process should be continuous in our settings, it is important for us to limit the work in process to the task winners, who should be able to pull the tasks on demand. Thirdly, both a market mechanism and kanban promote the idea of task transparency. Together, they can be used for effective management of the information flow in the software development process.

For our preliminary run, we start with 20 bidders (e.g. software developers), 400 tasks (e.g. coding a unit/function) and 10 auctioneers (e.g. stakeholders). For simplicity, we only assign developer role for all bidders. As soon as the auctions are created, they appear in the system demand pool, and further bidders start the bidding process. By executing the auction rules, in our model 400 tasks were auctioned to the participants. The kanban system identifies the necessary tasks for development, where the virtual market defines the auctioned items. In our approach, we make our calculations based on 120 virtual days of work for 400 auctions, all of which are integrated with the simulation of a virtual kanban workflow.

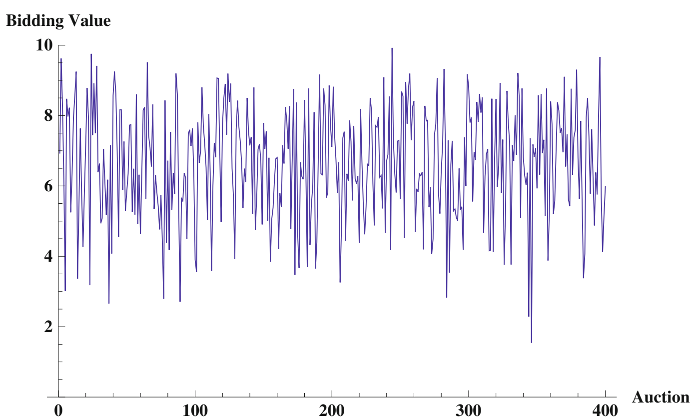


Fig. 2. Task Winners with Bidding Values in 400 Virtual Auctions

Figure 2 provides the distribution of tasks by using the winning bids for the tasks with respect to the number of virtual auctions. It also demonstrates resource consumption levels coupled with the tasks in our market control system.

Here, it is important to note that we use the range $[0-10]$ for the bidding values of tasks. In this scenario, the ability of market to mediate the resources in reply to business implications is not as significant as expected, particularly for the planned worked time schedule.

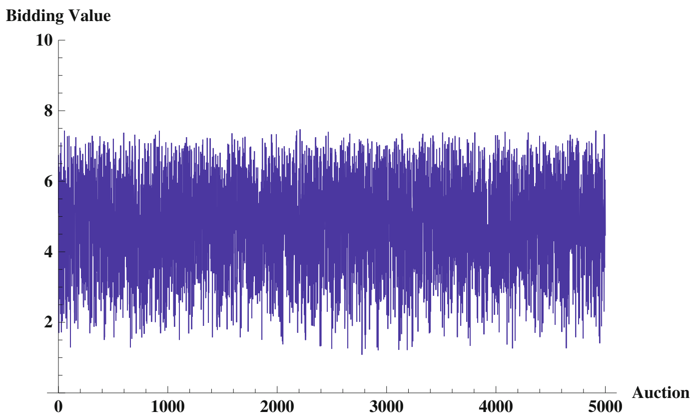


Fig. 3. Task Winners with Bidding Values in 5000 Virtual Auctions

In our second run, we create 5000 auctions for 300 tasks with 100 auctioneers, and 300 bidders. This iteration includes three different roles: (i) software developer, (ii) tester, and (iii) analyst, where we randomly generate different software teams limited to eight participants (up to 500 total). Therefore, in this scenario, our bidders are mostly considered as software teams. Figure 3 illustrates the results of the auctions versus task allocations with winning bids for a second price auction. In addition, it can be inferred from the figure that the health of our market economy relies on the ability of its development process which coordinates the flow of the bidding distribution.

5 Conclusion and Future Work

As the complexity of software development expands, an increasing amount of resources are required and consequently consumed during a software development process. One way to cope with this problem is through understanding the task allocations achieved during a software development life-cycle. For example, in the field of grid computing, there are a number of studies that propose models for allocating the resources scaled with resource consumers versus their providers in terms of the consumers needs (e.g. [31]). However, task based the resource allocation problem is not directly considered in a production model from a software management viewpoint, and further in light of a software methodology (i.e. kanban) to create a supply-demand chain that is simulated in an socio-economic landscape of software development.

This work proposes a mechanism for allocation of resources based the resource constrained task problems, particularly in cases either resource allocation is not possible or should be performed dynamically for economic reasons. Such a model can be useful to allocate software development tasks efficiently without the need of a (human) mediator. Based on the Monte Carlo method, we perform two different runs so as to assess the risk analysis on our two different virtual settings. The second iteration (i.e. a large scale run) demonstrates significantly better results.

Although we run a kanban process to control workflow and concurrent development of the distinctive features of a product. On the small scale, however, stabilization of workflow with respect to the auctioned tasks takes more time than expected. However, on a virtually large scale software development landscape - based on our kanban development process - the model confirms that it can be considered as a reasonable strategy for efficient allocation of resources based on our preliminary findings. From a social perspective of a software development process, first model does not allow participants to work together. However, second model enables participants to act cooperatively as a team. In our second hypothetical scenario, evidence suggests that a market based approach will act better than a static allocation technique. In addition, our method also shows that a kanban development process is useful to manage the auction mechanism.

From an industrial perspective, process simulation is an important asset for evaluating alternative scenarios. It is hard to observe which model or scenario works better than the others without simulating the process. We model an auction based market mechanism embedded in a kanban process to simulate virtual participants, auctions, and hypothetical events. By using the *Monte Carlo* method, we simulate our auction based market model with changing bidder roles that affects their behavior, and analyze the outcomes they produce. We argue that auctions processes can be utilized to improve the software development process, where bidder communications (collusion) can be organized to manage software process tasks and activities. In other words, our approach could be found as a convenient method to observe the interacting participants in terms of an auction based market mechanism, on a continuous scale, especially at a large scale software development settings.

Taken together, these results suggest that there is a common ground between auctions and software process improvement concerns, both of which are the processes that are dealing with the optimization of resource constrained task allocations. Finally, we conclude that there are still opportunities for resolving the issues of task allocation problems of the software development processes. In light of this, our proposed mechanism is expected to initiate new directions with its implication in the software process improvement community.

Acknowledgments. This work is supported, in part, by Science Foundation Ireland grant number 03/CE2/I303-1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

References

1. Selby, R.: Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research. Wiley-IEEE Computer Society Pr. (2007)
2. Yilmaz, M., O'Connor, R.V., Collins, J.: Improving Software Development Process through Economic Mechanism Design. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 177–188. Springer, Heidelberg (2010)
3. Dittrich, Y., Floyd, C., Klischewski, R.: Social thinking-software practice. The MIT Press (2002)
4. Xiong, J.: New Software Engineering Paradigm Based on Complexity Science: An Introduction to NSE. Springer (2011)
5. Ryan, S., O'Connor, R.V.: Development of a team measure for tacit knowledge in software development teams. Journal of Systems and Software 82, 229–240 (2009)
6. Stellman, A., Greene, J.: Applied software project management. O'Reilly Media (2005)
7. Nisan, N.: Algorithmic game theory. Cambridge Univ. Pr. (2007)
8. Lagesse, B.: A Game-Theoretical model for task assignment in project management. In: 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, pp. 678–680 (2006)
9. Grechanik, M., Perry, D.E.: Analyzing software development as a noncooperative game. In: IEE Seminar Digests, vol. 29 (2004)
10. Cockburn, A.: The end of software engineering and the start of economic-cooperative gaming. COMSIS 1, 1–32 (2004)
11. Cockburn, A.: Agile software development: the cooperative game. Addison-Wesley (2007)
12. Baskerville, R.L., Levine, L., Ramesh, B., Pries-Heje, J.: The high speed balancing game: How software companies cope with internet speed. Scandinavian Journal of Information Systems 16, 11–54 (2004)
13. Ko, S.P., Sung, H.K., Lee, K.W.: Study to secure reliability of measurement data through application of game theory. In: Proceedings of the 30th EUROMICRO Conference, pp. 380–386. IEEE Computer Society, Washington, DC (2004)
14. Holeman, R.: The software process improvement game. Software Engineering Education, 259–261 (1995)
15. Ogland, P.: The game of software process improvement: Some reflections on players, strategies and payoff. Norsk konferanse for organisasjoners bruk av informasjonsteknologi (NOKOBIT-16), 209–223 (2009)
16. Boehm, B., Ross, R.: Theory-W software project management principles and examples. IEEE Transactions on Software Engineering 15, 902–916 (1989)
17. Sullivan, K., Chalasani, P., Jha, S.: Software design decisions as real options. IEEE Transactions on Software Engineering (1997)
18. Vajja, K.K., Tv, P.: Quality attribute game: a game theory based techniquefor software architecture design. In: Proceeding of the 2nd Annual Conference on India Software Engineering Conference, pp. 133–134. ACM, Pune (2009)
19. Sazawal, V., Sudan, N.: Modeling Software Evolution with Game Theory. In: Wang, Q., Garousi, V., Madachy, R., Pfahl, D. (eds.) ICSP 2009. LNCS, vol. 5543, pp. 354–365. Springer, Heidelberg (2009)

20. Bavota, G., Oliveto, R., De Lucia, A., Antoniol, G., Gueheneuc, Y.: Playing with refactoring: Identifying extract class opportunities through game theory. In: 2010 IEEE International Conference on Software Maintenance (ICSM), pp. 1–5. IEEE (2010)
21. Hazzan, O., Dubinsky, Y.: Social Perspective of Software Development Methods: The Case of the Prisoner Dilemma and Extreme Programming. In: Baumeister, H., Marchesi, M., Holcombe, M. (eds.) XP 2005. LNCS, vol. 3556, pp. 74–81. Springer, Heidelberg (2005)
22. Feijs, L.: Prisoner dilemma in software testing. Computer Science Reports 1, 65–80 (2001)
23. Oza, N.V.: Game theory perspectives on client: vendor relationships in offshore software outsourcing, pp. 49–54 (2006)
24. Klein, M., Moreno, G., Parkes, D., Wallnau, K.: Designing for incentives: better information sharing for better software engineering. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, FoSER 2010, pp. 195–200. ACM (2010)
25. Hurwicz, L., Reiter, S.: Designing economic mechanisms. Cambridge Univ. Pr. (2006)
26. Zhao, X., Fang, F., Whinston, A.: An economic mechanism for better internet security. Decision Support Systems 45, 811–821 (2008)
27. Stef-Praun, T., Rego, V.: Ws-auction: Mechanism design for aweb services market. In: 26th IEEE International Conference on Distributed Computing Systems Workshops, ICDCS Workshops 2006, p. 41. IEEE (2006)
28. Friedman, E., Parkes, D.: Pricing wifi at starbucks: issues in online mechanism design. In: Proceedings of the 4th ACM Conference on Electronic Commerce, pp. 240–241. ACM (2003)
29. Varian, H.R.: Computational economics and finance: modeling and analysis with Mathematica, vol. 1. Telos Pr. (1996)
30. Shalloway, A., Beaver, G., Trott, J.: Lean-agile software development: achieving enterprise agility. Addison-Wesley Professional (2009)
31. Izakian, H., Abraham, A., Ladani, B.: An auction method for resource allocation in computational grids. Future Generation Computer Systems 26, 228–235 (2010)

Rule-Based Detection of Process Conformance Violations in Application Lifecycle Management

Rudolf Ramler[1], Hermann Lacheiner[1], and Albin Kern[2]

[1] Software Competence Center Hagenberg, Softwarepark 21, A-4232 Hagenberg, Austria
{rudolf.ramler,hermann.lacheiner}@scch.at
[2] Engel Austria GmbH, Ludwig-Engel-Straße 1, A-4311 Schwertberg, Austria
albin.kern@engel.at

Abstract. Software engineering processes are the basis for the development of quality software products within time and within budget. In this paper we present an approach to detect process conformance violations that reveal deviations between planned and executed software engineering processes. The approach is based on process rules that complement the process documentation. A framework for defining, executing and evaluating these rules has been implemented as extension to an Application Lifecycle Management (ALM) solution. The framework has been applied in context of introducing new processes and practices in an industrial environment. Over the timespan of more than a year, process conformance has been continuously evaluated as part of the nightly build. We were able to demonstrate that the results can be used to identify hot spots in process conformance calling for immediate action, to reveal long-term issues that motivate process improvement measures and, furthermore, that the continuous feedback provided by the approach has a positive impact on process quality. Finally, the paper documents useful lessons we have learned during the application of our approach in a real-world setting.

Keywords: Process Conformance, Process Quality, Application Lifecycle Management.

1 Introduction

Software engineering processes provide a basis for the development of quality software products within time and within budget. For this purpose, software engineering processes are aligned with the business and technical goals of an organization by defining best practices, guidelines as well as policies, rules, procedures and mechanisms concerning the involved roles and the different activities throughout the software life cycle. The benefit of applying a software engineering process is compromised, however, when the processes actually carried out deviate from the defined processes.

Reasons for such deviations, i.e. process conformance violations, are manifold. In this paper we present a rule-based approach to detect conformance violations applied in context of introducing new processes and practices in an industrial environment. The decision to introduce a new software engineering process raises the question

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 37–48, 2012.

whether the affected people are able to adopt and execute the defined process as initially planned. Checking for process conformance violations supports monitoring and managing the introduction phase as the progress will be made transparent and critical issues blocking the adaptation to the new process can be revealed. It is important to understand that detected conformance violations provides valuable feedback about problems in the initial process definition, ranging from ambiguous descriptions to unnoticed constraints and hidden requirements. Monitoring and analyzing conformance violations help to bring these issues to light.

Process conformance is an inherent issue in standards such as ISO 9001:2008 or ISO 12207 as well as in the area of process improvement and quality management research. Wil van der Aalst [1], for example, elaborates on conformance checking in context of process mining for business processes and workflow systems. Process mining has also been applied to study process enactment for software engineering processes by Huo et al. [2] and Da Cruz et al. [3]. A tool-supported approach has been described by Zazworka et al. [4], which has been used to study the adaptation of eXtreme Programming practices [5]. Furthermore, in Application Lifecycle Management (ALM, see [6] and [7]), solutions and tools have been proposed with the objective to support monitoring, controlling and managing software development lifecycle activities through: “1) enforcement of processes that span these activities; 2) management of relationships between development artifacts used or produced by these activities; and 3) reporting on progress of the development effort as a whole.” [7].

The approach described in this paper has been implemented as an extension to an ALM solution. ALM offers a useful technical platform for checking process conformance, which has been demonstrated in case of a large industry project. Section 2 provides a short introduction to ALM and the study context. The process rule framework is described in Section 3 and an overview of the implemented process rules is given in Section 4. Section 5 presents the results from applying the framework for more than a year in a real-world setting. The results are analyzed and discussed in Section 6. Finally, Section 7 concludes the paper with a summary and a list of lessons learned.

2 Background and Study Context

ALM is based on concepts from Product Lifecycle Management (PLM) [8] in manufacturing and service industries. The term ALM has been coined mainly by tool vendors to emphasize the move towards integrated tool suites covering activities that range from requirements engineering over design, implementation, integration, testing, deployment, to usage and maintenance. Despite the widespread use of the term ALM, a generally accepted definition has not yet evolved.

The key purpose of ALM is commonly acknowledged as the support for the development of software applications throughout all phases of development. However, when it comes to specific solution concepts, the broad scope of ALM is reflected in the different perspectives that contributed to the characterization of ALM. (a) Tool vendors and solution provider often have a strong focus on *support for engineering* *activities* – e.g., requirements engineering, issue tracking, configuration management, building – and the seamless integration of these activities. This focus can be explained by the history of many ALM tools and solutions, developed out of a heterogeneous set of specialized engineering and management tools. The ongoing evolution towards technically as well as functionally integrated tool suites is one of the major achievements in this area. (b) The advancement to integrated tool suites also leveraged the *support for management activities*. Typical examples are project planning, effort estimation, task and resource management, reporting and controlling. Hence, Pirklbauer et al. [9] emphasize the role of ALM for strengthening the business perspective in software engineering by closely integrating engineering and management activities. (c) The *support for software engineering processes* is the thread tying all aspects together. Process automation and process management capabilities are thus an integral part of ALM [7]. Many tools and solutions provide, for example, process templates that map the tool configuration to the building blocks of a software process model (e.g., [10] or [11]). In addition, several tools provide means to collect process-related metric data that can be integrated in the ALM tools' dashboards and reports.

The ALM solution we are referring to in this paper has been established for software development at Engel Austria GmbH. The company is the world's largest manufacturer of injection molding machines and one of the world's leading plastics processing machine manufacturers. Engel has more than 3.500 employees at eight production plants in Europe, North America as well as Asia and at subsidiaries and representatives in over 85 countries. Software development at Engel involves the development, customization and maintenance of large software systems for different product lines and, thereby, it has to cope with a broad range of applications spanning from visualizations at the HMI level to hardware level control of machinery and periphery equipment. Amplified by the different application levels as well as the distribution to separate development sites, a heterogeneous landscape of software technologies and an equally heterogeneous set of development tools and divergent software engineering processes have evolved. Although software development and all its activities is embedded in the context of mechanical and electrical engineering, the superordinate product development process provides only a rough guideline with considerable room for interpretation on how the activities, roles and deliverables of the software engineering process should be organized.

Engel Austria GmbH decided to introduce an ALM solution as a homogeneous tool platform in software development across the whole organization. This decision was accompanied by a process improvement initiative. Details about the process improvement and the tool introduction have been described in a previous paper [12], which shows how ALM has been used as infrastructure for evolving and improving the software engineering process. Thereby, monitoring and management of process conformance have turned out as a key factor for the successful introduction and roll-out of the ALM solution as well as the corresponding software engineering process. These activities were supported by a framework for the rule-based detection of process conformance violations on an operational level, which we implemented as an extension to the ALM solution.

3 Framework for Detecting Conformance Violations

This section provides an overview of the framework developed for detecting process conformance violations embedded in the ALM solution. The framework is able to detect violations in the process at the operative level that manifest themselves in the contents, characteristics or properties of produced work items (e.g., tasks, ideas, requirements or test cases) and documents (e.g., specifications, reports or source code).

The work items are stored as data records in the ALM tool's database; the documents are stored in the tool's repository. For both, work items and documents, the ALM tool maintains a rich set of meta-information including properties such as author, owner, status, priorities, flags, hierarchical structures, dependencies and trace links as well as the complete change history. All this data is accessible via the ALM tool's Java API. Violations of the defined software engineering process frequently result in traces and characteristic patterns visible in this data. Hence, the framework queries the data stored in the ALM solution and uses process rules to check for suspicious traces and patterns of violations.

The typical workflow for detecting process conformance violations includes following steps: (1) identification of patterns indicating conformance violations, (2) implementing process rules to checks for these patterns, (3) adding the rule to the rule base of the framework, (4) the framework executes all rules as part of the nightly build that is also used for building and testing the software system, (5) the results are generated into a process quality report part of the ALM tool's quality dashboard. Finally, (6) the engineers are triggered to correct immediate faults and deficiencies that resulted from the process violations, while (7) the process manager monitors the violation trend over time and adjusts and extends the rule set if necessary.

Fig. 1 provides an overview of the framework for detecting process conformance violations; the core components constituting are described in the following.

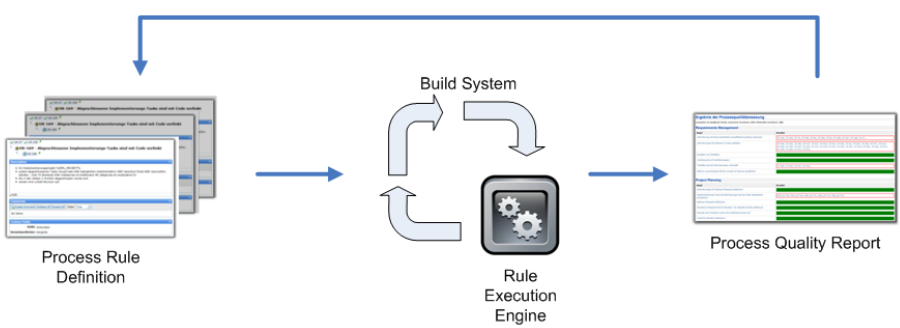


Fig. 1. Framework for detecting process conformance violations

3.1 Process Rule Definition and Rule Base

The process rules are strongly tied to guidelines and policies in the process documentation. The process documentation is also maintained in the ALM tool and includes the whole chain from process goals and requirements, over workflow descriptions, to how-to documents, guidelines and policies [12]. In fact, by linking a rule to a guideline or policy, it also becomes a part of the process documentation and can easily be traced to the corresponding process requirement.

An example is the policy that *"a task has to be defined for any accountable project work"*, which has been established in order to satisfy traceability requirements of the software engineering process. The tasks is managed with the ALM tool including all relevant properties such as description, priority, owner, assignee, responsible QA, status, estimated effort etc. Furthermore, in software development, the work has to be associated to the implementation of a specified software requirement and a planned development iteration. Some of these properties are mandatory (e.g., defining the owner of the task) and their use is enforced by the ALM tool. Other aspects are handled more flexible (e.g., the association of a task to a requirement or an iteration) to keep the administrative overhead in task handling low. Thus, process rules such as *"tasks have to be linked to a requirement"* or *"completed iterations must not have open tasks associated"* are used to ensure that the aspects involved in managing a task are accomplished according to the guidelines and policies at the end of the day.

Even though process rules are specified in a textual form, they are also executable. Therefore each rule is linked to a JavaScript function that queries the repository of the ALM tool in order to detect violations. A library of reusable functions implementing various generic queries has been developed. Thresholds and other relevant properties are embedded as variables in the textual description of the rules. The rule engine parses the description, extracts the variables and executes the query function that returns a list of items violating the rule. For example, the description of the rule mentioned above is presented in Fig. 2. The text contains several variables surrounded by the character %. These variables are set as parameters of the query function *checkLinkCardinality(projectId, fromItem, toItem, minLinks, maxLinks).*

In project %PROJECT_ID% every item of type %task% has to be linked to one item of type %req%. There has to be at least %1% and at most %1% link between a pair of these items.

Fig. 2. Example of a process rule

Using the ALM solution as technical infrastructure and environment for process management has the advantage that all features provided by the ALM tool can also be utilized for the development and maintenance of the software engineering process [12]. Thus, the process rules in the rule base are handled like any other work item stored in the ALM tool. Traceability between rules and other items in the process documentation can be applied as well as the ALM tool's change management process, the versioning system and the report generator.

3.2 Rule Execution Engine and Nightly Builds

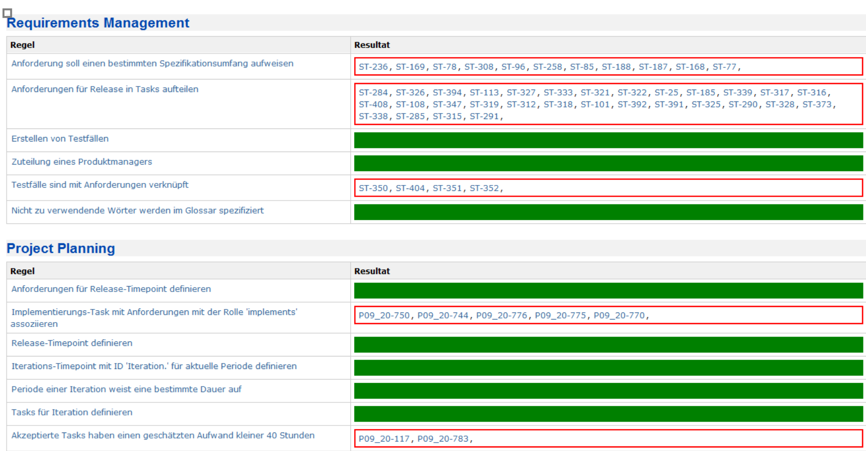
The key mechanism for running the process rules is the rule execution engine. The rule engine parses the description, extracts the variables and calls the underlying

JavaScript function. Furthermore, the engine wraps each process rule in a JUnit test case. With the wrapper in place, process rules can be treated like the JUnit tests used in software development. Thus, for example, they can easily be included in the nightly build used for software development projects.

The ALM solution uses Jenkins (http://jenkins-ci.org) as build system. It allows scheduling periodical build executions, logging executions and generating a results file in XML format, as well as visualizing the aggregated results as trend graphs – features, which are normally used to depict the test results in software development. The build system also offers automated notification on critical failures and the detailed analysis of the results. However, for that purpose we produce a customized process quality report.

3.3 Process Quality Report

The results from all executed process rules are compiled to a process quality report published as Wiki page (Fig. 3). Every project member has access to this report. The report is personalized as it contains only rule violations for items the user is responsible for. On the left side of the report the executed process rules are listed and on the right side the results are shown. The visualization of the process rule result is similar to a JUnit test report: a green bar indicates that no violation was found, whereas a red border denotes a detected violation. The items violating a rule are listed as links, so the user can open an item with one click and apply corrective actions.


Requirements Management

Regel	Resultat
Anforderung soll einen bestimmten Spezifikationsumfang aufweisen	ST-236, ST-169, ST-78, ST-308, ST-96, ST-258, ST-85, ST-188, ST-187, ST-168, ST-77,
Anforderungen für Release in Tasks aufteilen	ST-284, ST-326, ST-394, ST-113, ST-327, ST-333, ST-321, ST-322, ST-25, ST-185, ST-339, ST-317, ST-316, ST-408, ST-108, ST-347, ST-319, ST-312, ST-318, ST-101, ST-392, ST-391, ST-325, ST-290, ST-328, ST-373, ST-338, ST-285, ST-315, ST-291,
Erstellen von Testfällen	
Zuteilung eines Produktmanagers	
Testfälle sind mit Anforderungen verknüpft	ST-350, ST-404, ST-351, ST-352,
Nicht zu verwendende Wörter werden im Glossar spezifiziert	

Project Planning

Regel	Resultat
Anforderungen für Release-Timepoint definieren	
Implementierungs-Task mit Anforderungen mit der Rolle 'implements' assoziieren	P09_20-750, P09_20-744, P09_20-776, P09_20-775, P09_20-770,
Release-Timepoint definieren	
Iterations-Timepoint mit ID 'Iteration.' für aktuelle Periode definieren	
Periode einer Iteration weist eine bestimmte Dauer auf	
Tasks für Iteration definieren	
Akzeptierte Tasks haben einen geschätzten Aufwand kleiner 40 Stunden	P09_20-117, P09_20-783,

Fig. 3. Excerpt of the original process quality report

4 Development and Evolution of Process Rules

A process rule undergoes different states: Initially, when the process rule is defined but not yet implemented, it is in the state *Draft*. After the rule has been implemented it needs to be reviewed by the process manager. So the state is set to *Review*. Once accepted by the process manager, the state is set to *Active* and the rule is evaluated by the process execution engine in the nightly build. If a process rule becomes obsolete, it is kept in the rule base but its state is set to *Inactive*.

Currently the rule base contains 43 active process rules, which are linked to 82 different elements in the process documentation and 6 different CMMI level 2 core process areas. Table 1 provides an overview of the covered process areas and the number of associated rules. The majority of the process rules relate to the areas *Project Planning, Requirements Management* and *Project Monitoring and Control.*

Table 1. Coverage of process areas by process rules, sorted by the number of rules

Process Area	Area Description	Rules
Project Planning	establish and maintain plans that define project activities	14
Requirements Management	manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products	10
Project Monitoring and Control	provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan	8
Process and Product Quality Assurance	provide staff and management with objective insight into processes and associated work products	6
Measurement and Analysis	develop and sustain a measurement capability used to support management information needs	3
Configuration Management	establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits	2

Most of the process rules (32 rules) have been developed in an initial pilot phase, when the idea of supplementing process management with automated conformance checks was prototypically implemented with dynamic Wiki pages. Based on this first prototype the described framework has been developed and the rule base continuously evolved over more than a year (15 months) to its current size (43 rules).

Over time, however, not only the number of rules increased, but existing rules had to be adapted and refined. In total, 39.5% of the rules (17 rules) have been adjusted; 11.6% (5 rules) two or more times. One rule has been changed four times. This particular rule deals with *associating test cases to requirements*. Its change history reveals typical reasons for changes: (1) refine the association between requirement and test case, (2) exclude hardware requirements, (3) exclude requirements in draft status, and (4) increase the threshold for the number of test cases to be associated with a requirement. However, 60.4% (26 rules) did not change since their introduction.

5 Results

The process conformance is checked in every nightly build in addition to building and testing the software product. In total the rules have been executed 274 times over the studied period of one year. The results are shown in Fig. 4. The chronology of executions is depicted on the x-axis. The y-axis shows the number of process rules per execution, which increased from 32 to 43 throughout the year. The red area represents the failed rules indicating a process violation, while the blue area corresponds to the rules that executed successfully. The number of failed rules ranges from a minimum of 15.6% (5 out of 32 rules) to a maximum of 50.0% (18 out of 36 rules). On average 33.2% of the rules failed (12.51 out of 37.62; standard deviation s=3, median m=13).

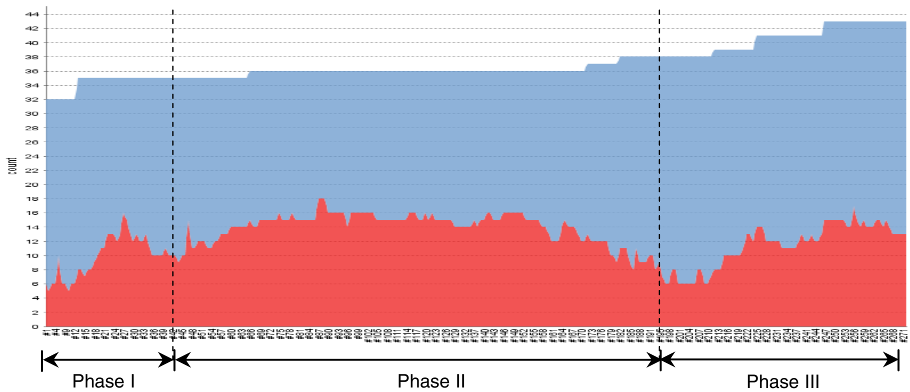


Fig. 4. Results from executing the process rules for one year

The studied timespan includes three distinguished phases. *Phase I* is characterized by the introduction of additional process rules, which resulted in an increase of process violations and failing rules with a maximum at 16. The phase starts from the beginning (6 failing rules) and ends with execution #44 (9 failing rules). *Phase II* starts with execution #45 (10 failing rules) and ends with execution #198 (6 failing rules). In this phase a continuous growth of the number of violated rules can be observed, which stayed for a considerable time on this level – the maximum were 18 failing rules – until initiated process improvement measures became effective. *Phase III* starts with execution #199 (8 failing rules) and is ongoing. The current snapshot is execution #274 (12 failing rules). The focus of this phase has been on improving overall process conformance by extending the rule base. Again an increase in the number of failing rules is observable with, up to now, a maximum of 17 failing rules.

A failing rule is associated with one or more work items that are affected by the checked process violation. In the last execution, for example, a total of 761 different work items were analyzed of which 8.8% were affected by a violation. Besides work items, documents and additional artifacts of software engineering are included in the checks, for example, requirements specifications, builds, source code or time points for iterations and releases.

6 Analysis and Discussion

The results from executing the process rules are presented in a personalized process quality report. In addition they are analyzed, first, to identify hot spots in process conformance that require immediate fixing and, second, to reveal long-term issues that have to be addressed by process improvement measures. Third, the effect of checking for process conformance and initiating improvements is evaluated by the process manager. This analysis is presented and discussed in the following.

6.1 Hot Spots in Process Conformance

Hot spots in process conformance are characterized by violated process rules that result in a high number of defective work items. The analysis of the current results is depicted as Pareto chart in Fig. 5. It shows the process rules on the x-axis sorted by the associated defective work items from left to right. The bars indicate the number of defective work items per rule; the curve represents their cumulated number that amounts to a total of 72 defective work items. As indicated in the figure, 80% of all defective work items are caused by 5 violated rules, i.e., about 11% of all rules.

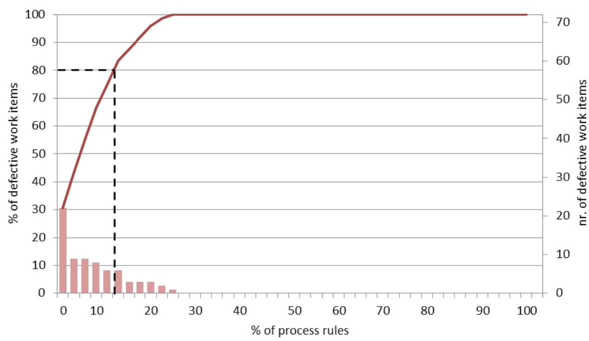


Fig. 5. Defective work items in relation to process rules

6.2 Frequently Violated Process Rules

Another key indicator is the number of times a process rule has been violated in the analyzed period. This analysis allows identifying the most critical process rules over time.

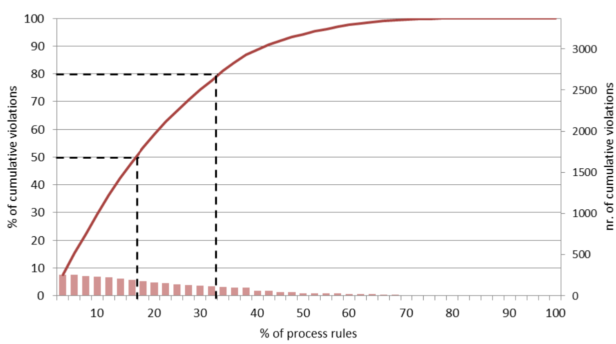


Fig. 6. Cumulative number of violations in relation to process rules

Fig. 6 shows the Pareto chart that relates the rules to their accumulated number of violations. The rules are sorted in descending order from left to right. The bars indicate the accumulated number of violations per rule; the curve represents the running sum over all rules with a total of 3,371 violations for the whole analyzed period. In this analysis 80% of the total number of violations is caused by 14 different rules that together represent about one third (32.6%) of all rules. Half of the violations (50%) can be traced back to 7 rules, i.e., 16.3% of all rules.

6.3 Impact on Process Quality

Various process and product related metrics are monitored in the ALM solution. Based on these measures the effect of process conformance checks – in particular of introducing new process rules and related process improvement measures – is evaluated. The following example illustrates the impact of establishing new rules for project management on the duration of development tasks, as represented by the metric *task duration.* This metric measures the time span from creating to resolving a task.

Fig. 7 shows the task durations in chronological order. Two different phases can be distinguished. The first phase is dominated by highly fluctuating task durations. On average a task was open for 63.8 days (standard deviation *s*=71.3); the maximum representing an extreme case was 299 days. The introduction of the new process rules is marked by the dashed vertical line. After that, in the second phase, task durations were reduced to an average of 18.5 days (*s*=17.0, *max*=71 days).

The new process rules clearly had a positive impact on project planning and project control. The rules encouraged a higher accuracy in the handover of tasks and maintaining status flags. As a consequence, the quality of measurement data improved as well as the transparency and traceability of the overall development status.

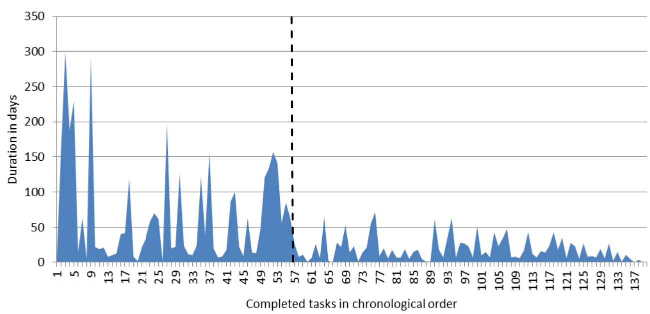


Fig. 7. Duration of the tasks completed in the analyzed period

7 Summary and Lessons Learned

In this paper we presented an approach to detect conformance violations between a planned and an executed software engineering process. The approach is based on process rules that complement the process documentation. A framework for defining, executing and evaluating these rules has been implemented as extension of an ALM solution. It executes the rules as part of the nightly build to identify violation patterns in the data stored in the ALM tool's repository.

Further, we described and discussed the application of this approach in context of the introduction of new processes and practices in an industrial environment. In the analyzed timespan of more than a year, a total of 43 process rules have been defined and implemented. Their continuous evaluation as part of the nightly build showed an average of 12.51 failing rules indicating violations of the defined process. We were able to demonstrate that these results can be used to identify hot spots in process conformance calling for immediate action, to reveal long-term issues that require process improvement measures and, furthermore, that the continuous feedback provided by the approach has a positive, sustaining impact on process quality.

Last, we have learned several useful lessons during the application of our approach in a real-world setting.

- Process rules have either been defined top-down together with the introduction of new guidelines and policies for software development, or they emerged bottom-up in response to problems compromising process conformance. We found that there is no "best way" for defining process rules; equally valuable rules have emerged out of both sources.
- The rules have to be adapted and maintained over time. Typical reasons for necessary changes are extending or narrowing the set of work items covered by a rule, refining the conditions under which a rule triggers, or adjusting the associated thresholds. Changes were required for about 40% of the rules in our rule base.
- The process definition has to be adapted and maintained too. The trend analysis of violations revealed three characteristic patterns: (a) rules that, once introduced, led to a decrease in the number of violations to zero, i.e., the underlying issue had been cleared; (b) cleared rules with periodical peaks of violations showing up again due to changing context factors; and (c) rules with a constantly decreasing number of violations that, however, never actually reached the zero line. An investigation of the last type revealed cases where practical constraints hindered the execution of the process in conformance to its definition. For these cases the solution was an adaptation of the process definition and the associated rules.
- Observing the results for over a year showed that mechanisms providing constant feedback are essential to keep the users' attention. The status of the process conformance is made transparent in a personalized quality report. However, further measures for "pushing" the information about violations to the concerned users are required. These measures range from (1) color-coding affected work items over (2) refusing status transitions in the development workflow to (3) triggering eXtreme Feedback Devices [13] such as lava lamps, traffic lights or audio devices physically signaling the process conformance status to the development team.
- Nevertheless, much of the success of the approach has to be attributed to the fact that for almost all violations corrective actions can directly be initiated by the users themselves. While all conformance violations are recorded and included in a

subsequent analysis by the process manager to identify improvement measures on the long run, the foremost idea of the approach is to encourage users to instantly resolve rule violations. Therefore, for example, the process quality report generates an up-to-date picture of the actual rule violations and guides the responsible users to the affected work items for applying quick fixes. The short feedback cycle motivates immediate actions and emphasizes the positive effects on process quality.

References

1. van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Huo, M., Zhang, H., Jeffery, R.: An Exploratory Study of Process Enactment as Input to Software Process Improvement. In: 4th International Workshop on Software Quality, WoSQ. ACM (2006)
3. Da Cruz, J.I.B., Ruiz, D.D.: Conformance Analysis on Software Development: An Experience with Process Mining. International Journal of Business Process Integration and Management 5(2), 109–120 (2011)
4. Zazworka, N., Basili, V.R., Shull, F.: Tool Supported Detection and Judgment of Nonconformance in Process Execution. In: 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM. ACM (2009)
5. Zazworka, N., Stapel, K., Knauss, E., Shull, F., Basili, V.R., Schneider, K.: Are Developers Complying with the Process: An XP Study. In: 4th International Symposium on Empirical Software Engineering and Measurement, ESEM. ACM (2010)
6. Kääriäinen, J., Välimäki, A.: Applying Application Lifecycle Management for the Development of Complex Systems: Experiences from the Automation Industry. In: O'Connor, R.V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 149–160. Springer, Heidelberg (2009)
7. Schwaber, C.: The Changing Face Of Application Life-Cycle Management. Technical Report, Forrester Research Inc. (2006)
8. Saaksvuori, A., Immonen, A.: Product Lifecycle Management, 3rd edn. Springer (2008)
9. Pirklbauer, G., Zeilinger, R., Ramler, R.: An Integration-Oriented Model for Application Lifecycle Management. In: Filipe, J., Cordeiro, J. (eds.) 11th International Conference on Enterprise Information Systems. LNBIP, vol. 24, pp. 399–403. Springer, Heidelberg (2009)
10. Gousset, M., Krishnamoorthy, A., Keller, B., Timm, S.: Professional Application Lifecycle Management with Visual Studio 2010: with Team Foundation Server 2010. Wiley (2010)
11. Krishna, S., Fenstermaker, T.: IBM Rational Team Concert 2 Essentials. Packt Publishing (2011)
12. Lacheiner, H., Ramler, R.: Application Lifecycle Management as Infrastructure for Software Process Improvement and Evolution: Experience and Insights from Industry. In: 37th EUROMICRO Conference on Software Engineering and Advanced Applications, Euromicro SEAA. IEEE (2011)
13. Clark, M.: Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Apps. The Pragmatic Programmers (2004)

A Model for Business Value in Large-Scale Agile and Lean Software Development

Jeanette Heidenberg[1,2], Max Weijola[1,2], Kirsi Mikkonen[3], and Ivan Porres[1,2]

[1] Åbo Akademi University, Department of Information Technologies,
Joukahaisenkatu 3-5 A, 20520 Turku, Finland
Givenname.Surename@abo.fi

[2] Turku Centre for Computer Science, Joukahaisenkatu 3-5 B, 20520 Turku, Finland

[3] OY L M Ericsson Ab, Hirsalantie 11, 02420 Jorvas, Finland
Givenname.Surename@ericsson.com

Abstract. As agile and lean methods continue to increase in popularity and move away from their home ground – small, co-located teams with an actively involved customer – they are faced with new challenges. One such challenge is the definition and communication of business value in large settings, where multiple development teams interact with multiple business stakeholders. This challenge is exacerbated by the fact that the term business value in agile contexts is not clearly defined, even though the creation of business value is one of the central themes in agile and lean development. In this paper, we propose a model for business value that is intended to make explicit different factors that constitute the concept of business value in agile and lean software development. This model has been jointly developed with industrial partners in the Cloud Software Finland research project. We aim to further evaluate and develop the model in the future within this research project.

Keywords: Business Value, Agile, Lean, Feature Prioritization, Large-Scale Agile.

1 Introduction

Agile and lean methods continue to increase in popularity in the field of software development [2]. One of the key concepts in agile and lean development methods is the creation of value for the customer. This value can be saved costs, increased revenue or some other form of added value. This is how the term business value is often used, both in literature and by the agile community. The definition of the term business value can, however, be rather ambiguous [17].

Despite the lack of a clear definition of business value, it is still used as a key factor in requirement prioritization in agile and lean contexts. Fundamental agile practices such as backlog prioritization and the planning game [4] depend on selecting the features with the highest business value. In such practices the business representative is expected to prioritize backlog features based on business value. How the business representative arrives at a specific conclusion regarding

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 49–60, 2012.

the business value of a certain feature is usually not explained, and he or she is expected to estimate this value based on previous experience or an intuitive feeling.

The simple view of business value is to think of it as the revenue or other monetary benefit expected from the features to be implemented. In reality, however, this is too narrow a definition, as exemplified by case studies performed by Racheva et al. [18]. Practitioners ask themselves not only what the feature is worth in dollars, but also questions like what the development organization will gain from implementing the feature, how the customer relationship will benefit from the feature, etc.

The use of business value in feature prioritization might be well manageable in traditional agile settings with the support of a customer on site. In this article we focus on large organizations that develop advanced software systems, often in the embedded systems domain. In these large-scale settings, often with multiple development teams and multiple product owners, the customer relationship is complex with either multiple customers or no possibility of direct customer interaction.

The research question that we attempt to answer is how to represent business value in a potentially large-scale agile context so as to support communication and understanding between all stakeholders. Our research method stems from design science [10] and the model proposed in this paper has been iteratively developed in a series of workshops with industry experts. We have also benefited from input from the academic world through literature research as well as the Finnish software research community in the Cloud Software Finland project. One of the goals of the Cloud project is to support Finnish software industry in transforming their operations with the help of agile and lean methods.

The proposed model for business value is intended to make explicit the different factors that constitute the concept of business value. The intended use of the model is two-fold: (i) To support the business representatives, such as the product owners, in creating a partial order that constitutes the prioritized list of features to be developed; and (ii) to communicate the business value of the features under development with the software development teams. We also outline a visualization aid that supports these usage areas.

The paper is structured as follows. In Section 2 we give a brief overview of earlier work and discuss the role business value has in agile practices. We continue by introducing our proposal for a model of business value in large-scale agile contexts in Section 3. In Section 4 we outline the intended use of the model. In Section 5 we discuss the model and our future work. Our conclusions are presented in Section 6.

2 Background and Related Work

The definition of business value varies to a notable extent in literature. In Section 2.1 we present some definitions of the term business value in the literature. We also state our definition of business value to clarify the usage of the term in this paper. In Section 2.2 we give an overview of how the term business value is used in some of the more common agile planning practices.

2.1 Definitions of Business Value

Despite the fact that the term business value is a central concept in software development, its definition is usually vague, ambiguous or non-existent. This is maybe best illustrated by the literature study [17] and interviews with practitioners [18] performed by Racheva et al. 2009-2010.

Racheva et al. note in their literature study [17] that of all the scientific papers reviewed only five papers include a definition of the term business value; in the rest of the papers the term is a "self-evident concept." When interviewing practitioners [18] they note that the definition of the term business value may even vary within the same organization depending on the structural differences between clients and projects

Business value is often used as a tool for determining whether a feature or product is going to be profitable in comparison to the investment. Will the organization profit from implementing the feature? Examples of this definition can be found in Patton [13]: "Business Value is something that delivers profit to the organization paying for the software ..." and Rawsthorne [19]: "business value is what management is willing to pay for." Another popular use of this definition of business value is the Business Model Canvas [12]. While this is an important aspect to business value, it is not the one we focus on in this paper.

Another, contrasting definition of business value is the one given by Pettit [14], who states that business value should be used as a "communication vehicle, a means by which the business-IT partnership can be strengthened." Business value as a means of communication is the definition we find important for the agile development context.

In agile contexts, the business value of the individual features (or user stories, requirements, product backlog items) is stated by the customer or a representative of the customer, such as the product owner in Scrum [20]. Different agile methods have different practices for using this input in their prioritization practices (see Section 2.2), but as has been demonstrated by Racheva et al. [16] these different practices can be abstracted into one overall process model of how the prioritization is done.

What is not clearly defined, however, is how the customer or product owner decides the business value of the feature. This is a challenge especially in contexts where there are multiple customers with different priorities or large products with multiple product owners. This problem is illustrated in Figure 1. The product owner(s) will make a business decision on which features are eligible for implementation. These features then need to be prioritized before negotiating scope with the development team. This step is usually not described in the literature. This is the step we attempt to support with a more tangible method than intuition or experience alone.

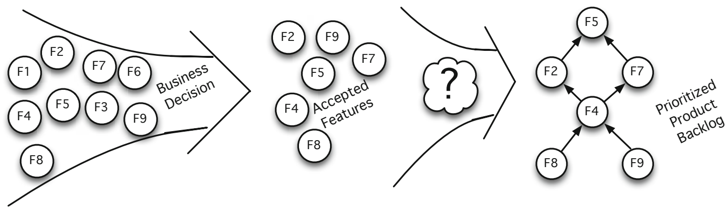


Fig. 1. Prioritizing features for the product backlog

2.2 Business Value in Agile Planning Practices

Agile software development methodologies started growing popular in the late 1990's and early 2000's as lighter alternatives to traditional heavyweight plan-driven development methodologies. Extreme Programming by Kent Beck [4] is one of the earliest, and still today a popular agile development methodology. Scrum [20] is a popular project management framework introduced by Jeff Sutherland and Ken Schwaber in the early 2000's. Scrum and Extreme Programming are often used together, as Scrum in itself does not concern itself with implementation practices.

Already from the start, the concept of business value has played an important role in the agile movement. It is even present in the first of the twelve principles of the agile manifesto, stating that the "highest priority is to satisfy the customer through early and continuous delivery of valuable software." However, the manifesto does not define what constitutes "valuable software." The knowledge of what is valuable is assumed to be known to the business stakeholder and communicated to the team in order to prioritize features and negotiate scope. In the sections below we briefly describe how business value is used in Extreme Programming and Scrum.

Extreme Programming: The Planning Game. Extreme programming consists of values, principles and basic activities [4]. One of the twelve practices included in the first version of Extreme Programming is the planning game [4]. A popular version of this is planning poker [9]. The planning game is a practice that aims to facilitate the communication and collaboration between business stakeholders and development staff, which according to Beck, often might have contradictory goals.

The planning game is executed [4] with story cards (containing features) as game pieces and game moves in three phases. There is a variation of planning poker for agreeing on business value, called the Business Value Game [1]. The game is played in the same way, but with cards representing business value instead of time.

Scrum: Sprint Planning Meeting. Scrum is a popular project management framework introduced by Jeff Sutherland and Ken Schwaber [20]. At the core of the Scrum framework is the *Scrum team* responsible for delivering the features in the *backlog* supported by a *product owner* responsible for maximizing the value of the product and maintaining the product backlog.

Potentially shippable increments of the developed product are created every sprint, a time-boxed unit of work no longer than one month. Before each sprint the features to be implemented are selected from the prioritized backlog by the Scrum team in the sprint planning meeting.

An extension to Scrum in large-scale contexts proposed by, e.g., de-Ste-Croix is the introduction of product owner teams [21] when the setting is too great for one person to handle. De-Ste-Croix proposes that the product owner team consists of experts from relevant fields and serves to jointly gather requirements and prioritize features.

As we can see from the two examples presented here – Extreme Programming and Scrum – different agile and lean methods have different planning practices. The common denominator is that they all have some means of using the business stakeholders' knowledge about business value for creating a concrete plan for implementation [16]. They do not, however, discuss how the business stakeholders obtain this knowledge. This is probably due to the fact that the typical context for agile methods is small companies with one development team and one actively involved customer. As agile and lean methods are gaining popularity also in other contexts, the need for a more explicit way of obtaining knowledge about business value arises. In the following section we present our proposed model for feature business value in large-scale agile settings.

3 A Proposal for a Model for Business Value

Based on the needs of the industrial partners of the Cloud Software Finland research project, we propose a model for representing business value. This model takes into account more than just monetary value. Even though monetary value is an important factor when prioritizing features of a software product, we find that there are other factors that also have an impact on the feature prioritization process. We believe that many software professionals already take these into account implicitly. With our model, we make these factors explicit and visible.

The model consists of six attributes, each graded from one to four, four being the highest value. Each of the attributes contribute to the notion of business value and were chosen to make the term more unambiguous to discuss between stakeholders, such as product owners.

We have selected a set of attributes that: (i) supports agile values by promoting communication and interaction between customer representatives and developers; (ii) supports both the customer's and the development organization's goals; (iii) supports long-term growth and development instead of solely aiming for short-term economic revenue; (iv) helps developers understand business goals; and (v) helps business stakeholders understand development goals.

3.1 Detailed Model Description

Our proposal for a model consists of six attributes that jointly contribute to the term business value. The six attributes are: 1) Monetary Value, 2) Market Enabler, 3) Technical Enabler,4) Competence Growth, 5) Employee Satisfaction, 6) Customer Satisfaction.

The attributes are given classifications on an ordinal scale, (described by Fenton & Pfleeger in [8]). The scale is graded from one to four, one being of least rank and four being of highest rank. This grading was chosen to avoid a possible extensive misuse of a middle value, instead aiming to require a decision from users to decide on a slightly lower or higher value. The total business value can then be expressed as a aggregation of the attributes, enhanced by a graphical representation depicting the relative impact of each of the attributes.

In the following sections we describe the different attributes together with example categories to illustrate the practical meaning of the attributes. The graphical representation is introduced in Section 4.

Monetary Value (MV). Estimated monetary value of the business case for a given feature can be expressed in monetary value or in points. This value is usually the result of the initial business analysis of the feature. The range of possible values is split into four categories from smallest (I) to largest (IV). The actual value ranges and categories will have to be decided based on the typical monetary value of features in the company in question. The numbers below are purely hypothetical and do not reflect the actual sizes in any of the companies in the Cloud Software Finland project.

I 0-50K€
II 50-100K€
III 100-200K€
IV 200K€+

Market Enabler (ME). If a feature is considered a market enabler it facilitates introduction to or retaining a market possible. For example, adding support for special characters in a software program can act as a market enabler for countries demanding native character support. When it comes to introduction to a new market or retaining an existing one, both the customer paying for the software and the developer aiming to sell the software can be seen as stakeholders. We suggest the following example categories:

I No enabler, can be added at any time without missing a marketing window.
II Feature can potentially work as a market enabler, alternatively, delays might affect marketing or sales aspects.
III Feature is to a significant degree a market enabler, alternatively, delays will to a significant extent harm or counteract marketing and sales efforts.
IV Feature is critical to complete before any other features are considered.

Technical Enabler (TE). A feature or other unit might work as a foundation for future functionality. These features are called technical enablers since they are inevitably needed to achieve the goal of the system under development. However, these enabling features do not necessarily provide any revenue or added value, which might lead to their omission in business value discussion. Examples of technical enablers include the introduction of support for new hardware types, the implementation of common functionality such as libraries or frameworks to be used by later features, or refactoring or other quality improvements that reduce the cost of maintenance. By including technical enablers in our model we want to take into account their importance, although they may not directly create revenue. We suggest the following example categories:

I No enabler, separate from all other features.
II Feature contains some foundation for further functionality.
III Feature contains significant foundation for further functionality.
IV Feature is critical to complete before any other features are undertaken.

Competence Growth (CG). Competence is an important key to success in software engineering. As argued by, e.g., Armour [3], knowledge is one of the central products of software development. In contrast to this, developing features requiring new knowledge with a steep learning curve might be less profitable in the short term, than developing software at high capacity using familiar methods. Learning new tools or techniques can however be important for the future and new competence may be of value to the development organization. We include this into the notion of business value since features not creating immediate revenue but providing competence growth can still be valuable. We suggest the following example categories:

I Maintaining knowledge in some fields.
II More complex feature, maintain knowledge in multiple fields, using technologies or techniques not used on a daily basis.
III Feature where some new tools and/or technologies needs to be used. Some delays due to learning are expected, teams competence grows into new fields.
IV Complex feature where a significant amount of new tools and/or new technologies are needed for progress. Delays are expected and the team's competence grows significantly.

Employee Satisfaction (ES). Since programming and software development can be regarded as creative work activities, the satisfaction and well-being of employees can be seen as an important factor in success. Many big companies such as Google employ programs allowing employees to work an amount of their official work time on projects of personal interest [11]. Even in day-to-day development work when electing and prioritizing features employee satisfaction could be an important factor in contributing to the business value and cost savings [22]. When not taking employee satisfaction into account, higher employee turnover might occur, resulting in more frequent training of new staff and reduced productivity. We suggest the following example categories:

I Work consists of routine tasks. Team members either do not understand why the feature is important or how it connects to the product. Neutral or even possibly some slight negative satisfaction.

II Work consists of a combination of routine and varying tasks. Team members have some understanding of why the feature is important and how it connects to the product. Positive or neutral satisfaction.

III Work consists of tasks not normally carried out, exploring new possibilities and viewpoints. Team members understand why the feature is important and how it connects to the product. Positive satisfaction.

IV Work consists of self-fulfilling tasks, which are novel and creative. Strong positive satisfaction by all team members.

Customer Satisfaction (CS). Customer satisfaction is often considered important in business as satisfied customers often return and continue collaboration. One proposed measurement of customer satisfaction is net promoter score [15] by Mary and Tom Poppendieck. Keeping customers satisfied might be expensive, making the decision to include or exclude certain desired features difficult. Obliging to all customer demands can be unprofitable or negatively impact other attributes. Caring too little about customer satisfaction can also have negative impact on future collaboration. We suggest the following example categories:

I A low risk feature or enabler that does not deliver high value but is demanded by the customer and is easy to verify as correct. Neutral satisfaction.

II A feature that delivers business value to the customer and helps other features by reducing risks, enabling or solves a problem in the customer's environment. Neutral satisfaction.

III A feature with higher risk that delivers substantial business value to the customer or is otherwise important to the project. Successful implementation always gives the customer positive satisfaction.

IV A critical feature that delivers a lot of value to the customer or is otherwise very important to the project. Successful implementation is crucial to the project and is much appreciated. Always gives strong positive satisfaction.

In the beginning of this section (Section 3), we listed five needs that we claim to fulfill with the six attributes presented above. We now argue for why we think they are fulfilled. The first need listed is to support agile values by promoting communication between the customer and the developers. This is achieved by the attribute set acting as a communication tool, making explicit the different aspects of business value. The second need: that both the customer's and the development organization's goals are met, is fulfilled since both viewpoints are present in the list of attributes. Attributes MV, ME, and CS are customer oriented, while attributes TE, CG, and ES are more geared towards the development organization's needs. The third need is concerned with long-term growth. This is met by taking into account long-term benefits from a market (ME, CS), technical (TE) and people (CG, ES) perspective. The fourth and fifth needs listed above (developers and business stakeholders understanding each other's goals) are supported by the model making explicit that the other party's goal is multifaceted. The business goals are not only about money (MV), but also about enabling new markets (ME) and satisfying the customer (CS). Similarly the development goals consist of enabling future technical growth using both technical (TE) and competence related (CG) means, as well as keeping the creative staff motivated (ES).

The purpose of the model proposed in this section is to serve as a tool for both business and technical stakeholders. We now proceed to illustrate how this tool could be used in software organizations.

4 Intended Use and Visualization

The business value model we propose is intended to support both business stakeholders and technical staff in software development projects. We see three main usage scenarios: product backlog prioritization, sprint planning, and business information radiator.

The business stakeholders can use the model when they prioritize the features in the product backlog. This is especially helpful in large settings, where multiple business stakeholders, such as a product owner team, have to agree on a prioritization. We suggest that our approach is combined with an agile planning practice, such as Agile 42's business value game [1].

When the business stakeholders and development team plan the upcoming sprint, our model can be used as a means of communication to clarify the different aspects of business value. This communication can be further enhanced if used in combination with a visualization format such as the one suggested in Figure 2. In this suggestion, each attribute is represented by a piece of the pie chart, where the relative size reflects the relative rank of the attribute in question. The order of the attributes is the same as in Section 3.1, starting from the topmost position and continuing clockwise (i.e., monetary value, market enabler, technical enabler, competence growth, employee satisfaction, and customer satisfaction.)

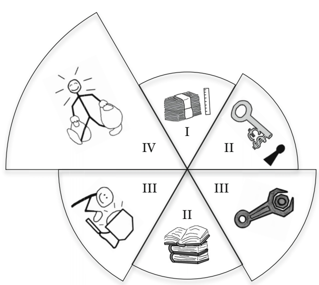


Fig. 2. Proposal for visualization of business value attributes

The final usage scenario is to add information about business value in the information radiators [6] used by the project, such as the sprint burndown chart. Even a simple list of features with a pie chart representing each of their business value can help the team understand what kind of value their work is adding to the product.

5 Discussion and Future Work

In this paper we propose a model of business value that is intended to support feature prioritization in the development of software using agile and lean methods on a large scale. Racheva et al. has done comprehensive work in analyzing the state of the art and practice [17,18] regarding the definition of business value in agile organizations. They arrive at the same conclusion as we do: the definition of business value is unclear and varied. Racheva et al. deduce a process model for how requirements are prioritized and reprioritized during an agile project [16]. Our work is complementary to theirs. Where Racheva et al. model a process for prioritization, we model the business value itself.

The Poppendiecks suggest a solution for defining and communicating business value to the team through the practice of providing each team with an accountant [15]. We find this to be a good suggestion, well in line with our proposal. The support an accountant can bring corresponds to the monetary value and market enabler attributes of our model. We do, however, suggest attributes that are not included in the Poppendiecks' solution, such as employee satisfaction, technical enabler and competence growth. Furthermore, we believe that one accountant per team can grow too expensive in larger settings. If an accountant is available, we instead suggest that the accountant support the product owner team in the process of defining business value for the features.

The most extensive set of agile prioritization factors is suggested by Cohn [7]. His definition of business value is the traditional one, based on financial return and cost. But he also lists other factors to be taken into account when planning an agile project. These are learning, risk and desirability. Learning corresponds somewhat to our competence growth attribute, although Cohn's definition focuses more on learning about the product and the progress of the project, whereas we also factor in the general competence growth of the project staff. Desirability corresponds to our customer satisfaction attribute.

There are attributes which we choose not to include explicitly. Managing technical debt through refactoring and restructuring is a crucial part of software development. This need is exacerbated when using agile methods, since they usually expect the architecture of the software to emerge. We consider this to be a part of the technical enabler attribute.

Implementing a certain feature may involve risks. This is not explicitly taken into account and a future improvement of our model could include this as a separate aspect. Now it is indirectly included in the customer satisfaction attribute.

Another aspect which we have chosen to exclude for now is the notion of an expiration date. Some features may have really high value if they are implemented rather quickly, for instance before the competition, but after a certain deadline, the value decreases dramatically. The market enabler attribute of our model is a first step towards supporting the notion of an expiration date, but it could be developed further to better include this aspect. At this point, we consider this concept to be too complicated to include in an easy-to-use model.

Our business value model is as of yet just a concept proposed as an aid in a challenge faced by the industrial partners of the Cloud software research project. Future work includes an evaluation and incremental improvement of the model when it is applied.

6 Conclusions

In this paper we have proposed a model for business value for use in software companies that apply agile and lean methods. The purpose of this model is to make explicit the factors that may be taken into account when the business stakeholder prioritizes the backlog for the development team(s). Our intention is for this to benefit the business stakeholder(s) when prioritizing the backlog, as well as the communication on the topic of business value between business stakeholders and software development teams. We also suggest a simple means of visualizing the resulting business value.

The concept of business value is central to the field of agile and lean methods, where one of the central tenets is to create value for the customer. However, as Racheva et al. have demonstrated, the definition of the term business value is vague and ambiguous, both among practitioners [18] and in the literature [17]. While Racheva et al. seek to remedy this by creating a process model for the requirement prioritization process in agile and lean teams [16], our proposal is a model of the concept of business value in itself.

Our proposal for a model consists of six separate attributes, which each model a different aspect of business value. The purpose of this model is not only to support the business representatives in their prioritization work, but also to serve as a communication aid between the business stakeholders and the development team. This can be especially helpful in large and complex settings, where the potentially contradicting needs of multiple customers need to be communicated to multiple development teams.

The business value model we have proposed in this paper is still early work that needs evaluation through implementation and incremental improvement to reach maturity.

Acknowledgements. This work was funded by the CLOUD Software Finland project [5]. From Ericsson Oy we would like to thank Kaisa Kettunen, Outi Väättänen and Leena Pitkäranta for their help and support.

References

1. Agile 42: Business value game,
http://www.agile42.com/agile-scrum-tools/business-value-game/

2. Ambler, S.W.: Has Agile Peaked? Dr. Dobb's Journal: The World of Software Development 33(6), 52–54 (2008)
3. Armour, P.: When executives code. Commun. ACM 47, 19–22 (2004)
4. Beck, K.: Extreme Programming Explained (2000)
5. Cloud Software Finland, www.cloudsoftwareprogram.org
6. Cockburn, A.: Crystal clear a human-powered methodology for small teams. Addison-Wesley Professional (2004)
7. Cohn, M.: Agile Estimating and Planning. Prentice Hall PTR, Upper Saddle River (2005)
8. Fenton, N., Pfleeger, S.L.: Software metrics, 2nd edn. A rigorous and practical approach. PWS Publishing Co., Boston (1997)
9. Grenning, J.: Planning poker or how to avoid analysis paralysis while release planning (2002)
10. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004), http://www.jstor.org/stable/25148625
11. Mediratta, B., Julie, B.: The google way: Give engineers room. The New York Times (October 2007)
12. Osterwalder, A., Pigneur, Y., Clark, T.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley Desktop Editions Series. John Wiley & Sons (2010)
13. Patton, J.: Ambiguous business value harms software products. IEEE Softw. 25, 50–51 (2008)
14. Pettit, R.: Business value applied: Aligning the day to day with business imperative. Agile Journal (2007)
15. Poppendieck, M., Poppendieck, T.: Implementing Lean Software Development: From Concept to Cash. Addison-Wesley Professional (2006)
16. Racheva, Z., Daneva, M., Herrmann, A., Wieringa, R.J.: A conceptual model and process for client-driven agile requirements prioritization. In: 2010 Fourth International Conference on Research Challenges in Information Science, RCIS (2010)
17. Racheva, Z., Daneva, M., Sikkel, K.: Value Creation by Agile Projects: Methodology or Mystery? In: Bomarius, F., Oivo, M., Jaring, P., Abrahamsson, P. (eds.) PROFES 2009. LNBIP, vol. 32, pp. 141–155. Springer, Heidelberg (2009)
18. Racheva, Z., Daneva, M., Sikkel, K., Buglione, L.: Business Value Is Not Only Dollars – Results from Case Study Research on Agile Software Projects. In: Ali Babar, M., Vierimaa, M., Oivo, M. (eds.) PROFES 2010. LNCS, vol. 6156, pp. 131–145. Springer, Heidelberg (2010)
19. Rawsthorne, D.: Managing the work in an agile project (2004), http://www.netobjectives.com/files/resources/downloads/ManagingTheWork.pdf
20. Schwaber, K., Sutherland, J.: The scrum guide (2011), http://www.scrum.org/scrumguides/
21. de Ste-Croix, A., Easton, A.: The product owner team. In: Proceedings of the Agile 2008, pp. 274–279. IEEE Computer Society, Washington, DC (2008)
22. Tessem, B., Maurer, F.: Job Satisfaction and Motivation in a Large Agile Team. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 54–61. Springer, Heidelberg (2007)

Establishing a Continual Service Improvement Model: A Case Study

Sanna Heikkinen and Marko Jäntti

University of Eastern Finland, School of Computing
Software Engineering Research Unit
P.O. Box 1627, 70211, Kuopio, Finland
{sanna.heikkinen,marko.jantti}@uef.fi

Abstract. The Continual Service Improvement (CSI) section of the IT Infrastructure Library (ITIL) version 3 provides IT companies with best practices for the improvement of services and service management processes. Unfortunately, many IT companies consider ITIL-based practices (including CSI) too abstract for their purposes. The research problem in this study is: which methods and practices are related to Continual Service Improvement in IT service management? The main contribution of this paper is an improved version of the CSI model that provides a more detailed and practical view of CSI activities: measurement, reporting and processing of the service development ideas. Our model is compatible with ISO/IEC 20000 standard requirements and ITIL v3 practices. The model emphasizes the importance of change management process in the management of development ideas. The CSI model was created in the cooperation with a Finnish IT service provider company and validated with three different service provider companies.

Keywords: IT service management, continual service improvement, ITIL.

1 Introduction

Due to continual change in technologies, services, products and organizational structures the IT service management is a challenging task for IT organizations. Processes, functions, and services require continual improvement in order to generate positive business results. By improving the service management process maturity, service excellence can be reached. Effective and efficient IT service management very likely leads to improved performance, early detection and prevention of errors and to address problem areas.

Information Technology Infrastructure Library (ITIL) is a set of good practices for IT service management [1]. ITIL is the most widely used IT service management (ITSM) approach that has gained a de facto standard status. ITIL focuses on aligning IT services with the needs of business. ITIL version 3 approaches service management from the service lifecycle viewpoint. The service lifecycle describes the way how service management is structured [2]. The service

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 61–72, 2012.

lifecycle consists of five phases: Service Stategy [3], Service Design [4], Service Transition [5], Service Operation [6] and Continual Service Improvement [7].

In this paper, we clarify how organizations measure, report and use the data to improve not only the new processes but also the IT services. To run the CSI as a process, we need clearly defined goals, documented procedures, inputs, outputs and identified roles and responsibilities. In order to be successful CSI must be embedded to the organizational culture [7]. The primary purpose of CSI is to continually align and realign IT services to the changing business needs by identifying and implementing improvements to IT services that support business processes. These improvement activities support the whole service lifecycle. CSI aims to find new ways to improve process effectiveness, efficiency as well as cost effectiveness.

Measurement is an important part of the service management system by steering and controlling IT to the desired direction. Thus, metrics must be designed to match customer requirements, benchmarked to ensure that they are achievable and monitored to ensure that they keep within desired threshold with action taken to correct any problems [8]. Metrics play an important role the Continual Service Improvement because the processes and services must continuously improved. Since improvement initiatives will be more than likely require changes, specific improvements will need to follow the defined ITIL Change Management process [7].

1.1 Related Work

Continual service improvement can be related to improving both organizational and process performance. Some studies deal with business perspective and services while others focus on process, people and technology perspectives. Much has been written about IT Service Management and especially implementing best practices of ITIL [9], [10]. Wegmann et al. [11] have described the main principles of ITIL Service Management and presented how ITSM methods can contribute to the definition of an SLA (Service Level Agreement) by modelling the service provided by an IT department, the stakeholders of this service and the value the stakeholders expect from this service.

Pyon et al. [12] report that service improvement should be considered from process viewpoint and customer viewpoint in the financial service industry. They have proposed a Business Process Management (BPM) framework for managing customer complaints.

Lahtela and Jäntti [13] have examined what types of challenges are related to the service support interface between an IT service provider and IT customers and have identified challenges such as problems in priorization of support requests, challenges in information sharing, and poor transparency of support processes. Jäntti and Kalliokoski [14] have presented knowledge management challenges in the service desk function on a large IT service provider company in Finland. Challenges were related to incident classification, quality of instruction documents, automatization of incident and order processes, escalation of incidents and the interface between incident management and problem management.

Moreover, Jäntti and Järvinen [15] have carried out a case study that focused on evaluating the deployment of an incident management process.

Continual service improvement is included in many standards and frameworks in addition to the ITIL framework. CSI is visible in the Plan-Do-Check-Act requirements of ISO/IEC 20000 standard [16] and COBIT framework (PO8:Manage Quality) [17].

The ISO/IEC 20000 standard [18] requires that organizations continually improve their services and document the improvement actions and results. Lima et al. [19] have studied improving the quality view. They have dealt with how to estimate quality percent of an IT service, in order to provide a continual activity in the service life cycle.

1.2 Our Contribution

This paper belongs to the results of ongoing KISMET (Keys to IT Service Management and Effective Transition of Services) research project.

The Continual Service Improvement in the IT Service Management was one of the KISMET focus areas. The main contribution of this paper is to:

- describe how the Continual Service Improvement process model was established,
- describe the main elements of the CSI process model and
- describe how the research team validated the CSI process model.

The goal of this study was to create a systematic model for managing improvement actions concerning IT services and service management processes. We aimed at helping IT service provider organizations to achieve a high quality of service and customer satisfaction. The study was carried out between August 2010 and June 2011.

The rest of the paper is organized as follows. In Section 2, the research problem and methods are described. In Section 3, we present the CSI process model and describe its elements. In Section 4, we provide the analysis of findings in the form of lessons learnt. The conclusions are given in Section 5.

2 Research Methods

This case study is a part of the results of KISMET (Keys to IT Service Management and Effective Transition of Services) research project at the University of Eastern Finland. The research problem of this study is: Which methods and practices are related to Continual Service Improvement in IT Service Management?

We used a case study research and constructive research methods with a single case organization to answer the research problem. The research problem was divided into following three research questions: 1)What is CSI and what functions, roles and responsibilities are related to it? 2)How IT services are measured and reported? 3)How does the CSI deal with the improvement initiatives?

In this study, we used a qualitative research approach to improve IT service provider's processes and practices. Additionally, we exploited triangulation of methods, a combination of the study case research, constructive research and literary review. The main contribution of the study was a detailed CSI process model that was targeted to IT service provider companies. We aimed to create a model that IT companies could use while improving their service management. Our model was divided into three sections: measurement, reporting and processing of the improvement ideas.

According to Yin [20] a case study is "a research strategy which focuses on understanding the dynamics present with single settings". A typical feature of the study is that the focus is often the processes. The case study focuses on studying and understanding of a particular object of study, activity in connection with their environment [20]. During the case study researcher is an outsider, who observes and analyzes the environment, making notes by combines different data collection methods.

Additionally, we used a constructive research method which is widely used in software engineering and computer science. Constructive research aims at producing novel solutions to practically and theoretically relevant problems. Constructive research can be defined as "managerial problem solving through the construction of models, diagrams, plans, organizations, etc.". To be considered constructive research, the research must combine problem solving and theoretical knowledge. The third method that was used in this study was a small-scale literary review. It focused on the studies in IT service management (ITSM), Information Technology Infrasfructure library (ITIL) and ISO/IEC 20 000 standard. It revealed that very few studies had investigated continual improvement from the IT service management viewpoint.

Our case organization Alfa is a Nordic IT service provider organization that has operations both in Finland and in Sweden. Alfa provides companies and organizations with easy-to-use IT services. Alfa has around 700 employees and its turnover was approximately EUR140 million in 2011. Alfa provides various types of services to its customers: application services, desk top services, servers and capacity services, network services etc. The case organization was selected by using elite sampling. Alfa was a unique case in Finland with a strong focus on ITIL, ISO/IEC 20000 and ITSM tool improvement. The case study with Alfa was performed in three separate phases (August 2010 - July 2011).

Multiple data sources were used in collecting data from the case organization's service management processes:

- **Documents**: Administrative documents, progress reports, meeting memos and other internal records
- **Archival Records**: Incident and service request records **Interviews and discussions**:
- **Participative Observation**: Process improvement meetings with CSI Managers and Process Managers from the case organizations
- **Physical Artifacts**: Access to the case organization's intranet, reporting tool, process tool and customer support tool

The data were collected in three phases. The start of the study and data collection started in August 2010, when a researcher began her master studies trainee period (3 months). The goal of the period was to identify and study the ITIL framework and start the development of CSI model. During the period the researcher had the opportunity to explore the literature review, participant observation, and business documents using. The first phase of the study resulted in the first version of CSI (Continual Service Improvement) model as well as descriptions of the roles and process metrics.

The phase 2 started in middle of the November, 2010 when the researcher started working in the KISMET project. The work included designing and building a process description for Continual Service Improvement. During the phase 2, the researcher carried out the literature review and analyzed the archival records. The result of the phase 2 was the 2.0 version of the CSI model and a process description of Improvement Process (v 0.1).

During the phase 3 researcher worked in the KISMET research pilot with the case organization. The organization carried out the study to identify and improve ITSM practices. Data collection was done by using the literature review and corporate documents and participant observation (meetings, information and training sessions). The main result of phase 3 was a model to support the handling of development proposals.

2.1 Data Analysis

A within case analysis technique [21] was used in this study with content analysis methods. Especially, theoretical content analysis was used in the data analysis. Content analysis is a research method for making replicable and valid inferences from data to their context, with the purpose of providing knowledge, new insights, a representation of facts and a practical guide to action. The aim is to attain a condensed and broad description of the phenomenon, and the outcome of the analysis can be concepts or categories describing the phenomenon. Usually, the purpose of those concepts or categories is to build up a model, conceptual system, conceptual map or categories.

3 Establishing a CSI Model

In this section, we will introduce how the CSI model was established in cooperation with the case organization and the KISMET project research team. The research work consisted of three main phases: Phase 1: Defining the research problem and investigation, Phase 2: Building the CSI model, Phase 3: Validation of the model. These phases are described in the following subsections.

3.1 Defining the Research Problem and Investigation

The kickoff meeting of the pilot project was arranged in August 16, 2010. In that meeting, the representatives (CSI manager, process owner of the incident management, change management and configuration management) of the case organization reported that they would like to improve the process of managing improvement initiatives. **The diagnosed problem:** Lack of systematic and easy-to-apply model for continual service improvement. The research work started by exploring the practices and tools of measurement and reporting in the case organization. The organization already used IT service management framework ITIL. Creating a process framework for CSI was started from the 7-Step improvement model of ITIL framework. Additionally, the researcher started examining ISO/IEC 20000 requirements for PDCA cycle.

3.2 Building the Continual Service Improvement Model

The CSI model was improved in two meetings with the case organization. Implementation of improvement ideas was combined to the change management process because change management is a natural place to handle improvement ideas. Figure 1 describes the activities of the CSI model.

CSI model has been designed to demonstrate and clarify how services and processes can be systematically improved through measurement, reporting and management of improvement ideas. Next, we briefly describe the elements of the model. Regarding the measurement, it is essential that company's business vision and goals have been identified before one can start building process performance metrics. It is possible that current tool does not meet the requirements of IT service management. Thus, the organization should make a decision whether to buy a new tool or improve the existing one. The organization needs three types of metrics to support the continual service improvement: technology, process and service metrics. The organization should define limit values for measurements to be able to benchmark results and identify deviations. Measurement data can be collected from service desk tools, monitoring tools, existing reports etc.

In reporting phase, collected data is converted into desired form and audience. Quite often, organizations have a reporting unit that is responsible for creating reports and maintaining reporting technologies and tools. Data should be processed into information to enable more effective data analysis. Reports are produced monthly or weekly to customers and management. In case of service level breaches, one should discuss about required improvements. Identified improvement areas are reported to all relevant stakeholders.

During the improvement phase, improvement areas can be identified by using several ways, such as customer feedback, survey results and audit and review results. Identified improvement ideas are dealt within the change management process. Change managers and Change Advisory Board meetings evaluate, prioritize, authorize and plan changes. The scope and business impact of the improvement defines whether it shall be implemented by an employee or an implementation group. The change can also turn out to be a major change. In that case, it shall likely initiate a project. The project management has a clear interface with continual service improvement. CSI should analyze the lesson logs of IT projects and define improvement actions. Project management shall likely produce project tails that are inputs for CSI. At the end of the improvement phase improvements

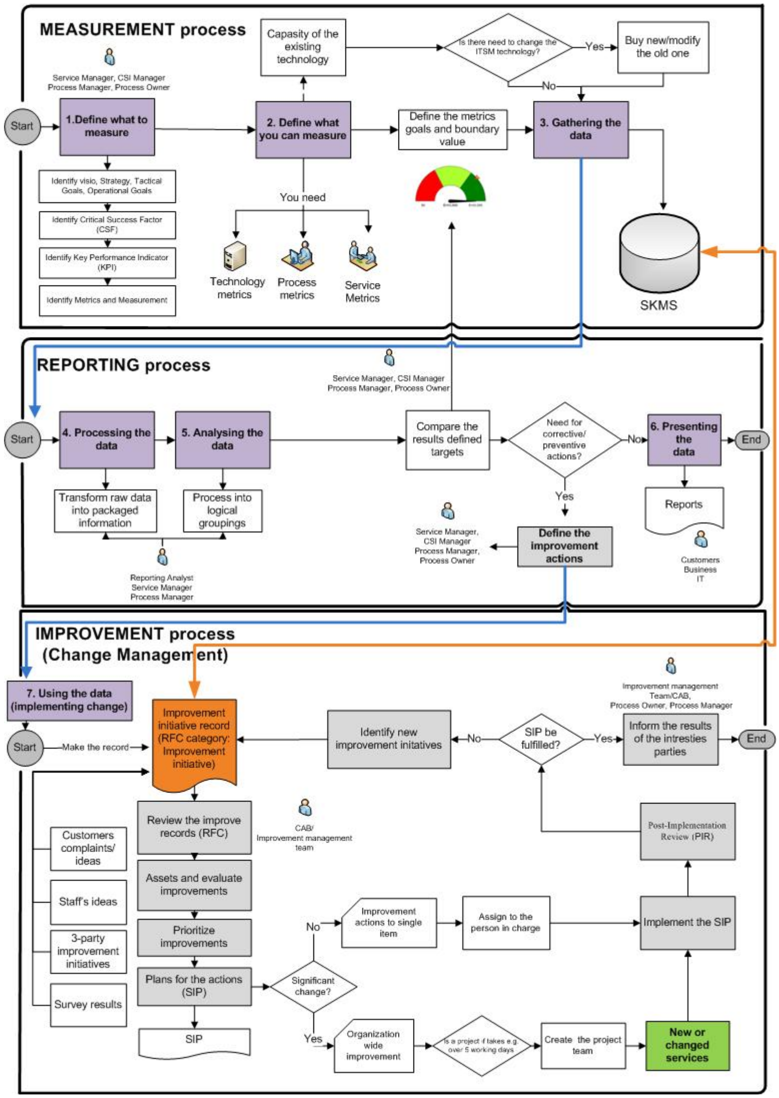


Fig. 1. The CSI model

are implemented, post-evaluated and communicated to stakeholder groups. Post-implementation review should be carried out both for failed changes and successful changes.

Swimlines were added to the CSI models: change manager, change advisory board, design and transition of new and changed services. Additionally, a new action, handling project tails, was inserted to the model. Project tails are issues (for example, change requests) that are not implemented during the project but need to be transferred to the operation. Usually, project tails are identified while writing the closure report of the project.

3.3 Validating the CSI Model

Besides Alfa, the CSI model was validated with two other organizations, 1) Beta, the Information System Management unit of a government agency that provides IT services (application services, user support, desktop, servers) to employees, 2) Gamma, an ICT provider company that provides IT services (server and data center services, network and telecommunication services, ICT acquisition services, IT consulting, project and introduction services) mainly for 2 main customers. Validation was done in three meetings June 15 (Alfa), October 5 (Beta) and November 29 (Gamma). The following comments have been collected from validation meetings. In order to keep comments anonymous, we have not categorized them according to companies.

- "In our company, especially incident management and service request management are measured very well"
- "Measurement plays an important role in IT service management"
- "It is not enough to measure volume of support requests, we need to know how much one ticket consumes resources?"
- "Regarding change management, we need to identify different types of changes: emergency changes, strategic changes..."
- "Good model that could also be used for other purposes than solely managing improvement ideas"
- " It is good to see interface between projects and CSI"
- "We would need unified reporting and measurement practices to our organization"
- "What is the interface between change and improvement idea in practice?
- "What types of inputs project managemement produces for CSI? We are planning PRINCE2-based project management."
- "We measure montly the customer satisfaction on service desk cases"
- "Additionally, we carry out reviews for projects and applications"
- "We have a lot of service desk related metrics: User support resolution time and volumes, volume of support requests assigned to local support"
- "We receive quite few formal complaints. Most common causes of the complaints are unclearly documented resolutions, delays in receiving answer or service and conflicting instructions given by service desk specialists"

Based on the validation meetings, we observed that IT service providers considered the model useful but still needed some practical guidelines to support the process model.

4 Analysis

In the analysis phase of this study, we summarized case study findings in the form of lesson learned. A source for each lesson is presented in parentheses (AR= Archives and records, D= Documentation, ID= Interviews and discussions, O= Observation, PA= physical artefacts, ST= Seminars and trainings organized by the research group).

Lesson 1: Create an organization-wide CSI policy (ID, O, AR, PA, D)
We observed that various types of continous improvement methods were used in the case organization. However, there was not a systematic organization-wide model for managing improvement suggestions. The same issue was addressed by Beta and Gamma. In Alfa, improvement ideas were assigned to unit leaders who shall analyze them and initiate improvement actions. Improvement ideas are often discussed on hallways, email and in meetings. The goal would be to capture and store them to the IT service management system.

Lesson 2: Start the measurement program by defining CSFs, KPIs and Metrics (AR, ID, D)
During the case study, we observed that the organization had numerous performance metrics in use. Additionally, IT service measurement was focused on certain parts of IT service production, especially, service desk, incident management and service request management. In order to select few good metrics that support the core business requirements one could use CSFs, KPIs and metrics as a basis of the measurement program.

Lesson 3: Clarify the concept of Service Improvement Plan (AR, ID)
During the study researchers were asked many times what is the SIP in practice. Both ITIL and ISO/IEC 20000 use the term SIP. According to ITIL a SIP is a formal plan to implement improvements to a process or IT service. An IT service provider could simply implement a SIP by collecting data on improvement suggestions regarding processes and services, analyzing them frequently (for example, monthly), creating requests for change to implement improvements and reviewing improvements.

Lesson 4: Check the interfaces between CSI and other processes (O, ID)
During the validation, we noticed that companies were really interested in the interface between continual service improvement and other ITSM processes such as change management and design and transition of new or changed services. In our model, change management is an active participant in the CSI process and responsible for managing the development ideas like other changes. We also added the interface to project management while organizations were very interested in that.

Lesson 5: Remember communication about the results of Continual Service Improvement (D,ID)
Organization had put a lot of efforts on creating unified service management practices. For example, there had been mandatory company-internal ITIL trainings for service center employees. During training events (organized four times per year) employees receive information on actual improvement plans of the organization and changes on work practices. Some employees indicated the need to get more information on improvements.

Lesson 6: Identify the sources of improvement ideas (O)
We observed that employees have difficulties to identify sources of improvement ideas and define a systematic process for handling them. The sources of improvement ideas may include feedback from customers and personnel, research results and survey results, third-party feedback, measurement results and project closure reviews. The organization would need clearer methods to support the process and service improvement because at the moment a lot of time goes to reactive work, "fire fighting" and there is often no time for proactive activities.

Lesson 7: Defining roles, responsibilities and tasks supports the service improvement (O, D)
Through the processes, people's expertise can be converted into knowledge, skills, models and theories. During the study, we defined two new roles; CSI manager and process owner role. The goal behind the CSI manager role is to enhance managing improvement suggestions and introduce the CSI methods in practice. The process owner is a role that ensures that the process follows agreed and documented practices and reaches the defined process objectives.

Lesson 8: Create a CSI calendar for process and service reviews (DI, O)
Organizing process and service audits and reviews is one of the CSI teams core tasks. An open CSI calendar with review dates would help process and service managers to prepare better for reviews. Internal reviews can be performed, for example, to check whether support request handling has followed documented processes and to identify deviations and their root causes. It is also a good idea to use external auditors frequently to identify improvement areas.

Lesson 9: Identify the wide scope of CSI (ID, O, AR, PA, D)
In order to implement CSI and identify improvement areas one can use varius methods and tools. The following list is based on our findings from the case organization Alfa: customer satisfaction surveys, process and quality improvement meetings, writing work instructions, carrying out trend analysis for incidents and service requests, weekly team meetings, project lesson logs and analysis of measurement data.

5 Conclusions

The ITIL framework provides best practices for continual improvement of services and processes. However, many service providers are not familiar with CSI practices or consider CSI practices too abstract for daily use. Additionally, the ITIL presents the CSI activities (measurement, reporting and management of improvements) as separate actitivies lacking the unified view of CSI. Therefore, CSI is a fruitful research target. The research problem in this study was: Which methods and practices are related to Continual Service Improvement in IT service management?

The main contribution of this study was to present an improved CSI model that provides a detailed description of activities within measurement, reporting and processing improvement ideas. The model was created in cooperation with a Finnish IT service provider organization Alfa that was the unit of analysis in the case study. In addition to Alfa, the model was validated with two other IT service provider organizations who provided valuable comments for the further work. Finally, we provided lessons learnt from establishing and validating the model.

This study included the following limitations. First, the CSI model was build mainly with one case organization. Multiple organizations would have provided a richer view on the structure of the model. However, we carried out the validation phase with two additional service providers. Second, generalization of the results might be weaker due to small number of cases. In order to increase the quality of the case study, we used various forms of triangulation: method triangulation, researcher triangulation, and data triangulation. Third, the model was validated with manager-level persons from IT service provider organizations. We could have had validation meetings with employees such as service desk workers who enter the data on improvement suggestions to the ITSM tool. Unfortunately, the lack of time was a limiting factor.

To summarize, more case studies are welcome in the area of continual service improvement. Further work could focus on deployment of Continual Service Improvement practices, analyzing the bottlenecks in CSI methods or investigating the interfaces of CSI with other IT service management processes.

Acknowledgment. This paper is based on research in Keys to IT Service Management and Effective Transition of Services (KISMET) project funded by the National Technology Agency TEKES (no. 70035/10), European Regional Development Fund (ERDF), and industrial partners.

References

1. Cabinet Office: ITIL Service Strategy. The Stationary Office, UK (2011)
2. IT Service Management Forum: Itsmf website (2007), http://www.itsmf.org/
3. Office of Government Commerce: ITIL Service Strategy. The Stationary Office, UK (2007)
4. OGC: ITIL Service Design. The Stationary Office, UK (2007)
5. Office of Government Commerce: ITIL Service Transition. The Stationary Office, UK (2007)
6. Office of Government Commerce: ITIL Service Operation. The Stationary Office, UK (2007)
7. Office of Government Commerce: ITIL Continual Service Improvement. The Stationary Office, UK (2007)
8. IT Service Management Forum: Metrics for IT Service Management. Van Haren Publishing (2006)
9. Iden, J., Eikebrokk, T.R.: Understanding the itil implementation project: Conceptualization and measurements. In: Proceedings of 2011, 22nd International Workshop on Database and Expert Systems Applications. IEEE, Washington, DC (2011)
10. de Sousa Pereira, R., da Silva, M.: A maturity model for implementing itil v3. In: 2010 6th World Congress on Services (SERVICES-1), pp. 399–406 (2010)

11. Wegmann, A., Regev, G., Garret, G.A., Marechal, F.: Specifying Services for ITIL Service Management. In: Proceedings of International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements, SOCCER 2008, pp. 8–14 (2008)
12. Pyon, C., Woo, J., Park, S.: Service improvement by business process management using customer complaints in financial service industry. Pergamon-Elsevier Science Ltd., vol. 38 (2011)
13. Lahtela, A., Jäntti, M.: Improving IT Service Management Processes: A Case Study on IT Service Support. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 95–106. Springer, Heidelberg (2010)
14. Jäntti, M., Kalliokoski, J.: Identifying Knowledge Management Challenges in a Service Desk: A Case Study. In: Proceedings of the Second International Conference on Information, Process, and Knowledge Management (eKNOW 2010), pp. 100–105 (2010)
15. Jäntti, M., Järvinen, J.: Improving the Deployment of IT Service Management Processes: A Case Study. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 37–48. Springer, Heidelberg (2011)
16. ISO/IEC: ISO/IEC FDIS 20000:1, IT Service Management, Part 1: Service management system requirements. ISO/IEC JTC 1 Secretariat: ANSI (2010)
17. COBIT 4.1: Control Objectives for Information and related Technology: COBIT 4.1. IT Governance Institute (2007)
18. ISO/IEC 20000: IT Service Management, Part 1: Specification for service management. ISO/IEC JTC1/SC7 Secretariat (2005)
19. Lima, A., de Sousa, N., Oliveira, A., Sauve, J., Moura, A.: Towards Business-Driven Continual Service Improvement. In: 2010 IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksps), pp. 95–98 (2010)
20. Yin, R.: Case Study Research: Design and Methods. Sage Publishing, Beverly Hills (1994)
21. Eisenhardt, K.: Building theories from case study research. Academy of Management Review 14, 532–550 (1989)

Application of the ISO/IEC 15504 Standard Based Model – innoSPICE

Tanja Woronowicz[1], Jeremy Besson[2], Michael Boronowsky[1], and David Wewetzer[1]

[1] TZI-Universität Bremen, Am Fallturm 1, Bremen, 28359, Germany
[2] Vilnius University, 3 Universiteto Street, Vilnius, LT-01315, Lithuania
{worono,mb,wewetzer}@tzi.de, jeremy.besson@gmail.com

Abstract. The improvement of scientific knowledge and technology transfer (KTT) for a better economic sustainability is one of the European key challenges. innoSPICE is an evaluation procedure related to this kind of transfer that supports process improvement of knowledge-intense institutions to generate more innovation while helping investors and research institutions optimize public funds to achieve economic added value. As a standard-based model, *innoSPICE* has become an international instrument for quality management in the field of innovation and KTT. The model will be officially published within the family of *ISO/IEC 15504* models like Automotive SPICE or Enterprise SPICE. Since the established systematic approach to outcome quality management is process quality management, this process reference and assessment model (PRM) partially borrows from recognized process capability models and was tested by KTT practitioners from 13 European countries. As a quality management system, innoSPICE offers research organizations and those responsible for technology transfer an instrument by which they can support KTT activities and make organizational structures and their functionality more transparent through a standard report that can also be published to the outside. This paper presents an introduction to the *innoSPICE* Model, the assessment methodology and provides a first analysis of the *innoSPICE* assessments performed so far.

Keywords: Knowledge & technology transfer (KTT), innovation, capability assessment, innoSPICE.

1 Introduction

The successful transfer of knowledge and technologies from science to practice is one of the key European challenges. The constant technological improvement of products, services, processes and work environment is a critical factor for the development of our economy and society. It strongly depends on the ability to develop knowledge and technology, to transfer it from the point of generation to the site of adaptation and application, and to put the technology into use for the benefit of the acquirers and transferors. So far, there is no widely accepted, reliable, predictable and efficient method to evaluate to what extent an organization performs such activities, i.e., innovation, *knowledge- and technology transfer*[1] *(KTT).* The European Baltic Sea Region

[1] KTT will be used in the following as term for knowledge- and technology transfer.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 73–84, 2012.

INTERREG 4B Project *BONITA* [14] has developed a *KTT* Model, to support knowledge-intense institutions in generating more innovation while helping investors and research institutions optimize public funds for achieving economic added value. The BONITA transfer model will be published via the *ISO/IEC15504 SPICE user group* as standard-based model *innoSPICE*™. The *innoSPICE* process reference and assessment model (PRM and PAM) have been evaluated in more than 30 research and transfer organizations, ministries, science parks and business development agencies in 13 European countries. During these assessments, the model has proven its applicability and showed that it has reached an implementation level. Currently the model is mainly used to provide guided self-assessment evaluation and to support an intra-organizational *PDCA-Cycle* (*Deming Cycle* – Plan, Do, Check, Act). Understanding the potential behind the process assessment of innovation and *KTT* capability opens huge opportunities to support the improvement of processes for a better contribution of research to economic growth.

This paper focuses on the application aspects of *innoSPICE* and will describe the assessment methodology that was applied during the evaluation period of the model. We will present a brief description of the *innoSPICE* model, as more elaborate descriptions can be found at [3, 14]. We will present some details from three assessments as examples and provide a discussion of organizational profiles that were synthesized from assessments performed so far. The paper will close with a short conclusion and an outlook of next steps.

2 The innoSPICE Innovation, Knowledge and Technology Transfer Model

A motivation to develop a standard-based model like *innoSPICE* was the "urgent need to improve data availability and the breadth and quality of indicators to measure and monitor innovation performance, ranging from technological innovation to other forms of innovation, e.g., public sector innovation" as stated from the European Commission [4]. Technology transfer activities take place in quite different ways, depending on traditions, resources and economic structures, but always depending on the active commitment and participation of the involved actors within their organizational structures. Conventionally, innovation capacity and *KTT* is treated as a black box that is studied through comparison of its inputs and outputs using statistical data. Instead, the approach we describe here is a "white box" approach, i.e. it is an attempt to dissect innovation and technology transfer activities into its individual processes and performance descriptions. *KTT* activities that are a priori complex, creative at some extent, and organization-dependent, are expressed here in generic, process-oriented terms. *innoSPICE* can be used together with the existing capability measurement framework of *ISO/IEC 15504* to assess organizations involved in *KTT* activities. Technology development, transfer and innovation activities are analyzed by means of process quality and the depending organizational capabilities. It uses the *ISO/IEC 15504* framework for assessing the organizational process dimension of stakeholders within the complete innovation value cycle. The assessment methodology is described in greater detail in the next section. These assessments provide a structured feedback of the current transfer performance: Understanding the activities in the context of the complex transfer procedure helps to identify organizational strength and weaknesses on the detail level. A variety of 52 processes with quality and numeric indicators relevant for the process performance is addressed. Therefore, on an abstract level, it defines three roles that are involved in the transfer and innovation chain along with their specific activities (see Table 1):

Table 1. Roles in the technology transfer process

Roles	*Activities*
Technology Developer	Its activity is to develop new technologies/concepts/knowledge.
Technology Transfer Driver	Its activity is to drive technology commercialization from the technology developers to the technology recipients/acquirers (innovators).
Technology Recipients/Acquirer and Innovator	Its activities are: • identifying and selecting innovations to be introduced • acquiring new technology to be introduced into the organisation's products and/or services to perform the innovation • introducing innovations into products, services and processes

Moving from abstraction towards real activities, organizations often perform more than one role in the innovation and technology transfer process. Likewise, one role can be performed by more than one organization. The goal is to ensure that important organizational processes are dedicated to *KTT* as an added value to the core business. This will support the organizations in the following way:

- Self-reflection of their own transfer and innovation capability in the context of the complete set of *KTT* and innovation processes.
- Pointing out obvious and problematic differences between actual and expected capability levels.
- Better understanding of the technology transfer and innovation landscape and their related key concepts.
- Sharing knowledge and experience on technology transfer and innovation in a structured and standardized way.
- Pointing out preliminary possible paths for technology and innovation capability improvement.

Planned future extensions of *innoSPICE* applications are:

- Objective inquiry of the transfer capability; performance indicators must be established.
- Objective inquiry of the transfer capability of regional technology transfer systems.

- Determination of the maturity level of a transfer institution. This requires developing and evaluating a metric of the capability levels.

innoSPICE is part of a quality management system for innovation, knowledge- and technology transfer and comes along with important improvements within the innovation cycle. Processes are, according to *ISO/IEC 15504,* represented by a process purpose, a number of outcomes and a number of base practices. As foreseen, *innoSPICE* groups processes into *Organizational, Primary* and *Support process categories*. For the adoption into innovation, knowledge and technology transfer, the Primary process category is split into subcategories that are structured according to the aforementioned roles.

3 Assessment Methodology for Guided Self-assessment KTT Capability

“All quality management approaches or standards recommend the continual improvement of products and services offered to customers through periodic assessment of (...) the quality and maturity of the product or service itself, as well as of the processes behind them” [1]. In that sense, KTT should have been for quite a long time in the focus of QM approaches since it addresses improvements in the services and products. But in contrast to the corporate strategy development in which the core objectives are defined based on economic indicators, quality management in public research institutions needs to be adapted to the interlinking structures of research, teaching, and transfer ambitions. Up to now concepts that enable the proactive management (not retrospective ranking!) of the relevant organizational processes of this triad were missing.

An assessment of these processes is therefore the very first step of such an improvement project (Deming - PLAN), "*and not only part of the CHECK step as it can serve to set measurable goals of improvement and help to prioritize them.*" [1]. Following this concept, the authors employed a guided self-assessment approach during innoSPICE assessments. Partially aberrant from the guidelines for process assessment of the ISO, this approach combines lessons learned from process consultation with the well-established assessment by an external assessor (team). An assessment report following ISO is prepared afterwards with recommendations for process capability improvement.

A theoretically well founded and practically established concept of coaching and supervision is the so called *systemic consultation*, which builds on systems theory [13]: "Systemic consultation intends to support clients in the handling of problems by gathering a reasonable problem perspective" [12]. The focus of the intervention is a so-called "irritation" of the client [13], with a more positive understanding than the colloquial one. Via irritation, the client system becomes encouraged to dissociate from well-established patterns of perception and discover its changeability. The dissociation enables not only self reflection following [12], but moreover the reflection of

- functional aspects, such as the initiation of new procedures,
- methodological aspects, such as structuring of information and decision making,
- social aspects, such as the cooperation and consensus-building in the team and
- aspects of personal skills, because motivational and change aspects are mostly addressed in connection with change processes and improve the action and responsiveness.

The presented form of innoSPICE assessments is following this approach. The organizational activities, their patterns of perception and dynamic are structured in base practices of the related innoSPICE processes. The main advantage mentioned for initiating an internal improvement project in the assessed public research organizations was the shift from an auditing situation to a learning situation within the organizations. The assessor changes his role and becomes a consultant. For half a century, the role of a "consultant" in organizational development is generally defined as a voluntary relationship between a professional supporter (consultant) and a needy system (client) in which the consultant tries to guide the client in solving current and potential problems. The relationship is regarded by both parties as fixed-term [10]. In addition, the consultant is external, i.e. he is not part of the hierarchical system of the client [9].

To see the auditor as a consultant means that he gets involved in the problem-solving capacity of the client system. Via the assessment of current process performance, he supports identifying and solving the organizational challenges of the client. Thereby the performance increase of the organization and the individual development of the involved persons are considered as equal objectives. As organizations change only by the actions of its employees, as far as possible, all those involved in problem determination and resolution should be part of the assessment. In doing so, the client becomes an expert in the content determination of a well-structured problem and a potential solution, while the consultant accompanies the client in activities-related interventions.

As for supervision and systemic consultation, numerous evaluations (e.g. [2]) document the development and increase of professional cooperation via the methodological approach. Following [5] *evaluations show in detail, (...) that during the supervision process experience, cognitions, behaviors and consciousness as essential elements of the employment, the organization and the working environment might change.* The authors applied a comparable evaluation setting to the chosen approach of guided self assessments and got comparable results of validity.

Based on this knowledge about supervision, the innoSPICE guided self-assessment supports the organization by checking beside the concrete base-practices the adequacy of the organizations' own perception in terms of the professional activity, the current situation of the team and/or the organization. If appropriate, it can it help to change them. In [5] it is stated that *"supervision is knowledge as well as a learning process, providing assistance in uncertainties and exposure to the increase in complexity and diversity of - possibly conflicting - work items."* It is this statement that we refer to for the methodological implementation of innoSPICE guided self assessments:

Step1	The concepts of process capability modeling and assessment are described according to ISO/IEC 15504.
Step2	The key concepts of technology transfer and innovation are presented and discussed, including process and organization categories and innovation and transfer cycle.
Step3	Organization members describe their organization work, structure, role and responsibilities in technology transfer and innovation.
Step4	The scope of the assessment is agreed upon, i.e., which processes will be considered and which part of the organization will be evaluated.
Step5	For each considered process, the process and its scope are explained at a generic level. Concrete examples of concrete activities related to this process can be provided. Purpose and outcomes of the process are provided.
Step6	For each considered base-practice, • its scope is explained at a generic level by the assessors. Concrete examples can be provided. • organization members and assessors decide whether or not the base-practice has to be considered, i.e., if it is applicable to the organization or not. • the concrete activities within the organization are analyzed. This step is performed by applying a participative approach by the personnel using daily-work wording and language. This task is crucial because the type of relevant activities is usually not documented. The challenge for the assessor is to clarify the base-practice in making it adoptable for the assessed organization. Guiding the organization towards a correct interpretation of the base-practice, which is relatively abstract by nature, requires a strong understanding of the underlying model and experience in technology transfer and innovation as well as supervision/ coaching skills. • Processes capability is assessed as capability level 1, i.e., to which extent the process is achieved. Institution's members start proposing a process capability rating in a participative way and stay to be the owners of the assessment process. During this reflection, organization members rate, according to their opinion, the extent with which the base practice is achieved and start reflecting on what could be done to improve its achievement. This participative approach enables each team member to understand his and related stockholder's concrete roles with respect to the base practice, i.e., the concrete bottlenecks which impede KTT and innovation within the organization. Finally, the client system reaches a consensus about the actual state of the KTT and innovation activities in the organization and draws preliminary possible paths for capability improvement.
Step7	Feedback

4 Assessment Results

In this section the results of the assessments that have been performed during the European Project BONITA [14] with the *innoSPICE* process model will be discussed. A synthesis of the results of fourteen assessments is presented. Capability level 1 has been considered to know to what extent technology development, transfer and innovation processes are performed and process goal and process outcomes are achieved. Only Primary processes are presented. For the sake of clarity, only groups of processes which share a common meaning are considered. The process groups of *innoSPICE* are briefly presented in the following:

- **Project Proposal:** Prepare R&D scientific project proposal (D.Project);
- **Knowledge Creation:** Create basic and applied science knowledge (D.KW);
- **Prototyping:** Develop prototypes (D.Proto);
- **Technology Development:** Develop a new technology related solution/ product (D.Tech);
- **Technology Release and Support:** Provision of the developed technology and initial customer support (D.RelSup).
- **Technology Transfer Concept:** Understand available knowledge/ technologies and needs for developing a technology transfer concept (T.Concept);
- **Technology Analysis:** Analyze the technical aspects of technologies (T.TechAnal);
- **Technology Value Evaluation:** Evaluate the (acquirer's relative) value of a technology (T.TechVal);
- **Technology Transfer Decision:** Decide whether or not to go to market according to market opportunities (T.Dec);
- **Technology Transfer Go-to-Market:** Identify the best route to go-to-market (T.Market);
- **Technology Transfer Financing:** Raise financing for commercialization (T.Finan).
- **Innovation Preparation:** Identify and define improvement needs and requirements for the organization and select technologies to be introduced to perform the innovation. (I.Crea);
- **Innovation Deployment:** Ensure preconditions that the technology deployment will be successful and successfully deploy the introduced technology into its intended environment. (I.Deploy);
- **Innovation Management:** Manage the innovation within organization to ensure easy adoption of further technology improvements. (I.Mana).

4.1 Examples of Assessment Results

To illustrate the application of *innoSPICE,* three examples of assessment results will be discussed in this section. The process capability rating (from 0 to 100 according to [6] to *ISO/IEC 15504-2*) is provided for each process group. The process capability rating is the percentage scale representing extent of achievement of the process category. It is computed as the average of the ratings of the processes belonging to the process group.

The first organization (see figure 1) is a of pure public research organization which mainly, if not only, focuses on basic knowledge creation. The assessment results confirm it. The knowledge creation process category (D.KW) capability rating is the highest one. Prototyping (D.Proto) and technology development (D.Tech) processes are achieved to a lower extent, which confirms that this organization is related more to basic science than prototype and technology development. Technology transfer related processes are only partially achieved except for one, which means that the organization does not perform much of scientific result commercialization and tends to exploit them only with scientific publications.

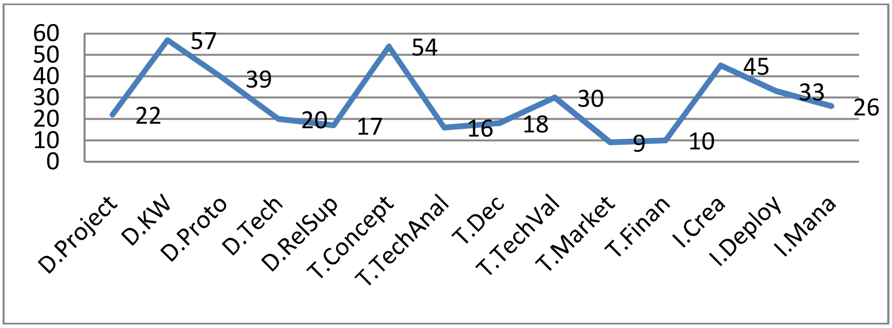


Fig. 1. Pure public research organization capability assessment

The second organization (see Figure 2) is a public technology development organization which manages at the same time to develop basic science knowledge and to closely work with companies, providing them with under-development innovative technologies that have not yet reached the market stage. From the assessment results, we can see that this organization effectively develops both basic knowledge and prototypes and to a lower extent, technologies. Technology release (D.RelSup) and support process capability ratings are higher than that of the knowledge creation organization. Interestingly, technology analysis (T.TechAnal) and technology transfer concept (T.Concept) process categories are achieved at a high level which is not the case for technology transfer decision (T.Dec), technology value evaluation (T.TechVal) and technology transfer go-to-market (T.Market) process categories. Indeed, this organization is able to understand the concepts of technology transfer and evaluate the technical value of their technologies but does not have the competencies to understand the market. Also, it is up to the company that acquires the technology to decide whether or not the technology is brought to the market and how to finance the commercialization. It is obvious from the assessment results that this organization acts as a prototype developer that takes into consideration the possible applications of its developed technologies.

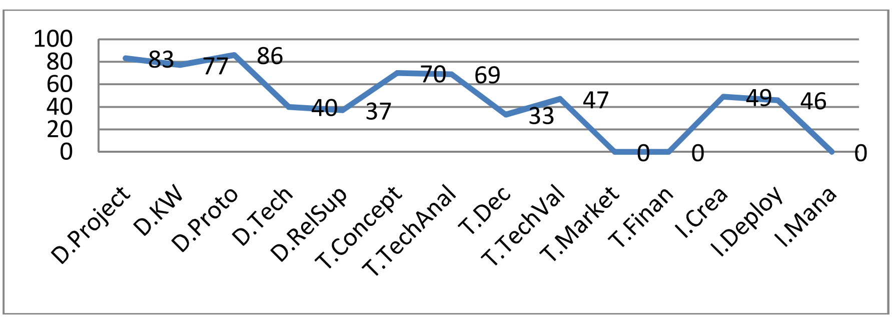


Fig. 2. Public technology developer organization capability ratings

The third organization (see Figure 3) is a private organization that develops and commercializes technologies that were mainly developed in research institutes. This organization is deeply involved in the three faces of technology: development, transfer and innovation. The assessment results confirm it. Technology development (D.Tech) is rated higher than prototype development (D.Proto) which is also rated higher than knowledge creation (D.KW). Indeed, the main goal of this private organization is to commercialize technologies and thus tend to focus more on developing technologies rather than basic knowledge. Contrary to the two first examples, this organization deals with all the aspects of technology transfer, especially, technology transfer decision (T.Dec) and technology transfer financing (T.Finan). They are the true owners of the technology transfer activities because they both own the technologies and make profit out of it. We can also see that this organization performs innovation processes at a higher level. Indeed, a common intuition is that private organizations are much more concerned by continuously improving their own product (good or service), process, marketing or organizational methods, especially when considering innovation management.

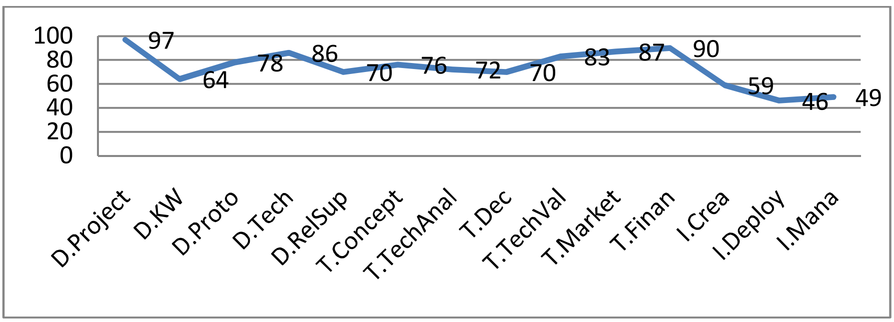


Fig. 3. Private technology developer and transferor organization capability ratings

4.2 Profiles

From the assessments results, we have drawn process capability profiles. Profiles are patterns that provide the ideal capability ratings that an organization of a given category should ideally reach. An organization category defines the role of an organization in technology transfer and innovation. In a profile, a rating indicates to what extent a process category is important for an organization of that given type. For example, one can ideally expect that organizations have a high capability level for and only for processes that are part of their core business. This way designed profiles serve as a master-shape for capability diagrams.

Two types of profiles have been defined: pure profiles and target profiles. A pure profile defines an abstract profile of a pure organization. A so-called pure organization is an organization that performs one and only one aspect of technology development, technology transfer or innovation. A target profile is the set of capability levels that a real organization should reach for a given role in transfer. Figure 4 and 5 present respectively the pure and target profiles.

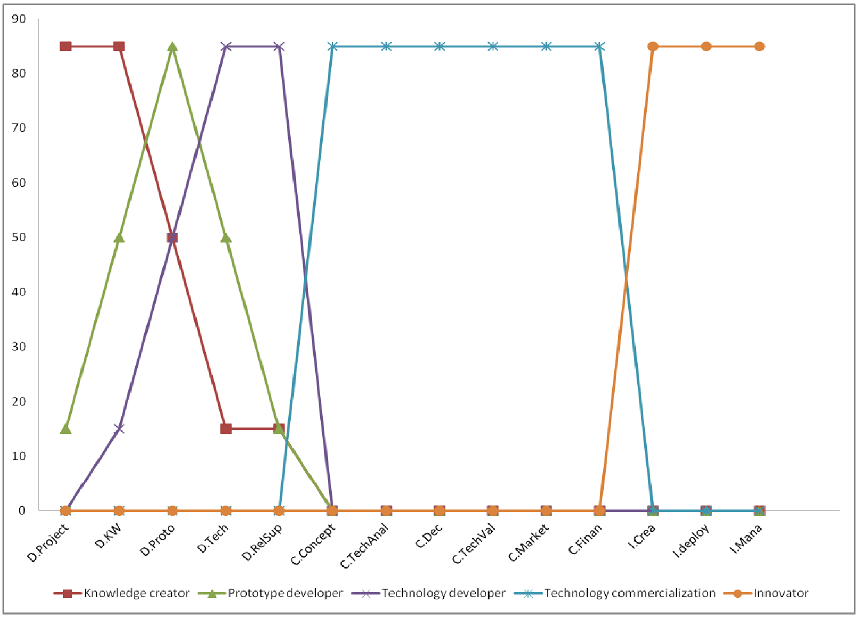


Fig. 4. Pure profile

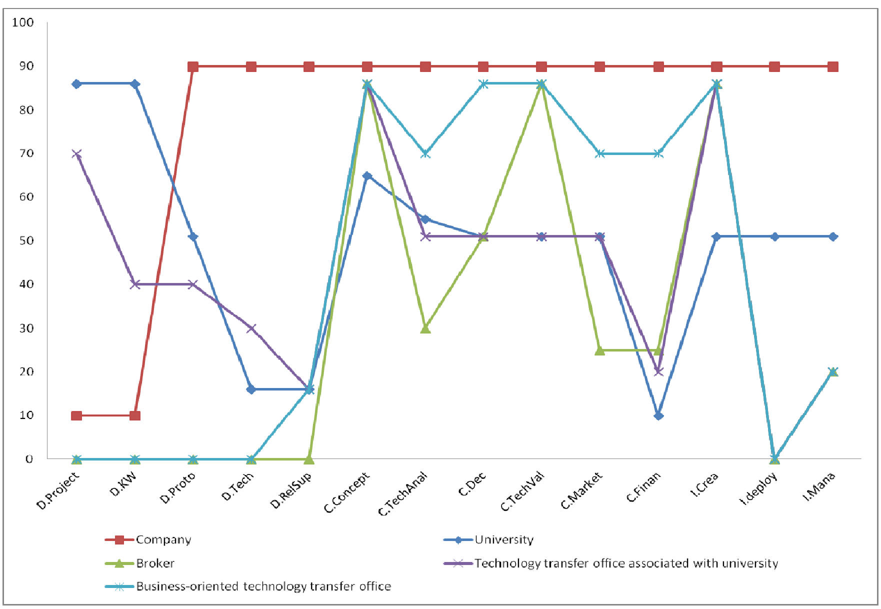


Fig. 5. Target profiles

With these profiles, one could spot the role of an organization more easily. With the introduction of categories as an additional layer on top of processes, the complexity was reduced from 53 processes (vectors) down to 14 attributes by clustering. The combination of the original processes to categories allows the application of weights to them. Though this was not yet performed, we are sure that this will lead to a more differentiated analysis without increasing complexity again. The target profiles can be used as a blueprint for a specific target in organization development. Once designed, one can apply this master-shape to a given assessment to check the state of the organization's development. With this approach a multi-level analysis [11] from base-practices, processes and categories could be applied to assessment ratings of complex organizations.

5 Conclusion and Outlook

The *ISO/IEC 15504* standard based model *innoSPICE* provides a new application area for the SPICE standard. Improving KTT and innovation is of high interest for research organizations and governments. Beside several benefits like described in [2], *innoSPICE* can, e.g., help turn research organizations into knowledge suppliers by applying similar mechanisms, which for years now have been established in the industry for technological suppliers.. Additionally, assessing the process capabilities in the field of *KTT* enables and introduces structured process improvements in public sector organizations that are not typical users of such methods.

The model was developed and evaluated within the Baltic Sea Region INTERREG 4B project BONITA with practitioners and experts in *KTT* domain and the assessments were performed as guided self-assessments. Partially aberrant from the guidelines for process assessment of the ISO 15504-3 [7], this approach combines lessons-learned from supervision and process consultation with the established SPICE assessment. Referring to those two domains, the innoSPICE guided self assessment supports by checking beside the concrete base-practices the adequacy of own perception in terms of the professional activity, the current situation of the team and/or the organization. If appropriate, it can it help to change them. On one hand, deviating from the standard way of performing the assessments necessitates new requirements for the assessors. Additional competence is required for *KTT* domain and the assessment process itself. On the other hand, it is expected that reassessments of organizations will converge to the *ISO/IEC 15504* standard assessment when the knowledge of the need for particular *KTT* processes is persistent in the organization. Currently a study is in preparation to compare the innoSPICE guided self assessment approach with, e.g., an integration of *innoSPICE* in the *TIPA* framework [1], and to evaluate *ISO/IEC15504* conformant innovation and *KTT* process capability assessments.

The analysis of the assessment results presented in this paper showed clearly specific characteristics within different types of KTT related organizations. Based on these results, typical capability profiles were extracted, that can serve as a reference profile for a follow-up process improvement.

Further developments of *innoSPICE* will be on different levels. On scientific level, e.g., suitable performance indicators have to be identified and validated to increase the objectiveness of the assessments and to allow certification of *KTT* capabilities by external assessors. On practical level, the implementation of *innoSPICE* in knowledge intense institutions has to continue raising the practical experience of assessments in the *KTT* domain. This will also help to increase the awareness of process capability assessments and improvement in public sector organizations.

Acknowledgements. The work presented here is result of the BONITA project and was part financed by the European Union (European Regional Development Fund) INTERREG 4B Baltic Sea Region Programme 2007 – 2013.

References

1. Barafort, B., Betry, V., Cortina, S., Picard, M., St-Jean, M., Renault, A., Valdes, O.: ITSM Process Assessment Supporting ITIL. Van Haren Publishing, Zaltbommel (2009) ISBN: 9789087535643
2. Berufsverband für Supervision und Organisationsberatung. Beratung messen – Wirkung sichtbar machen. BSO-Journal (January 2006)
3. Besson, J., Woronowicz, T., Mitasiunas, A., Boronowsky, M.: Innovation, Knowledge- and Technology Transfer Process Capability Maturity Model - innoSPICE^{TM}. In: Mas, A., et al. (eds.) SPICE 2012. CCIS, vol. 290. Springer, Heidelberg (2012)
4. Communication from the Commission to the European Parliament, the Council, the European Economic Committee and the Committee of the Regions. Europe 2020 Flagship Initiative Innovation Union, SEC 1161 (2010)
5. Hausinger, B.: Der Nutzen von Supervision. In: DGSv (Hrsg.): Der Nutzen von Supervision. Verzeichnis wissenschaftlicher Arbeiten, pp. S.5–S.7. DGSv, Köln (2006)
6. ISO/IEC 15504-2. Information Technology – Process Assessment – Part 2: Performing an Assessment. International Standards Organization (2003)
7. ISO/IEC 15504-3. Information Technology – Process Assessment – Part 3: Guidance on performing an assessment. International Standards Organization (2004)
8. Performing an Assessment. International Standards Organization
9. Jonas, E., Kauffeld, S., Frey, D.: Psychologie der Beratung. In: von Rosenstiel, L., Frey, D. (Hrsg.) Enzyklopädie der Psychologie. Hogrefe, Göttingen (2007)
10. Lippitt, R.: Dimensions of the consultant's job. Journal of Social Issues 15, 5–12 (1959)
11. Snijders, T.A.B., Bosker, R.J.: Multilevel analysis: an introduction to basic and advanced multilevel modeling. Sage Publishin Ltd. (1999)
12. Walger, G., Scheller, C.: Das Angebot der Unternehmensberatung in Deutschland, Österreich und der Schweiz. Eine empirische Analyse. QUEM-report, Heft 54 (1998)
13. Wimmer, R.: Organisationsberatung – Eine Wachstumsbranche ohne professionelles Selbstverständnis. In: Hofmann, M. (Hrsg.) Theorie und Praxis der Unternehmensberatung. Bestandsaufnahme und Entwicklungsperspektiven, p. 45. Physica, Heidelberg (1991)
14. Woronowicz, T., Mitasiunas, A., Boronowsky, M.: BONITA - Improve Transfer From Universities for Regional Development. In: Proceedings of the 3rd ISPIM Innovation Symposium, Quebec City, Canada (December 2010) ISBN 978-952-265-004-7

Software Process Improvement Health Checklist

Natalja Nikitina and Mira Kajko-Mattsson

School of Information and Communication Technology
KTH Royal Institute of Technology
Stockholm, Sweden
{nikitina,mekm2}@kth.se

Abstract. Positive results achieved by software process improvement (SPI) efforts tend to degrade with time. To prevent degradation, companies need to regularly evaluate their SPI efforts and tackle potential SPI problems in a timely manner. This can be done by evaluating the conditions necessary for succeeding with the SPI implementations and sustaining their results. In this paper, we suggest an *SPI Health Checklist* to be used for diagnosing the health of the SPI efforts.

Keywords: SPI, success, effort, evaluation, sustainability.

1 Introduction

Software process improvement (SPI) efforts often lead to immediate positive results, which are many times hard to sustain in the long run [1]. Positive SPI results may degrade with time due to many reasons. One of them is the fact that interest and effort dedicated to SPI decrease with time and so does the quality of the SPI effort [2].

To be successful with SPI, organizations need guidance not only on how to improve their development processes but also on how to upkeep the quality of their SPI efforts [1]. Such guidance would help them to regularly evaluate the health of their SPI processes, identify their bottlenecks and attend to them, and thereby, it would prevent the SPI efforts from degrading. To the knowledge of the authors of this paper, there is no published guidance on how to sustain the good quality of the SPI efforts.

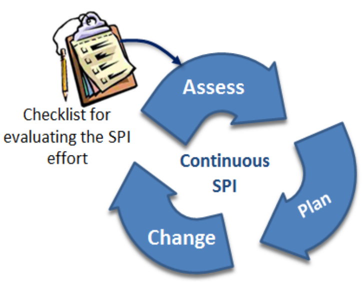
In this paper, we suggest an *SPI Health Checklist.* As illustrated on the left side of Figure 1, the checklist is intended to be used for evaluating the quality of the SPI efforts during continuous process improvement. It consists of forty two checklist items representing the properties (attributes) of healthy SPI efforts. By evaluating the states of those attributes, software companies may identify problems with their SPI efforts and find their underlying causes. The forty two checklist items (SPI health properties) have been evaluated within fifteen software development companies.

The remainder of this paper is as follows. Section 2 presents our research method. Section 3 presents the checklist and its items. Section 4 reports on the results of the checklist evaluation, and finally, Section 5 presents conclusions and future work.

2 Research Method

Our research method consisted of six steps: (1) *Literature study*, (2) *Checklist design*, (3) *Questionnaire design*, (4) *Pilot interviews*, (5) *Interviews*, (6) *Analysis and*

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 85–96, 2012.



1. Present the clusters of the SPI health checklist
2. For each cluster do the following:
   a. Present the cluster.
   b. For each checklist item ask the following:
      i. Is this checklist item significant for SPI? On the scale 1-5, please grade the significance of this checklist item for the SPI success? Please motivate your grading
   c. Are there any checklist items that are missing in this cluster? If yes, please list the missing ones.
3. Are there any clusters that are missing in this framework? If yes, please list the missing ones.
4. Do you think the *SPI health checklist* can be useful for the SPI? Please motivate your answer.
5. Do you have any suggestions for how one could evaluate the success of the SPI with the use of the *SPI health checklist*? If yes, please describe it.

Fig. 1. Placing the SPI Health Checklist in Continuous SPI and Interview Questionnaire

checklist improvement. In the first step, *Literature study*, we have studied [1-15] in order to identify conditions contributing to or subtracting from the success of the SPI efforts. As a basis for this step, we have used the SPI success factors that we have published in [3]. Then in the *Checklist design* step we transformed those factors into the checklist items and grouped them into seven clusters.

As a third step, we created an interview questionnaire. The questionnaire is presented on the right side of Figure 1. It investigates the significance and completeness of the checklist items. The significance was graded on a scale from 1 to 5, where 1 meant that the checklist item did not contribute to the SPI success, and 5 meant that the checklist item had a very high impact on SPI and might even fail the project.

As a next step, we conducted two pilot interviews during which we tested the clarity and relevance of both the checklist items and the questionnaire. Our goal was to improve the smoothness and quality of both the questionnaire and checklist by eliminating all kinds of problems, incomprehensibilities, and duplications before starting collecting data from the target population. This step had resulted in an improved questionnaire and some minor changes to the checklist.

During the *Interviews* step, we interviewed seventeen software engineers that were involved in SPI or were affected by it. We interviewed ten engineers and delegated interviews with seven another engineers to students on a master level. Altogether, our interviewees came from fifteen software companies involving six engineers, six technical managers, three SPI managers and two consultants.

Finally, during the *Analysis and checklist improvement* step, we transcribed the interviews and analyzed their results using the hermeneutics approach [16]. The answers were grouped and analyzed. They provided feedback for either the creation of new or reformulation of the existing checklist items.

3 Checklist

The *SPI Health Checklist* includes items indicating the properties (attributes) of healthy SPI efforts. As shown in Figure 2, they are grouped into seven clusters: *Software process, SPI process, Process tools, Organization, Stakeholders, Process*

1. Software process
1.1 Software process is well defined
1.2 Software process is transparent
1.3 Software process is frequently monitored and measured
1.4 Software process is frequently reviewed
1.5 **NEW:** Software process quality is being improved

2. SPI process
2.1 SPI goals and vision are clear and realistic
2.2 SPI goals and vision are aligned with business goals and strategies
2.3 **NEW:** SPI method and strategy is defined
2.4 SPI process is tailored to fit the organization
2.5 SPI focuses on one or a few goals at a time
2.6 SPI activities are implemented in a priority order
2.7 SPI related information is disseminated
2.8 SPI process is effectively managed
2.9 The results of SPI are continuously measured
2.10 SPI leads to the expected results
2.11 **NEW:** SPI process is frequently reviewed and improved, if needed
2.12 **NEW:** SPI goals and vision are frequently audited and improved, if needed

3. Process Tools
3.1 Infrastructure to support software process and SPI is in place
3.2 Mechanisms to control the adherence to the software process are in place

4. Organization
4.1 Management understands the long-term ROI of SPI
4.2 Management commits to support and sponsor the SPI effort
4.3 Resources are assigned for SPI activities
4.4 Culture is considered in SPI

5. Stakeholders
5.1 SPI roles and responsibilities are defined and clear for each stakeholder
5.2 Accountability for software process is assigned
5.3 **NEW:** Accountability for SPI is assigned
5.4 Stakeholders reach consensus on SPI goals and vision
5.5 Stakeholder feedback is considered in SPI
5.6 Technical staff participates in SPI
5.7 Stakeholders follow the software process
5.8 Stakeholders have realistic expectations from SPI

6. Process Literacy
6.1 Stakeholders understand the software process
6.2 Stakeholders have been trained in software process
6.3 Stakeholders understand the extent and purpose of the SPI activities
6.4 SPI leaders possess knowledge and competence to lead SPI

7. Social Factors
7.1 Process discipline is adopted by all the stakeholders
7.2 Technical staff is committed to continuous SPI
7.3 Technical staff accepts the planned SPI activities
7.4 Technical staff owns the software process
7.5 **NEW:** SPI leaders are passionate about SPI
7.6 Stakeholders trust and respect SPI leaders
7.7 Stakeholders are rewarded for their contribution to SPI success

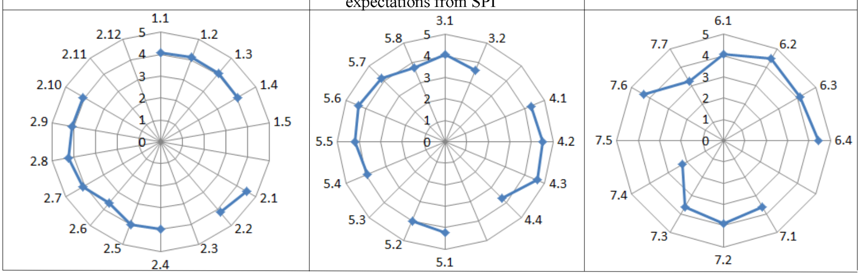


Fig. 2. SPI Health Checklist

literacy and *Social factors*. Each checklist item is placed in one cluster only. The choice of its parental cluster was based on the item's degree of impact on the cluster. For instance, the item *Technical staff owns the software process*, may be relevant in three clusters such as *Software process*, *Social factors* and *Stakeholders*. However, it has the highest impact on staff's affection to the process. Hence, we place it in the *Social factors* cluster. Below, we present the checklist clusters and their inherent items.

3.1 Software Process

The *Software process* cluster covers a portfolio of processes that undergo the improvement. It includes a list of software development processes, methods and standards in an organization. It includes the following items:

1) Software process is well defined: All process phases, activities, inputs and outputs are identified, established and to some extent documented. Well defined software processes facilitate process participation, understanding and communication [2]. They are easier to understand, analyze, and improve, and therefore, process defects are easier to detect.

2) Software process is transparent: All process phases, activities, roles involved, information managed and outcomes are clearly visible. This aids in identifying the process strengths and shortcomings [2].
3) Software process is frequently monitored and measured: This provides indispensible data for reviewing the process, and defining or correcting the SPI goals, vision and strategy [2-8]. SPI cannot be regarded as successful unless evidence is provided.
4) Software process is frequently reviewed: A software process should be continuously reviewed and reflected upon. Process reviews help to identify process problems and their solutions, acknowledge the SPI benefits, determine future SPI activities, learn from previous experience, and thereby, they contribute to a self-driven continuous process improvement [2-4]. Process reviews ensure the continuity of the SPI efforts [5-7].
5) Software process quality is being improved: Tangible results showing evidence of process improvement indicate healthy SPI effort [3][5-6].

3.2 SPI Process

The *SPI process* cluster includes a portfolio of processes and recommendations that are used for improving software processes. SPI processes include methods and techniques for SPI initiation, preparation, planning, management, execution and the evaluation of its results. The *SPI process* cluster includes the following checklist items:

1) SPI Goals and Vision Are Clear and Realistic: They make the SPI focus on relevant improvement tasks thus contributing to the structured management of the SPI effort. They add to better understanding of the SPI process, improved motivation and engagement, and thereby, to realistic expectations. Lack of them misguides people in their SPI efforts, and, may cause frustration and decrease their motivation towards SPI. [1-2][5-6][9-10]
2) SPI Goals and Vision Are Aligned with Business Goals and Strategies: This makes software process productive and the organization prosperous. It helps in getting management support for the SPI effort. Lack of alignment can fail the SPI efforts [2-3] [5-6][8-9][11].
3) SPI Method and Strategy is Defined: SPI is a complex and time consuming process. It demands a clear and well defined SPI method and strategy that are suitable for the organizational specifics and needs [3][12-13]. Otherwise, the SPI effort is deemed to fail.
4) SPI Process is Tailored to Fit the Organization: When defining an SPI process, one should consider the contextual specifics of the organizations such as size, culture, policies, needs and experience. Not considering the organizational context may result in a non-efficiency of the SPI process, and thereby, a waste of time and resources [1][3] [5][11].
5) SPI Focuses on One or a Few Goals at a Time: Being overwhelmed with the great amount of suggested improvement goals, the organizations should focus on just a subset of them. This leads to easier and more efficient implementation of SPI, better controlled measurement of its results, and thereby, a more successful SPI effort.[3]

6) SPI Activities Are Implemented in a Priority Order: By prioritizing the SPI activities according to their criticality and urgency, one assures that the most benefitial SPI activities get implemented first [3]. If the SPI activities are implemented spontaneously and randomly, they may lead towards unexpected and unpleasant consequences.

7) SPI Related Information is Disseminated: The SPI changes have to be communicated to all the stakeholders involved [1][3][6][8-9]. This helps to avoid misunderstandings of the SPI activities, confused personal, wrong SPI implementation, or resistance to change.

8) SPI Process is Effectively Managed: As any other project SPI should be managed in a prudent and efficient manner so that its resources are used in a wise and sustainable manner [2][13]. Without proper management, the SPI efforts are likely to fail [1].

9) The Results of SPI Are Continuously Measured: This provides evidence on the positive or negative SPI results and aids in evaluating the adequacy of the current SPI goals and the suitability of the current SPI process [2][4][10]. Moreover, measured and acknowledged SPI results increase team moral, motivation and engagement.

10) SPI Leads to the Expected Results: SPI goals and objectives predefine a set the expectations to be fulfilled from the SPI effort. Reaching the expected results provides evidence that the SPI expectations have been met. It shows that SPI was worth the effort and aids in continuing and boosting motivation and engagement towards SPI. [3]

11) SPI process is Frequently Reviewed and Improved, if Needed: Since the organization, its needs and structure change over time so do the SPI vision and goals. To address those changes, the SPI process should be frequently reviewed, and adjusted, if needed [4-7].

12) SPI Goals and Vision Are Frequently Audited and Improved, if Needed: They should accurately represent the current needs of the organization and reflect the key problematic areas [4-5][7][9]. Therefore, as once set, the SPI goals and vision should be improved when needed.

3.3 Process Tools

The *Process tools* cluster includes a list of mechanisms supporting the software process and the SPI process. It includes the following checklist items:

1) Infrastructure to support software process and SPI is in place: This contributes to the successful process implementation and improvement, and effective communication of process related issues [2][8].

2) Mechanisms to control the adherence to the software process are in place: Lacking either dedication or commitment to the new process, people may be tempted not to follow it [2]. The optimal solution is that the staff has some space of freedom defined for following the processes so that their work does not become too much prescribed. Still however, to guarantee that the critical process steps are followed, organizations should impose some control of process adherence [3-4].

3.4 Organization

The *Organization* cluster comprises a list of organizational properties that impact the direction and realization of SPI. It includes the following checklist items:

1) Management Understands the Long Term ROI (Return on Investment) of SPI: It is critical that the managers and the SPI sponsors are aware of the cost and long-termedness of the SPI results [2][5-7][9][11]. Unawareness may lead to the choice and approval of ineffective or short-term SPI steps, and thereby, to loss of valuable resources.
2) Management Commits to Support and Sponsor the SPI Effort: In order to provide long lasting results, the management must be strongly committed to process improvement and continuously sponsor it with resources in form of time, effort, and other [1-8][10-13]. This helps retain high priority of SPI and assure continuous supply of the required resources.
3) Resources Are Assigned for SPI Activities: SPI cannot run on its own. It requires the resources to be dedicated and assigned to it [3][7-8][10-13]. Otherwise, it will fail, it may be discontinued, or at its worst, it may be terminated.
4) Culture is Considered in SPI: Cultural aspects such as knowledge, beliefs, behavior, social norms that are commonly shared by individuals in software organizations, strongly affect ways of working and decision making, capacity to learn and capability to make changes. Both national and organizational culture may influence attitudes towards process changes, ways of choosing SPI actions, and ways of communicating on SPI results. [9]

3.5 Stakeholders

The *Stakeholders* cluster lists the responsibilities of the roles that are involved in or affected by SPI. By a stakeholder, we mean a person, a group or an organization that affects or is affected by an SPI effort. Stakeholders include large variety of roles, which, due to space restrictions, cannot be listed in this paper. The role of a *technical staff,* however, requires additional clarification. By *technical staff*, we mean a stakeholder who executes the development process. The *Stakeholders* cluster includes the following items:

1) SPI Roles and Responsibilities Are Defined and Clear for Each Stakeholder: This helps in evaluating the SPI process, identifying its problems, and ensuring that all the required SPI activities are conducted. Without clearly defined SPI roles and responsibilities, there may be confusion upon who is responsible for what, and as a result, the SPI may be delayed, neglected, done in a rush, or even forgotten [1-3][5-6][8-10].
2) Accountability for Software Process is Assigned: Software process quality should be respected and prioritized. Therefore, there should be a role that is accountable for software processes, with ultimate responsibility for defining, evaluating and monitoring them [2][11]. Without accountability, software processes are doomed to degrade.
3) Accountability for SPI is Assigned: Without assigned accountability for SPI, the SPI effort will most likely not be long lasting [7][10][14].

4) Stakeholders Reach Consensus on SPI Goals and Vision: Different stakeholders may have different or even conflicting interests in and objectives with SPI. Therefore, all conflicts and disagreements should be resolved and a consensus should be reached. Otherwise, some stakeholders get uncommited to SPI effort and dissapointed in its results.
5) Stakeholder Feedback is Considered in SPI: Stakeholder feedback helps to identify problems with the process, suggest relevant solutions, and evaluate the SPI effort [5-6].
6) Technical Staff Participates in SPI: Their process knowledge and participation in SPI strongly increases their motivation and commitment towards improvements, reduces resistance to change, and thereby, strongly impacts the SPI success [1-3][5-6][10-11]. Technical staff involvement was found especially important in immature or small organizations [12].
7) Stakeholders Follow the Software Process: Only by following the newly defined and improved process, the organizations may take full advantage of it and the stakeholders may contribute to the same goal and pull in the same direction [2]. If the stakeholders do not follow the process, then the SPI effort will simply be a waste of effort and time.
8) Stakeholders Have Realistic Expectations from SPI: Their expectations should be neither too low nor too high [2-3][11]. If they are too low, SPI will most likely not get enough support from both management and technical staff. If they are too high, then they can lead to disappointment and frustration among the stakeholders, and result in discontinuety of the SPI effort. Realistic expectations, on the other hand, contribute to long lasting engagement and sustainability of the SPI effort [2].

3.6 Process Literacy

The *Process literacy* cluster represents the abilities of the stakeholders to understand, perform and improve software process and the SPI process. Knowledge of the process and continuous process training is essential to SPI. This cluster includes the following items:

1) Stakeholders Understand the Software Process: The understanding of the software process increases stakeholders' commitment to follow it and improves the accuracy of their everyday decisions [1][3][7].
2) Stakeholders Have been Trained in Software Process: Only when the stakeholders are properly trained, mentored and coached in the process, they will likely follow it [2-6] [9] [11-12].
3) Stakeholders Understand the Extent and Purpose of the SPI Activities: Only when the stakeholders involved understand how the SPI will impact and benefit their daily work, they will accept the changes and will implement them properly [1][3][5-6] [10] [12-13].
4) SPI Leaders Possess Knowledge and Competence to Lead SPI: To change not only the process but also deeply ingrained organizational culture, habits, working patterns and manners, SPI leaders need to have knowledge of and experience in SPI. They should also possess good leadership skills and knowledge of their staff so that they are capable of coaching them towards achieving the SPI goals [3-5][7][9] [11-13].

3.7 Social Factors

The *Social factors* cluster covers a portfolio of human behaviors that affect, drive or are triggered by SPI. They indicate people's positive or negative attitudes and reactions towards SPI. This cluster includes the following checklist items:

1) Process Discipline is Adopted by all the Stakeholders: Process discipline defines organization's capability to define, follow and improve their software processes. For the success of SPI, it is important that the organization and all the stakeholders adopt process discipline that is relevant for the context at hand.[2]
2) Technical Staff is Commited to Continuous SPI: Their commitment is the driving wheel of the process improvement [3][11][14-15]. It is especially important in smaller organizations that practice agile development methods and have flat organizational structures.
3) Technical Staff Accepts the Planned SPI Activities: This can decrease inertia to change and assure that the changes are implemented in a uniform and tractable way [2][3][11]. Otherwise, the technical staff would most likely not implement the SPI activities and, thereby, they would not follow the newly suggested process.
4) Technical Staff Owns the Software Process: When the technical staff feel more affiliated with the process, they become more responsible for its improvement [3-5]. Defined process ownership may lead to a build-in and self-driven continuous SPI.
5) SPI Leaders Are Passionate about SPI: Their passion is one of the most important driving forces behind SPI. The SPI leaders should inspire people, engage them to support SPI, recognize their individual efforts and demonstrate the SPI results [4].
6) Stakeholders Trust and Respect SPI Leaders: Even if SPI leaders have a privileged position, they may still have low authority and be mistrusted by the technical staff members [3][10-11]. If so, then their ideas will not be supported by the technical staff and will not be successfully transmitted to process changes.
7) Stakeholders Are Rewarded for Their Contribution to SPI Success: Moral appreciation and financial rewarding acknowledge the individual contributions, and thus engage and motivate people to continue with the SPI effort [2] [4].

4 Results

In this section, we describe the interview results. Due to the space limitations, we only report on the checklist items that were found either controversial or context dependent. The significances of the checklist items are all visualized with the radar charts at the bottom of Figure 2.

4.1 Software Process

All the items in the *Software process* cluster were approved by the interviewees. As shown in Figure 2, they all scored from 4,04 at their lowest to 4,13 at their highest. Regarding the cluster completeness, the interviewees lacked a checklist item communicating continuous improvement of software process quality using different process quality characteristics, such as process efficiency and effectiveness. For this reason, as marked with the flag *New* in Figure 2, we added the checklist item 1.5 *Software process quality is being improved* to our list. In general, the interviewees provided the following comments on the *Software process* checklist items.

- Process transparency (*Checklist Item* (*CI) 1.2* in Fig.2) depends on the accessibility of process information which makes the process be easily seen through and aids in detecting its defects. Both well defined and transparent processes (*CI 1.1* and *CI 1.2*) were claimed to be important for new and less experienced employees.
- Software process should be reviewed on various granularity levels (*CI 1.4*); on the holistic level spanning several processes and on the individual process level. Together with monitoring and measurement, this aids in improving the software process quality.

4.2 SPI Process

All the items in the *SPI process* cluster were approved by the interviewees. They all scored from 3,66 at their lowest to 4,5 at their highest. Regarding the cluster completeness, the interviewees missed (1) the definition of SPI methods and strategies, (2) regular reviews of the SPI process, and (3) audits of SPI goals and vision. For this reason, we added the following checklist items to this cluster: 2.3 *SPI method and strategy is defined* ensuring that the company follows a structured and organized SPI. 2.11 *SPI process is frequently reviewed and improved, if needed,* assuring that the SPI process is suitable for the company, and 2.12 *SPI goals and vision are frequently audited and improved, if needed* assuring that the SPI vision and goals are up to date. Overall, the interviewees had the following comments:

- SPI goals should be clear and realistic (*CI 2.1*), however they are not the determining factors for the SPI success; the process may be improved "on the fly".
- The alignment of SPI goals with business goals (*CI 2.2*) helps to get support of management and sponsorship of the SPI effort. However, rapidly changing business goals may bring disruptions and instability into the SPI effort.
- There were divided opinions about tailoring the SPI to the organization (*CI 2.4*). Some interviewees pointed out that tailoring increase the acceptance of SPI whereas others claimed that it makes the SPI process less flexible and is often undoable, since too many details need to be considered.
- The importance of the order in which the SPI activities are implemented (*CI 2.6*) is significant especially in organizations undergoing large process transformations.
- The SPI related information should be disseminated to all the stakeholders (*CI 2.7*). However it should not contain too many details. It should cover just SPI vision and goals, reasons behind SPI, and future SPI activities.
- Some of the SPI activities imply high risks, and therefore, may not always bring expected results (*CI 2.10*). This should not stop the SPI effort.

4.3 Process Tools

All the items in the *Process tools* cluster were approved by the interviewees. They all scored from 3,6 at their lowest to 4 at their highest. Our interviewees commented only on the significance of the control mechanisms on the success of SPI (*CI 3.2*).

The significance of those control mechanisms depends on the importance of the process standards that they control, and the relevance of those standards to the SPI activities.

4.4 Organization

All the items in the *Organization* cluster were approved by the interviewees. They all scored from 3,69 at their lowest to 4,63 at their highest. In general, the interviewees have pointed out the following:

- Without dedicated resources (*CI 4.3*), SPI efforts will end before even being properly begun. However, there are cases when SPI efforts may succeed without management support or dedicated resources (*CI 4.2* and *4.3*). It is in companies with flat organizational structures where SPI is driven by the technical staff themselves.
- The impact of culture (*CI 4.4*) varies a lot between the organizations. Nevertheless, the consideration of cultural aspects is crucial for the organizations with geographically distributed software development teams or mixed ethnical groups.

4.5 Stakeholders

All the items in the *Stakeholders* cluster were approved by the interviewees. They all scored from 3,72 at their lowest to 4,37 at their highest. Regarding the cluster completeness, the interviewees lacked information on the accountability of the SPI process. They claimed that SPI effort is more successful, if there is a person accountable for it. For this reason, we have added the item *5.3 Accountability for SPI is assigned* to our checklist.

The interviewees mentioned that participation of technical staff in SPI (*CI 5.6*) strongly impacts their motivation and engagement in SPI. Moreover, technical staff can contribute to SPI with deep process knowledge. However, in large organizations both stakeholder feedback and participation of technical staff (*CI 5.5* and *5.6*) only contribute to the increased stakeholders' motivation and provide little or no value for defining the SPI strategy and vision.

4.6 Process Literacy

All the items belonging to the *Process literacy* cluster have been approved by the interviewees. They all scored from 4,07 at their lowest to 4,45 at their highest. The interviewees have provided the following feedback on the process literacy:

- Understanding of the software process and SPI activities (*CI 6.1 and CI 6.3*) is especially important for the technical staff, since it is they who work with the process on a daily basis and it is they who get affected by the process changes the most.
- Competent SPI leaders with strong leadership skills (*CI 6.4*) will most likely lead the company towards better results. The importance of SPI leaders increases in the organizations with hierarchical organizational structure.

4.7 Social Factors

All but one items belonging to the *Social factors* component were approved by the interviewees. The checklist item that had mixed responses was *7.4 Technical staff owns software process*. Its significance was graded as 2,23. All other items belonging to the *Social factors* cluster scored from 3,2 at their lowest to 4,32 at their highest. Regarding the component completeness, the interviewees lacked the enthusiasm of SPI leaders. For this reason, we added the item *7.5 SPI leaders are passionate about SPI* to our list. Overall, the interviewees had the following comments:

- The importance of technical staff commitment (*CI 7.2*) is conditioned by the development method and organizational structure. In agile projects with flat hierarchy, commitment of technical staff is more important.
- The ownership of software process (*CI 7.4*) has generated the most controversial responses. In the strictly agile contexts, process ownership by technical staff can be the key driver of the SPI process. Some interviewees, however, stated that partial ownership of the process by technical staff can be beneficial for SPI in any context. The majority of the interviewees have pointed out that ownership of the process by technical staff may be harmful to SPI. Because the technical staff lacks knowledge and picture of the overall SPI processes, they can make the SPI process unstructured, chaotic and inefficient.
- Rewards (*CI 7.7*) are beneficial for SPI but not essential for its success. Rewards can boost motivation. However, they should not be the main driver of the SPI projects.

5 Conclusions

In this paper, we have presented the *SPI Health Checklist* consisting of forty two items. Its items define the norms necessary for evaluating the conditions of the overall SPI efforts. During this study, we have confirmed the importance and usefulness of the *SPI Health Checklist* within the organizations that are involved in SPI. Except for the item *7.4 Technical staff owns software process,* all the checklist items were considered significant to the SPI success by all the interviewees. This may be clearly seen by the scores that are presented in the radar charts in Figure 2. Almost 70% of the evaluated items scored over 4 out of 5. The most significant checklist items as recognized by the interviewees, were following: • *4.3 Resources are assigned for SPI activities*, • *4.2 Management commits to support and sponsor the SPI effort*, • *2.1 SPI goals and vision are clear and realistic* and • *6.4 SPI leaders possess knowledge and competence to lead SPI.*

The interviewees have stated that the presented SPI checklist can already be used for evaluating SPI efforts. They have also pointed out that the importance of the checklist items varies with organizational contexts. The most influential contexts that they have identified are size, competence and organizational maturity level and structure, software development method, dimensions and scope of SPI, and the SPI initiators (either managers or technical staff). Therefore, companies should tailor the *SPI Health Checklist* before using it.

The goal with the *SPI Health Checklist* is to create a basis for evaluating the health of the SPI efforts. Right now, the checklist is still too general. It only provides support for identifying important features of a healthy SPI process. Despite this, we believe that it may already raise awareness of the state of the SPI efforts. Another goal with the checklist is to supplement the currently defined SPI frameworks with the indicators and alerts of SPI health. To be useful, however, both to the industry and other SPI frameworks or models, the checklist needs to be further evolved and considered in various SPI contexts. This is what we intend to do in our future step. We plan to use the *SPI Health Checklist* for both evaluating the industrial SPI efforts and for validating the checklist.

References

1. Stelzer, D., Mellis, W.: Success Factors of Organizational Change in Software Process Improvement. J. Softw. Process: Improve. Pract. 4, 227–250 (1998)
2. Zahran, S.: Software Process Improvement: Practical Guidelines for Business Success. Addison Wesley (1998)
3. Nikitina, N., Kajko-Mattsson, M.: Success Factors Leading to the Sustainability of Software Process Improvement Efforts. In: 6th International Conference on Software Engineering –Advances, pp. 581–588. IEEE Press (2011)
4. Rainer, A., Hall, T.: Key Success Factors for Implementing Software Process Improvement: A Maturity-based Analysis. J. Syst. Softw. 62, 71–84 (2002)
5. Dybå, T.: An Empirical Investigation of the Key Factors for Success in Software Process Improvement. J. Softw. Eng. 31, 410–424 (2005)
6. Dybå, T.: An Instrument for Measuring the Key Factors of Success in Software Process Improvement. J. Emp. Softw. Eng. 5, 357–390 (2000)
7. Curtis, B., Paulk, M.: Creating a Software Process Improvement Program. J. Inf. Softw. Technol. 35, 381–386 (1993)
8. Sulayman, M., Urquhart, C., Mendes, E., Seidel, S.: Software Process Improvement Success Factors for Small and Medium Web Companies: A Qualitative Study. J. Inf. Softw. Technol. 54, 479–500 (2012)
9. Beecham, S., Hall, T., Rainer, A.: Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis. J. Emp. Softw. Eng. 8, 7–42 (2003)
10. Goldenson, D., Herbsleb, D.: After the Appraisal: A Systematic Survey of Process Improvements, its Benefits, and Success Factors that influence Success. Technical report, SEI (1995)
11. Hall, T., Rainer, A., Baddoo, N.: Implementing Software Process Improvement: An Empirical Study. J. Softw. Process: Improve. Pract. 7, 3–15 (2002)
12. Niazi, M., Wilson, D., Zowghi, D.: Implementing Software Process Improvement Initiatives: An Empirical Study. In: Münch, J., Vierimaa, M. (eds.) PROFES 2006. LNCS, vol. 4034, pp. 222–233. Springer, Heidelberg (2006)
13. Basri, S., O'Connor, R.V.: A Study of Software Development Team Dynamics in SPI. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 143–154. Springer, Heidelberg (2011)
14. Nikitina, N., Kajko-Mattsson, M.: Developer-driven Big-bang Process Transition from Scrum to Kanban. In: 2011 International Conference on Software and Systems Process, pp. 159–168. ACM, New York (2011)
15. Coleman, G., O'Connor, R.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software 81(5), 772–784 (2008)
16. Myers, M.D.: Qualitative Research in Business & Management. Sage Publications, London (2009)

Software Sustainability from a Process-Centric Perspective

Giuseppe Lami, Fabrizio Fabbrini, and Mario Fusani

Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie della Informazione
Via Moruzzi, 1 – I-56124 Pisa, Italy
{giuseppe.lami,fabrizio.fabbrini,mario.fusani}@isti.cnr.it

Abstract. ICT significantly contributes to the global carbon dioxide production. In the last years the research addressed the problem of increasing ICT sustainability from different perspectives. In this paper this problem is addressed from a software process perspective. Sustainability of software process is approached in a systematic way by defining a core set of processes that represent the activities to be performed in order to introduce and integrate the greenness culture in an organization developing software. The processes have been defined so that they can be measured in terms of process capability according to the ISO/IEC 15504 standard. The relationships between process capability and sustainability are discussed as well.

1 Introduction

Problems related to the *environment* preservation and to the sustainable development are among the most important human beings have to face today. Because software intensive systems and applications are more and more pervasive in all the activities of everyone's day life, environmental impact of any related aspect has indeed become an issue. The global ICT industry is claimed to account for approximately two percent of global carbon dioxide (CO_2), a figure equivalent to the aviation industry [5]. The aspects related to the sustainable design, development, use, maintenance and disposal of software and software-intensive systems (that in this light can be denoted as Green Systems) have been in the last years addressed from the following different perspectives:

- *Development of Green Systems:* systems to be designed adopting technical solutions able to minimize the power consumption during their usage, and using materials having a reduced environment impact during production and disposal. In fact, for some ICT products (such as servers or set-top boxes) it is essential to reduce the power consumption during use, because the use phase comprises the largest share in their total life cycle impact; for others it is more important to optimize their design for recyclability or to avoid negative effects during end-of-life treatment. [6, 7, 8, 10]. These issues are addressed by the ISO 14000 family of standards related to environmental management. It aims at supporting organizations to minimize how their operations negatively affect

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 97–108, 2012.

the environment (i.e. cause adverse changes to air, water, or land); to comply with applicable laws, regulations, and other environmentally oriented requirements, and to continually improve in the above aspects [16, 17].

- *Design of Green Software Products:* software products shall be designed in order to adopt efficient algorithms able to reduce both the direct carbon footprint of software (e.g. power consumption due to the CPU cycles) and the indirect effects on sustainability (i.e. the effects depending on the system where the software is executed and on the domain where the system is used). Although software doesn't consume energy, it deeply affects the consumption of hardware equipment, for this reason software is indirectly responsible for energy consumption. [9, 11]
- *Green Software Development:* the focus in this case is on the whole software life cycle, including the development phase, that shall follow principles and adopt techniques aiming at optimizing sustainability. This is strictly related with the concept of green software engineering. [1, 12, 13]

What is still to be addressed in a complete manner in the literature is the sustainability of the software process. Software process determines and drives the organizational *modus operandi* in all the activities directly and indirectly related to the software development. The availability of models and methods for assessing and improving software process in terms of sustainability contributes to the sustainability of software products as well as to spread the organizational greenness culture.

In this paper we provide definitions and principles related to the green and sustainable software process. In addition we refer to the well-known standards ISO/IEC 12207 and ISO/IEC 15504 to clearly identify the core activities related to green and sustainable software process as well as a way to evaluate the capability of an organization in such activities.

This paper is structured as follows: in Section 2 principles and definitions about sustainable software process are provided, in Section 3 factors determining the sustainability of software process are addressed as well as possible sustainability objectives. In section 4 a core set of process definitions (according to the requirements of the ISO/IEC 15504 standard) able to cover the software process sustainability is provided. In Section 5 the relationships between process capability (measurable by means of the ISO/IEC 15504 standard) and sustainability are discussed. Finally, in Section 6 possible future advancements are identified and conclusions are provided.

2 Green and Sustainable Software Process Principles and Definitions

In this section key definitions, taken from the literature, are provided in order to give a clear picture of what green and sustainable software process means and its relationships with key other concepts.

Green & Sustainable Software:
software, whose direct and indirect negative impacts on economy, society, human beings, and environment that results from development, deployment, and usage of the software are minimal and/or which has a positive effect on sustainable development. [1]

Green & Sustainable Software Engineering:
the art of developing green and sustainable software with a green and sustainable software engineering process [1].

Therefore it is the art of defining and developing software products in a way, so that the negative and positive impacts on sustainable development (i.e. a pattern of growth in which resource use aims to meet human needs while preserving the environment so that these needs can be met not only in the present, but also for generations to come) that result and/or are expected to result from the software product over its whole lifecycle are continuously assessed, documented, and used for a further optimization of the software product.

Software lifecycle is a concept introduced in the '80s [19]. The software lifecycle is a framework in which the activities related to development, exercise, maintenance and disposal of software product are identified, and their order of performance established. In other words, a lifecycle describes or prescribes how an organization shall perform the software development, maintenance and disposal.

In [1] the green software lifecycle is addressed with the main objective to assess the ecological, social, human and economic compatibility of a product during its whole life cycle.

A related concept, but different from the software lifecycle, is that of software process. Software process is a collection of general definitions of the interrelated activities that can be suitably performed during the development, maintenance and disposal of a software product. The whole software process is usually considered as composed of different processes each of them covering a specific activity. This approach is established in the ISO/IEC 12207 [3] standard that gives the requirements for an extended set of processes suitable for software development. For this reason hereafter we refer to software processes instead of a unique software process.

The relationships between software lifecycle and software processes can be represented as Figure 1 shows. Software processes are general activities having their own characteristics and requirements. Software processes are mapped on a specific software lifecycle that establishes which processes are used, their sequence of performance and their possible iterations. In general software processes are independent of the specific software lifecycle they are applied in.

Each software process represents a set of interrelated activities (including also the use of resources), that transform inputs into outputs to be used for developing a specific product. Their actual usage in real projects determines the software lifecycle.

Software sustainability can be approached from a process-centric perspective and a definition of Green and Sustainable software process can be given as follows.

Green and Sustainable Software Process:
Software process that meets its (realistic) sustainability objectives, expressed in terms of direct and indirect impacts on economy, society, human beings, and environment that result from its definition and deployment.

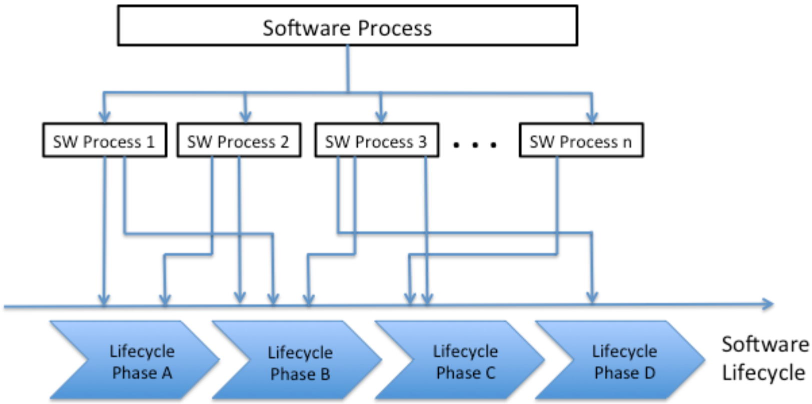


Fig. 1. Software Lifecycle vs. Software Process

3 The Sustainability Factors of the Software Processes

The process sustainability factor (i.e. the part of the carbon footprint due to or save by the specific process used in developing and managing all the aspects and phases related to the software lifecycle) cannot be ignored.

In fact, the way the interrelated activities composing the whole software process are performed and managed may significantly impact on the whole sustainability of the software.

The principal sustainability factors directly impacted by the process are identified in the following list:

- *Power Consumption*: this factor is directly related to the efficiency of the software process. Inefficient processes require unnecessary workload with a consequent waste of power and then avoidable CO_2 emissions.
- *Paper Consumption*: the amount of paper consumed in a software development project depends on the media used for sharing information and knowledge within the organization. Knowledge management and communication infrastructure as well as their usage depend on the procedures defined in the processes. Infinite examples can be provided showing waste of paper in distributing and using documents.
- *Fuel Consumption*: travels and face-to-face meetings can be reduced (with a consequent reduction of fuel consumed) both introducing suitable technologies and defining suitable processes to guide their usage.

The three factors indicated above are not mutually independent. For instance, it would be expected that savings in terms of paper consumption are mitigated by an increase in terms of power consumption. Generally speaking, progress in the efficiency of producing software means that the same results can be obtained using less input. By increasing efficiency, input factors (e.g. energy input) can in principle be saved in absolute terms. However, in practice these savings may be balanced out or even overcompensated by an increase in demand for the output, because the output is getting cheaper in terms of money or time or by a different destination of saved income that can be spent on other goods or services. These phenomena are known as rebound effects [14, 15].

The rebound effect shall be taken into account when sustainability objectives are defined for a software development project. The overall sustainability objective for software processes is to support sustainable development, minimize resource requirements and produce minimal waste. More detailed sustainability objectives can be derived as:

Carbon Footprint: the amount of carbon dioxide a software development, management, or maintenance activity shall emit. The discussion of techniques and methods for calculating reference values to be associated to the carbon footprint objective is out of the scope of this paper.

Energy: the amount of energy consumed during the software development. This objective is easier to be established because it is based on real power consumption no matter how energy is produced.

Waste: the amount of physical, energy or process resources consumed in activities that add no visible value to software and users.

Travel: travelling time required during the software development. This objective can be refined according to the type of vehicle used.

4 Addressing Sustainability in Software Processes

In this section, a model in which the concepts of sustainability and software process are integrated is provided. Once such a model has been given, the evaluation of the degree of process sustainability of an organization will be possible.

For this purpose we refer to the ISO/IEC 12207 standard.

To address sustainability issues in software process we define a set of sustainability-specific processes representing the activities to be performed in order to introduce and integrate the greenness culture in the software process. The definition of this new set of processes follows the rules of the ISO/IEC 12207 and ISO/IEC 15504 standards.

According to those rules a process definition shall contain the following parts:

- Process Title: a definition that conveys the scope of the process as whole
- Process purpose: the statement of the goal of performing the process
- Process outcomes: observable expected results of the successful performance of the process

In the following the core set composed of three processes necessary to embrace the process sustainability as a whole is provided. According to the definition of green and sustainable software process provided in Section 2, these processes represent a way to introduce and establish an organizational *modus operandi* able to support the development of green software.

These three processes are introduced and described using a tabular format respecting the process definition rules provided above.

4.1 Sustainability Management Process

Sustainability management process aims at ensuring the achievement of established sustainability objectives in software development.

It can be decomposed into four phases that are described in the following:

Preliminary phase: the sustainability principles and criteria that will drive the sustainability-related decisions are established. In the following we indicate as *sustainability activities* those activities performed in a product development (technical, managerial and support activities) aimed at satisfying the sustainability objectives.

Planning phase: the sustainability activities to be deployed in the development of the product are identified. The planning of the sustainability activities is produced and the necessary resources to accomplish with it are identified and allocated.

Monitoring phase: the performance of the sustainability activities is monitored for its compliance with the planning. Possible deviations are treated until solution.

Supplier sustainability control: the sustainability requirements for supplied products are defined and a monitoring policy is agreed.

A tabular representation of the process definition is provided in Table 1.

Table 1. Sustainability Management Process

Process ID	SUS.1
Process Name	Sustainability Management
Process Purpose	The purpose of the Sustainability Management Process is to ensure that products, services and life cycle processes meet sustainability objectives.
Process Outcomes	As a result of the successful implementation of the Sustainability Management process: 1) Principles and criteria for sustainability are established. 2) The scope of the sustainability-related activities for the project is defined. 3) Activities for sustainability are planned and implemented. 4) Tasks and resources necessary to complete the activities for sustainability are sized and estimated. 5) An organization structure for sustainability (responsibilities, roles, reporting channels, interfaces with other projects or OUs …) is established. 6) Activities for sustainability are monitored, sustainability non conformities are reported, analysed, and resolved. 7) Agreement on sustainability policy and requirements for supplied products or services is achieved. 8) Supplier's activities for sustainability are monitored.

4.2 Sustainability Engineering Process

Technical solutions for sustainability shall be injected into the development process in order to comply with defined sustainability objectives. The sustainability engineering process addresses the application of techniques and methods able to guarantee that the sustainability activities are suitably integrated into the engineering activities to achieve defined sustainability objectives.

Techniques and methods are applied on the basis of an analysis aimed at verifying their suitability to achieve sustainability objectives.

Possible change requests are analyzed, their impact evaluated and the planning modified accordingly.

A tabular representation of the process definition is provided in Table 2.

Table 2. Sustainability Engineering Process

Process ID	SUS.2
Process Name	Sustainability Engineering
Process Purpose	The purpose of the Sustainability Engineering process is to ensure that sustainability is adequately addressed throughout all stages of the engineering processes.
Process Outcomes	As a result of the successful implementation of the Sustainability Engineering process: 1) Factors affecting sustainability (e.g. resource consumption sources) are identified 2) Sustainability analysis is performed in order to determine the sustainability impact of factors affecting sustainability 3) Sustainability objectives are defined for the product development 4) Green principles are applied to development processes to identify techniques and methods suitable to achieve the sustainability objectives. 5) Techniques and methods for sustainability are applied 6) Impact on sustainability of change requests is analysed

4.3 Sustainability Qualification Process

Software and system development may need the acquisition and use of external resources (as for example: core engineering tools, engineering support tools, management support tools). The overall sustainability is affected by the sustainability of these external resources, then this process addresses the assessment of sustainability of external resources and their management.

To achieve the purpose of this process, a sustainability qualification strategy and a plan that implement such a strategy are to be prepared and implemented. Moreover the outcomes of the qualification are to be documented.

A tabular representation of the process definition is provided in Table 3.

Table 3. Sustainability Qualification Process

Process ID	SUS.3
Process Name	Sustainability Qualification
Process Purpose	The purpose of the sustainability Qualification process is to assess the suitability for sustainability of external resources when developing a software or system.
Process Outcomes	As a result of the successful implementation of the Sustainability Qualification process: 1) Sustainability qualification strategy for external resources is developed. 2) Sustainability qualification plan is developed and executed. 3) Sustainability qualification documentation is written. 4) Sustainability qualification report is produced.

5 Evaluating and Improving Software Processes Sustainability

According to the well-known motto: *You cannot control what you cannot measure* [4] the availability of means and models for measuring the quality of processes and then for improving them, if necessary, is crucial. The compliance of the process definitions provided in section 4 with the ISO12207 and ISO/IEC 15504 requirements allows the use of the ISO/IEC 15504 for measuring the capability of such processes.

Process capability is a concept related to the probability a process meets its goal in terms of quality of outcomes, costs and time; the higher the capability of a process the lower the risk of missing its objective. This concept is represented graphically in Figure 2.

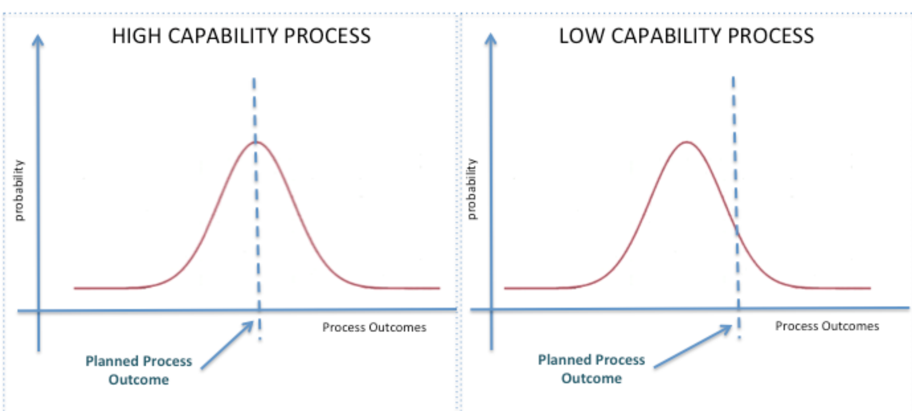


Fig. 2. Software Process Capability graphic representation

The measurement of process capability is addressed in two well-known reference models: the CMMI [18, 20] and the ISO/IEC 15504 [2]. The latter, in particular, provides a flexible way to the determination of software process capability because it is structured in two dimensions: the process dimension and the capability dimension.

Table 4. Capability Level vs. Sustainability in SUS.x processes

Capability Level (CL)	SUS.1	SUS.2	SUS.3
0 Not Performed	There is a general failure to attain the purpose of the process. There are little or no easily identifiable work products or output of the process		
1 Performed	The organization is generally able to ensure that products, services and life cycle processes meet sustainability objectives.	The organization is generally able to ensure that sustainability is adequately addressed throughout all stages of the engineering processes related to software/system development	The organization is generally able to assess the suitability for sustainability of external resources when developing a software or system
2 Managed	The organization follows specified, planned and tracked procedures to deliver work products (conformant to specified standards and requirements) related to: • Definition of sustainability principles and criteria • Planning and monitoring of sustainability activities (including those performed by suppliers) • Resource and infrastructures allocation for sustainability	The organization follows specified, planned and tracked procedures to deliver work products (conformant to specified standards and requirements) related to: • Definition of sustainability objectives • Application of suitable techniques to achieve them.	The organization follows specified, planned and tracked procedures to deliver work products (conformant to specified standards and requirements) related to sustainability qualification strategy definition, planning, deployment and reporting
3 Established	The procedures identified for CL 2 for each SUS.x process are implemented using a documented standard process or an approved tailored version of it		
4 Predictable	A quantitative understanding of the capability of each SUS.x process and an improved ability to predict and manage their performance and the quality of the related work products is achieved by the analysis of detailed performance measurements		
5 Optimizing	The performance of each of the SUS.x process is monitored against organizational business and efficiency goals. Quantitative feedbacks are collected and used for improvement purposes.		

The process dimension comprises processes defined according to certain requirements that are including those of the ISO/IEC 12207; the capability dimension provides a measurement framework composed of process attributes, and process indicators as well as a mechanism to rate the capability of processes.

In the case of processes related to sustainability defined in Section 4, achieving a higher capability level means that the structure of the organization supports the achievement of the sustainability goals in developing software and that the confidence in achieving such goals is enforced.

Nevertheless, the Process Capability and Process Sustainability concepts are different. There is not a direct relationship between process capability and sustainability; in other words, a process having high capability can have a low sustainability and vice-versa.

The three processes defined in Section 4 are, in our understanding, enough to cover all the basic aspects of sustainability in software process . In Table 4 the meaning, in term of sustainability, of the achievement of a certain capability level according to the ISO/IEC 15504 measurement framework is explained.

To be noted that nothing can be directly inferred about the sustainability of the final product from the achievement of a certain capability level according to the ISO/IEC15504 scheme. On the contrary, measuring capability of the processes defined in section 4 means evaluating, from a managerial and technical point of view, the behavior of an organization when it deals with sustainability in developing software.

Applying a capability determination mechanism to the set of processes defined in section 4 can be considered as a contribution in the evaluation of the greenness of an organization and it may support it for improving its greenness as well.

6 Conclusions and Future Works

This paper aims at contributing in the current studies for solving the problem of the ICT sustainability by widening the perspectives from which it is addressed. ICT sustainability has been addressed in this paper by discussing first the factors directly related to the software process. Then a process model composed of the definition of a core set of processes able to address the basic activities to be performed in order to introduce and integrate the greenness culture in organizations developing software has been provided. These processes would represent an addendum to the ISO/IEC 12207 process set and, while they do not impact on the actual Process Reference Model of the ISO/IEC 12207, they influence the way most of those processes are performed because they inject sustainability issues in the whole software development process.

These processes have been defined according to the requirements of the ISOI/EC 15504 standard in order to make possible their assessment in terms of process capability. Finally, the relationships between process capability and sustainability have been discussed.

This paper is to be considered as an initial step towards the definition of a sound framework aimed at integrating the sustainability with the traditional quality characteristics of the software process as productivity, efficiency, and suitability for purpose. To do that we intend to continue with in this promising discipline by performing ISO/IEC 15504 actual process assessments of the core set of sustainability processes defined in this paper, in order to get feedback and possibly extend or modify them.

We are also investigating the definition of a Process Assessment Model for Sustainability and Capability starting from the existing ISO/IEC 12207 and ISO/IEC 15504 standards. To do that the processes defined in the ISO/IEC 12207 standard should be revised with the aim of injecting sustainability-related outcomes and practices. It would be a way to make the ISO/IEC 12207 greener. Moreover the Process Attributes of the ISO/IEC 15504 standard should be revised along with the related indicators in order to make it suitable to provide a measurement of the sustainability level of the software process as well.

References

1. Naumann, S., Dick, M., Kern, E., Johann, T.: The GREENSOFT Model: A Reference Model for Green and Sustainable Software and its Engineering. Sustainable Computing: Informatics and Systems 1, 294–304 (2011)
2. ISO/IEC IS 15504, Information technology – Process assessment, Parts 1-10, International Organization for Standardization (2001-2011)
3. ISO/IEC IS 12207:2008, Systems and software engineering — Software life cycle processes. International Organization for Standardization (2008)
4. De Marco, T., Lister, T.: Controlling Software Projects: Management, Measurement, and Evaluation, Seminar Notes. Atlantic Systems Guild Inc., New York (1984)
5. Gartner, Inc.: Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions, `http://www.gartner.com/it/page.jsp?id=503867` (accessed March 2, 2012)
6. Sverdrup, H., Svensson, M.G.E.: Defining the Concept of Sustainability. Systems Approaches and Their Application, 143–164 (2005)
7. Gordon, J., Skinner, A., Button, A. (eds.): World Economic Forum Green Technology: Driving Economical and Environmental Benefits from ICT (2009)
8. Hilty, L.M., Lohmann, W.: The Five Most Neglected Issues in "Green IT". CEPIS Upgrade The European Journal for the Informatics Professional XII(4) (October 2011)
9. Taina, J.: Good, Bad, and Beautiful Software – In Search of Green Software Quality Factors. CEPIS Upgrade The European Journal for the Informatics Professional XII(4) (October 2011)
10. Rivoire, S., et al.: Models and Metrics to Enable Energy-efficiency Optimization. IEEE Computer (December 2007)
11. Capra, E., Merlo, F.: Green IT: Everything Starts from the Software. In: 17th European Conference on Information Systems, Verona, Italy (2009)
12. Johann, T., Dick, M., Kern, E., Naumann, S.: Sustainable Development, Sustainable Software, and Sustainable Software Engineering. In: Int.l Symposium on Humanities, Science and Engineering Research, Kuala Lampur, June 6-7. IEEE Publisher (2011)
13. Tate, K.: Sustainable software development: An agile perspective. Addison-Wesley, Upper Saddle River (2006)

14. Binswanger, M.: Technological Progress and Sustain- able Development: What About the Rebound Effect? Ecological Economics 36, 119–132 (2001)
15. Hilty, L.M., Köhler, A., von Schéele, F., Zah, R., Ruddy, T.: Rebound Effects of Progress in Information Technology. Poiesis & Praxis. International Journal of Technology Assessment and Ethics of Science 1(4), 19–38 (2006)
16. Darnall, N., Gallagher, D.R., Andrews, R.N.L.: ISO 14001: Greening management systems. In: Sarkis, J. (ed.) Greener Manufacturing and Operations, pp. 179–190. Sheffield, Greenleaf (2001)
17. Miles, M.P., Russel, G.P.: ISO 14000 total quality environmental management: the integration of environmental marketing, total quality management, and corporate environmental policy. Journal of Quality Management 2(1), 151–168 (1997)
18. Chrissis, M.B., Konrad, M., Shrum, S.: CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley (2004)
19. Abdel-Hamid, T.K., Madnick, S.E.: A model of software project management dynamics. In: Proc. IEEE COMPSAC 1982, pp. 539–554 (November 1982)
20. Software Engineering Institute ”CMMI for Development, Version 1.3” CMU/SEI-2010-TR-033 (November 2010)

Integrating Governance, Service Management and Project Management of IT*

Mercedes de la Cámara[1], Fco. Javier Sáenz Marcilla[1], Jose A. Calvo-Manzano[2], and Eugenio Fernández Vicente[3]

[1] Departamento de Lenguajes Proyectos y Sistemas Informáticos
Universidad Politécnica de Madrid, Escuela Universitaria de Informática - 28031, Madrid, España
{mcamara,jsaenz}@eui.upm.es
[2] Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software
Universidad Politécnica de Madrid, Facultad de Informática - 28660 Boadilla del Monte, Madrid, España
josentonio.calvomanzano@upm.es
[3] Departamento de Ciencias de la Computación
Universidad Alcalá de Henares - 28801, Madrid, España
eugenio.fernandez@uah.es

Abstract. IT governance is critical for business improvement in organizations. In the same way, the value of IT must be improved. Often the product of an IT project is an IT Service. A good management of IT projects help to improve IT Services value. For this purpose, some frameworks and standards have been developed. Among them, project management framework PRINCE2 and the IT governance and management standards ISO/IEC 38500 and ISO/IEC 20000 are highlighted. Their application means a clear improvement, but this is far from covering the organizations expectations. IT project approach integrating the needs of business and technology that support is a key to achieve the expected success for IT projects in organizations. This paper presents a summary of the study performed to find out if PRINCE2 meets the governance and management requirements of ISO/IEC 38500 and ISO/IEC 20000. The study could help improving success expectations of PRINCE2 projects.

Keywords: Project Management, IT Corporate Governance, IT Services Management, ISO/IEC 20000, ITIL, ISO/IEC 38500, PRINCE2.

1 Introduction

The concept of IT governance appeared in the 90's. Since then, there have been different definitions. Webb's study [1] shows some of them. IT governance can be defined as a set of capabilities, responsibilities and procedures which help decision-making related to the incorporation of IT in processes of the organization, improving the performance of its objectives. Actually, organizations choose the IT to accomplish

* Authors are ordered by University & School.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 109–120, 2012.

their goals, but many of them are compelled to incorporate IT in order to continue competing. Organizations must make decisions about their IT, which must be integrated with the business objectives. IT projects are addressed by organizations through IT department. IT governance decisions must be integrated with business objectives. Furthermore, they must be transmitted to IT services management and IT projects management processes.

IT Governance covers five areas: strategic alignment, resource management, risk management, performance measurement, and delivering value to the organization. All of them are related and it is difficult to get the return value if organizations are not successful in all these areas [2]. Some organizations focus IT from the service point of view, which provides a value defined in business cases. They are ITSM-based organizations. This paper is focused on projects, having IT Services as products. So, the success of new services or the modifications of existing ones depends on: IT governance guidelines, service management and project management practices. All of them must be integrated. Governance and management enable IT projects: (1) be controlled throughout entire life cycle by several processes, so there are less deviations from the strategic objectives, (2) manage the risks that allows to obtain the expected performance of resources, and (3) measure the value for the organization.

The strategy and new project management approach are directed towards the management of IT programs integrated into business strategy [3][4]. IT governance and management, applied to the context of project management, facilitate defining controls and measures which help to make decisions without losing sight of the risks and business objectives at three levels: portfolio, program and project.

Galorath [5] and Rupinder & Sengupta [6] jobs show that the maturity of IT projects is still below the levels expected by organizations. Three aspects are highlighted as possible causes: (1) cost of the project, (2) timeliness of delivery, and (3) benefits of the project's product. The reports show the existence of large gaps in the entire project life cycle. Thus, in order to improve the PRINCE2 framework, our research is oriented to know the directives of IT governance standards which are not included in project management processes. This paper shows in section 2 the background in governance, management and IT project management. Section 3 presents some aspect about integration of PRINCE2 (2009 version) framework [7], and governance and management standards ISO/IEC 38500 [8] and ISO/IEC 20000 [9][10][11]. Finally, Section 4 presents the conclusions of this paper.

2 IT Governance Standards and Project Management with PRINCE2

Recently, there have been various initiatives to implement standards and frameworks for governance, IT services management and IT project management. Most of these are focused on customer needs and clearly aimed towards improving the performance of IT projects. In the area of project management, some frameworks have been appeared, among them PMBoK (Project Management Body of Knowledge) from PMI (Project Management Institute) [12][13]; PRINCE2 (Project In Controlled

Environment) [7]; NCSPM (National Competency Standards For Project Management) [14]; ICB (IPMA Competence Baseline) [14]; Project Excellence Model [15]; OPM3 Organizational Project Management Maturity Model [16], and standards: ISO 10006:2003 Quality Management System [17]; ISO/DIS 21500 Guidance on Project Management [18] or UNE 157001 General Criteria for Project Development [19]. Currently, project management is evolving towards Project Management Office (PMO, Project Management Office) where the Chief Project Officer (CPO) and IT management perform IT governance efforts.

With the same objective, to integrate business and IT, the area of IT governance defines new frameworks and models, improving the effectiveness and efficiency of their IT services and the performance of business processes. Thus it arises ITIL (Information Technology Infrastructure Library) [20] in its different versions, COBIT 4.1 (Control objectives for information and related technology) [21] and 5.0 [22], Val IT 2.0 [23], Risk IT [24], COSO (Committee Of Sponsoring Organizations) [25], and standards as ISO/IEC 20000 Information Technology: Management Service [9][10] and ISO/IEC 38500 Corporate Governance of Information Technology [8] and family (under development) among others.

On the one hand, PMBoK and PRINCE2 are highlighted among those project management frameworks that have been studied. PMBoK from United States and PRINCE2 with origin in the United Kingdom have been established as the standards for project management. Although both are having a great influence in several countries, in recent years PRINCE2 is having a major growth by extending its scope to more than 150 countries worldwide [26]. On the other hand, ISO/IEC 38500 standard is the first international standard for IT governance. Therefore, PRINCE2 is the project management framework chosen to be integrated with ISO/IEC 38500 IT Governance standard. The guidelines for IT governance must be transmitted both the processes of project management and the management of IT services. Project management processes must reflect the policies of governance among its activities. We could have chosen ITIL, because ITIL and ISO 20000 include some activities about implementing IT governance guidelines in the IT management services. Some of them could be applied to project management. We have chosen ISO 20000 and 38500 because they are guides from international standardization organization.

2.1 PRINCE2 as Project Management Framework

The main structure of PRINCE2 project management model is based on seven *processes* involved in a project life cycle. Each process has to comply with seven *themes* or knowledge areas, everyone defined by seven *principles*. PRINCE2 establishes and defines the responsibilities of project management into nine *roles*. The necessary activities to fulfill the responsibilities of the organization during the life of the project are allocated among the above nine roles.

Processes explain what needs to happen and when throughout the project life cycle. Each process covers all the themes and principles. The main processes are: (1) Starting Up a Project where the requirements needed to begin the project are collected, (2) Initiating a Project, the above information is analyzed in order to make a plan that justifies the project, (3) Directing a Project, the Project Board manages and controls the project, (4) Controlling a Stage where Project Manager day by day does activities in relation to monitoring and controlling the project, (5) Managing Stage Boundaries, to manage in a controlled manner the end of one phase and planning the next (6), Managing Product Delivery, that includes activities to deliver the products or services which will be used by users, and (7) Closing a Project, which manages the delivery of products or services produced in the project.

The *principles* must be fulfilled in each of the areas or themes. These are: (1) continued business justification of the returning value to the customer, (2) learn from experience, (3) define roles and responsibilities, (4) management by planned phases, (5) management by exceptions, (6) approach towards the products, (7) and continuous alignment with the environment project, program and portfolio.

Themes are areas of knowledge to be applied to the project: (1) *Business Cases*, where the justification for an organizational project is done containing costs, benefits, risks and timescales, continuingly testing viability, (2) *Organization*, describing the sponsor of the project which needs to allocate the work to manages who will be responsible for it and steer it to completion, (3) *Quality* explains how the outline is developed so that all participants understand the quality attributes of the products to be delivered, (4) an approved *Plan* on which the project is based, (5) Risk Management showing how the project management manages the uncertainties in its plans in the wider project environment, (6) Change describes how the project management assesses and acts upon issues which have a potential impact on any of the baseline aspects of the project, and (7) Progress theme addresses the ongoing viability of the plans.

The *roles* describe the sets of specific responsibilities. PRINCE2 describes the followings: (1) Project Board is the accountable for corporate or program management, (2) Senior User is the responsible for specifying the needs of those who will use the project's products, for user liaison with the project management team, and for monitoring that the solution will meet the expectations, (3) Executive is the ultimately responsible for the project, supported by Senior User and Senior Supplier, (4) Senior Supplier represents the interests of those designing, developing, facilitating and implementing the project's products (5), Project Manager has the authority to run the project daily, (6) Project Assurance covers the primary stakeholder interests, (7) Changes Authority is to whom the Project Board delegates its authority for approving requests for change, (8) Project Support can be provided by a project office or by specific resources for the project, and (9) Team Manager's prime responsibility is to ensure production of those products defined by the Project Manager with the appropriate quality level.

2.2 ISO/IEC 20000. IT Service Management

ISO / IEC 20000 standard family are published by ISO (International Organization for Standardization) and IEC (International Electro technical Commission). It is the internationally recognized standard in IT Service Management Information Technologies. Its purpose is to set a specific standard for IT service providers, that helps to the success of IT projects. First part of ISO/IEC 20000 is a control way, and the code of practice to help them to develop, deliver and maintain the products, services and systems. ISO/IEC 20000 covers the following sections:

- *IT Service Management System.* Defines the requirements for a management system.
- *Planning and implementing service management.* The standar uses the Deming cycle, known as "Plan-Do-Check-Act" (PDCA), for applying it to all processes.
- *Planning and implementing new or changed services.* The objective is to ensure that new services, changes to services and their elimination will be deliverable and manageable at the agreed cost and service quality.
- *Service delivery process.* It includes: Service level management, which defines the agreement, record and manage levels of service, Service reporting, Service continuity and availability management, Budgeting and accounting for IT services to budget and account for the cost of service provision, Capacity management for ensuring that the service provider has always sufficient capacity to meet the current and future agreed demands of the customer's business needs, and Information security management which manages information security effectively within all service activities.
- *Resolution processes.* Incident and problem management processes cover: Incident management to restore agreed service to the business as soon as possible or to respond to service requests, and Problem management to minimize disruption to the business by proactive identification and analysis of the cause of incidents and by managing problems to closure.
- *Control processes.* It covers: Configuration management which defines and controls the components of the service and infrastructure and maintains accurate configuration information, and Change management ensuring all changes are assessed, approved, implemented and reviewed in a controlled manner.
- *Release process.* Its aim is to deliver, distribute and track one or more changes in a release into a real environment.

2.3 ISO/IEC 38500. IT Governance Standard

ISO / IEC 38500 has been created with the aim of providing a framework that can be used by directors and managers for management and monitoring the use of IT. It helps organizations to understand and comply with legal, regulatory, and ethical obligations relating to the use of IT.

ISO/IEC 38500 establishes six principles for a good corporate governance of IT. These principles represent the behavioral guide for decision making. The statement of each principle refers to what should happen, but does not provide how, when or by whom these principles should be implemented. The direction of the organization should be required to apply all these principles.

- *Responsibility*: individuals and groups within the organization understand and accept their responsibilities about IT offer and demand.
- *Strategy*: the business strategy and plans of the organization must take into account the current and future capacities of IT.

- *Acquisitions*: IT acquisitions are justified based on a proper analysis with a clear and transparent decision making. There must be a proper balance among the benefits, opportunities, costs and risks, both short and long term.
- *Performance*: IT supports the organization providing services, service levels and quality service required to meet the requirements of current and future business.
- *Conformance*: IT complies with all laws and regulations required. The policies and practices are clearly defined, implemented and enforced.
- *Human Behavior*: IT policies, practices and decisions reflect respect for the human factors of staff, their current needs and its evolution within the organization.

For each principle the model proposed in ISO/IEC 38500 applies cycles based on three main tasks: evaluation, direction and monitoring.

Table 1. ISO/IEC 38500 model: principles and tasks for IT governance

PRINCIPLES	EVALUATION	DIRECTION	MONITORING
Responsibility	• Assignment Options. • Competences of leaders decision making.	• Assignment plans. • Information.	• Appropriate governance mechanisms. • Governance Competence. • Governance activities performance.
Strategy	• Develop Strategy and Business Case. • Activities aligned. • Possibility of Risk.	• Plans. • Political. • Innovation proposals.	• Progress of the proposals. • Use of IT and expected profits.
Acquisitions	• Delivery options. • Risk Analysis. • Performance.	• Appropriate acquisitions. • Required capabilities. • Organizational needs.	• Skills required. • Suitable comprehension. • Internal and external.
Performance	• Business Case capacity (to do, know how). • Continuity operations. • Integrity and protection. • Efficacy and efficiency of the governance.	• Adequate resources. • Support for business.	• Monitoring of political of Business Support. • Allocate resources. • Data Accuracy. • Efficient IT use.
Compliance	• Regulations. • Internal political. • Standards. • Internal compliance.	• Establish mechanisms. • Internal Obligations. • Professional behaviour. • Ethical Actions	• Audits. • Activities.
Human Resources	• Identified. • Considered.	• Consistent with activity. • Reported by anyone.	• Activities. • Appropriate use.

- *Evaluation* of proposals for implementation and use of IT, examining and judging. It is necessary to consider continuously and dynamically both internal and external pressures without losing sight of the goals of the organization.
- *Direction* implementation projects of these proposals. The manager should assign responsibilities, governing the preparation and implementation of plans and policies, establishing the government investment in IT projects and operations, and ensuring the transition of project outputs to the operating state.
- *Monitoring* the operations of the success of each principle is dependent. The goal is to achieve compliance policies and act according to the plans established. It needs to control, through appropriate measurement systems, the performance and compliance of IT.

Related to the responsibility about particular aspects of IT, it can be delegated to managers inside the organization. However, with the aim of achieving an effective, efficient and acceptable use and delivery of IT to the organization, the accounting issues belonging to managers cannot be delegated.

The practices described in ISO/IEC 38500 are not exhaustive, however, they provide a starting point for discussion of the responsibilities of managers for IT governance. ISO/IEC 38500 delegates the responsibility to identify specific actions needed to implement the principles of IT governance within the organization. The organizations will discuss the most appropriate way to manage the risks and opportunities using IT. Table 1 shows the focus of ISO/IEC 38500. Above each principle, we develop the three tasks: assessment, direction and monitoring.

The hypothesis of this work assumes that standards ISO/IEC 20000 and ISO/IEC 38500, integrated within the framework of PRINCE2, can help to prevent failures and improve the cost, delivery and therefore the guarantee of IT projects.

The main research question was: which aspects of governance of ISO/IEC 38500 and ISO/IEC 20000 standards are integrated into PRINCE2? Themes, processes and roles of PRINCE2 were confronted with activities of governance and management standards ISO/IEC 38500 and ISO/IEC 20000. The process used for this research and its results are presented in section 3. Finally, in section 4 we have distilled a set of conclusions which will enable us to continue our working definition of an improved model for IT project management.

3 Integrating Standards of Governance and IT Management in PRINCE2

In previous sections we have shown that IT governance and management are responsible for the demand and supply of IT services through projects. Although the principles and tasks of ISO/IEC 38500 directly involve senior management and managers, governance includes decisions that affect all stakeholders, particularly those who are involved in IT projects. Although ISO/IEC 20000 standard has its focus on service management, it also includes activities of governance and corporate strategy. Furthermore, [5] and [6] show that the most important fail of IT projects are in costs, timeliness of delivery, and benefits of the project's product, all of them related to good IT Governance and Management decisions. So, we need to know the direction and gaps from IT governance decision to IT project management processes. Murray [26] points out that when an IT project management framework is adopted by an organization, some further aspects of governance of project management will be achieved too. He focus the problem through governance of project management in the Management of Portfolios (MoP) guide, the Portfolio, Program and Project Office (P3O) guide and the Portfolio, Program and Project Management Maturity Model (P3M3). Senior management commitment to good governance is critical for both the governance of a project and the governance. Almost every study about project management performance shows that appropriate senior management involvement directly correlates with successful projects. Good governance should not be seen as an additional bureaucratic burden for an organization but as an aid to reducing costs and gaining more value by avoiding poor project selection and execution.

IT management projects need some of IT services management processes. The guidelines of ISO/IEC 38500 and ISO/IEC 20000 should be applied throughout the service lifecycle, including the stage at which the service is in project. ISO/IEC 38500 defines the guidelines for IT governance. ISO/IEC 20000 provides some guidelines of governance and a set of IT management processes which allow to manage the IT governance decision and guidelines. IT project management processes have to be developed in this sense.

In this context, the main research question was: "does PRINCE2 meet the governance requirements of ISO/IEC 38500?" and "does PRINCE2 meet the governance requirements of ISO/IEC 20000?" In the previous section we have seen that (1) both ISO/IEC 38500 and PRINCE2 have principles to be applied to their respective activities and processes, (2) ISO/IEC 20000 has no core principles, (3) both ISO/IEC 20000 and PRINCE2 are based on processes, and (4) although ISO/IEC 38500 does not define processes, the standard offers a set of activities contained in three types of tasks. Thus, research has been approached from three perspectives:

- *Themes and principles:* does PRINCE2 themes and principles meet the governance ISO 38500 principles?
- *Process:* does PRINCE2 processes meet the core guidelines of the principles and tasks from ISO/IEC 38500 shown in Table 1?
- *Roles*: does PRINCE2 roles meet the responsibilities described in ISO/IEC 38500 and ISO/IEC 20000?

After analyzing each item of ISO standards matching with themes, processes and roles of PRINCE2, each item value was increased if there was coincidence. This paper shows the percentage of these coincidences. To answer the first question, it has been constructed a matrix shown in Table 2. Rows show the six principles of ISO/IEC 38500 and fifteen ISO/IEC 20000 processes. Columns show themes, processes and roles of PRINCE2. In relation to the themes of PRINCE2, it should be noted that the framework applies each of the seven principles to each of the six subjects. For example, the business cases are analyzed from the perspective of a continued commercial justification, returning value to the customer, learning from experience, etc. This is repeated in each theme. Therefore, the topics have been chosen as a factor to be considered.

The aim was to study whether each principle of ISO/IEC 38500 is reflected in every theme of PRINCE2. A "1" in the cells of Table 2 means that there is a match between the two parameters compared and 0 means that there is no coincidence. Regarding the PRINCE2 processes and roles, the comparison has been made related to both ISO/IEC 38500 and ISO/IEC 20000. To do this, it has been defined the last big columns in Table 2: PRINCE2 processes and roles. The cells of the matrix are filled according to the degree of integration between both ISO/IEC standards and management project framework.

Table 2. Integrating ISO/IEC 38500 and ISO/IEC 20000 with PRINCE2

IT Governance ISO 38500 (Principles)	THEMES							PROCESSES							ROLES								
	Business Case	Organization	Quality	Plans	Risk	Change	Progress	Starting Up	Directing a Project	Initiating a Project	Controlling a State	Managing Product Delivery	Managing a Sstage Boundary	Closing a Project	Project Board	Executive	Senior User	Senior Suppplier	Project Manager	Team Manager	Project assurance	Change Authority	Project Support
Responsability	1	1	1	1	1	1	1	1	3	2	1	2	1	1	2	1	2	2	2	0	3	0	1
Strategy	1	1	1	1	1	1	1	3	1	2	2	2	2	1	3	3	2	2	2	1	2	0	1
Acquisition	1	1	0	1	1	1	1	1	3	1	2	2	2	3	2	2	2	2	3	1	3	1	0
Performance	1	0	1	1	1	1	1	2	2	3	2	1	3	3	1	2	3	2	2	1	3	1	1
Conformance	1	0	1	1	0	0	0	1	1	1	1	1	1	1	1	2	1	1	2	1	3	0	1
Human Behaviur	1	1	1	0	1	0	0	2	1	1	1	1	1	1	0	1	2	1	3	1	3	0	1
IT Governance & Manag. ISO 20000 (Req. & Procc.)																							
Management System	0	1	0	1	0	0	0	1	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0
Planning and Implementing	1	1	1	1	1	0	1	1	1	1	2	1	1	1	1	1	0	2	2	1	1	1	0
Plan-Implem. Changed	1	1	1	1	0	1	1	1	2	1	2	2	1	1	1	1	0	0	1	0	0	2	0
Reporting	1	1	1	1	1	1	1	0	2	0	3	2	3	0	2	1	2	0	1	3	2	1	3
Continuity and Availability.	0	0	1	1	1	0	1	0	3	0	2	2	2	3	1	1	2	2	2	2	3	0	1
Budgeting and Accounting	1	0	0	0	0	0	0	3	2	2	0	0	1	2	2	3	0	0	0	0	0	0	0
Service Level Management	1	1	1	1	0	0	1	3	3	0	2	2	3	2	1	3	3	3	3	0	2	0	0
Capacity Management.	0	0	1	1	0	0	1	2	1	0	2	2	1	2	1	1	0	2	2	2	2	0	1
Security Management	0	1	1	1	1	1	1	0	2	0	2	2	1	0	1	0	0	0	3	2	2	0	1
Business Relationship M.	1	1	0	0	0	0	0	3	2	2	0	0	3	0	3	1	0	0	1	0	0	0	0
Incident Management	0	0	0	0	1	1	1	0	3	0	2	2	2	0	2	3	0	3	1	0	2	0	0
Problem Management	0	0	1	0	1	1	1	0	0	0	3	0	0	1	1	0	2	0	3	2	0	0	1
Configuration Manag.	0	0	1	1	0	1	0	0	0	0	3	0	0	1	1	0	0	0	3	2	1	0	1
Change Management	0	0	1	1	1	1	1	0	3	0	3	1	1	1	1	0	0	0	3	0	1	2	3
Release Process	0	0	1	1	0	1	1	0	2	0	3	2	0	0	2	0	0	0	1	2	1	3	0

Also, it was necessary to establish a range of integration among processes and roles of PRINCE2, ISO/IEC 38500 and ISO/IEC 20000. The directives and activities were included in a table and they were identified or no for PRINCE2 processes and roles. Once the number of them is known, a range was established: [0-25], [25-50], [50-75] and [75-100]%, respectively. Thus, “0”, “1”, “2” and “3” show an index about the number of directives of each principle from ISO/IEC 38500 present in PRINCE2. In the same way, an index about the activities and guidelines from ISO/IEC 20000, which are included in PRINCE2, are shown.

4 Conclusions

The research work allows to know an index of PRINCE2 compliance for governance directives, so it may guide us about the gaps using this project management framework. This may provide a basis for developing a new IT project management framework. The new framework could specify governance policies related to management activities in IT projects. So a IT project could turn back to the organization the confidence needed when addressing IT projects.

An organization may have to choose between two project management frameworks. The result of this research shows them a compliance value for each governance principle.

Both standards define what needs to be addressed in IT governance and management, but no one specifies how to deal with governance and management directives specifically. This specification is delegated on the organization on both standards. This makes them flexible enough to be applied but also it has a negative impact to know the returned value after applying them to IT projects.

From the study of themes of PRINCE2, it can be seen that only the Business Cases include 100% of ISO/IEC 38500 principles. This is because activities and responsibilities related to Business Cases cover most of the tasks of governance policies.

If the degree of ISO/IEC 38500 principles and ISO/IEC 20000 guidelines integration on the themes of PRINCE2 is analyzed, it can be seen a set of gaps:

- The principle of investment is not covered by the issue of quality. This is a serious shortcoming since quality means not only achieving customer expectations, but the return of value.
- Although the overall aims of the "Organizing" PRINCE2 is corporate governance, none of the responsibilities and guidelines described in this theme include the principle of performance guidelines. This gap may negatively affect both the cost, and on time deliveries and expectations of value generated by the project.
- Similarly, the principle of conformity is provided in the Business Case, but is not contemplated in themes such as Organization, Risk Management, Change Control and Project Progress. We believe that the activities which help to manage regulations, define its governance mechanisms, and monitor them should be specified in greater detail.
- The human factor is treated in a very superficial way in ISO/IEC 38500. The human factor is not presented on themes such as PRINCE2 Plans, Change Control, or Progress. All of this need to assess risks, opportunities, problems and concerns of stakeholders as human beings.

PRINCE2's effort to define roles is reflected in the high compliance with the guidelines of governance in the Project Manager and Project Assurance Manager. However, their activities are focused more on the evaluation and monitoring of the product, not covering much of the government guidelines such as innovation, business support, ethical and human factors.

We must emphasize the low involvement of some PRINCE2 roles as Change Authority in ISO/IEC 20000 governance guidelines. This role is essential to the project success, it should have a greater involvement in planning and change management processes, and should contemplate guidelines including innovation changes.

Budgeting and accounting for IT services process is shown only in terms of governance. It is reflected just at the executive level and project board. Moreover, this is not referred enough in PRINCE2 processes.

For next steps, the definition of a concrete guideline framework specifying IT governance is being developed. The relationships between roles and activities involved are being establishing in the three corporate levels: (1) IT innovation as driver of change and business opportunities, (2) IT projects for generating business value considering IT as a service, and (3) IT services, as agents for generating customer value.

Acknowledgements. This work is 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. Webb, P., Pollard, C., Ridley, G.: Attempting to define IT Governance: Wisdom or Folly. In: Proceedings of the 39th Hawaii International Conference on System Sciences (2006)
2. ITGI. Enterprise Value: Governance of IT Investments, The Val IT Framework, USA (2006) ISBN 1-933284-32-3, `http://goo.gl/F2i1i`
3. Iamratanakul, S., Patanakul, P.: Project Portfolio Selection: From Past to Present. In: Proceedings of the 2008 IEEE ICMIT, pp. 287–292 (2008), `http://goo.gl/gpC3V`
4. Patanakul, P.: Effectiveness in project portfolio management: Toward a conceptual definition. In: 2011 Proceedings of PICMET 2011, Technology Management in the Energy Smart World (PICMET), July 31-August 4, pp. 1–7 (2011), `http://goo.gl/j6BbU`
5. Galorath, D.: Software Project Failure Costs Billions. Better Estimation & Planning Can Help (2008), `http://goo.gl/1n6op`
6. Kaur, R., Sengupta, J.: Software Process Models and Analysis on Failure of Software Development Projects. International Journal of Scientific & Engineering Research 2(2) (2011), `http://goo.gl/rkKxK`
7. OGC. Managing Successful Projects with PRINCE2. OGC (2009) ISBN-10: 0113310595
8. ISO. ISO/IEC 38500:2008. Corporate governance of information technology (2008)
9. ISO. ISO/IEC 20000-1:2005. Information technology. Service management. Part 1: Specification (2005)
10. ISO. ISO/IEC 20000-1:2011. Information technology. Service management Part 1: Service management system requirements (2011)
11. ISO. ISO/IEC TR 20000-4:2010. Information technology. Service management. Part 4: Process reference model (2010)
12. IEEE. IEEE Guide. Adoption of the Project Management Institute (PMI®) Standard A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 4th edn. (2011), `http://goo.gl/8Hm01`

13. PMI. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - 4th edn. PMI (2008) ISBN13:9781933890517
14. IPMA. ICB-IPMA Competence Baseline, version 3, Netherlands (2006), http://goo.gl/K6gdp
15. Westerveld, E.: The Project Excellence Model®: linking success criteria and critical success factors. International Journal of Project Management 21(6), 411–418 (2003) ISSN 0263-7863, http://goo.gl/oHlhp, doi:10.1016/S0263-7863(02)00112-6
16. PMI. Organizational Project Management Maturity Model (OPM3®) Knowledge Foundation - 2nd edn. PMI (2008) ISBN: 978-1-933890-54-8
17. ISO. ISO TC/SC2 10006:2003, Quality management systems. Guidelines for quality management in projects, Switzerland (2003), http://goo.gl/WQxAl
18. ISO. ISO/DIS 21500 Guidance on project management (2006), http://goo.gl/QfQF3
19. UNE. UNE 157001:2002 Criterios Generales para la Elaboración de Proyectos. AEN/CTN 157, Proyectos (2002), http://goo.gl/XYQtE
20. OGC. The Introduction to the ITIL Service Lifecycle Book (Official Introduction). TSO, Ed. OGC, London (2010) ISBN-10: 0113311311
21. ITGI. CObIT 4.1. (2007), http://www.itgi.org
22. ITGI. CObIT 5.0. (2011), http://www.itgi.org
23. ISACA. Enterprise Value: Governance of IT Investments. The Val IT Framework 2.0 Extract (2010), http://goo.gl/1WeAM
24. ISACA. THe Risk IT Framework (2009), http://goo.gl/qAY0T
25. Singleton, T.: The COSO Model: How IT Auditors Can Use It to Evaluate the Effectiveness of Internal Controls. In: ISACA (2007)
26. Fernández, R.J.: Datos sobre las certificaciones PRINCE2 y PMP (2010), http://goo.gl/fhHWb
27. Murray, A.: PRINCE2® and governance. White Paper (November 2011), http://goo.gl/OQEDB

A Harmonized Multimodel Framework for Safety Environments

Xabier Larrucea[1], Izaskun Santamaria[1], and Paolo Panaroni[2]

[1] TECNALIA, Software Systems Engineering, Parque tecnologico de Zamudio 202, 48170 Bizkaia, Spain
{Xabier.larrucea,Izaskun.santamaria}@tecnalia.com
[2] INTECS, SpA. Salita del Poggio Laurentino 7I– 00144 Roma, Italy
paolo.panaroni@intecs.it

Abstract. Safety critical systems developments are dealing with complex environments especially when they are satisfying with a wide range of models, regulations and standards at the same time. Therefore there is an evident impact in software process improvement (SPI) settings. These multimodel environments are not just specific for critical systems but for a wide set of environments. Some experiences adopting two different standards or models have been reported during these recent years. Basically they are based on mappings between process models at a high level. However these works are appropriated for people with a huge knowledge on two specific standards, but they do not provide a roadmap on how to effectively apply to company settings. This paper is focused on to provide a harmonised framework not only covering practices based process models but also covering products characteristics. In addition we reconcile this framework for safety critical systems.

Keywords: Multimodel framework, quality models, process improvement.

1 Introduction

Organizations related to safety critical systems developments are dealing with complex environments where quality [11] and safety assurance [15] are some of their business' cornerstone. Multiple models, regulations, standards and laws coexist in the same environment generating and adding complexity to products developments. These multimodel environments are not just specific for critical systems developments but for a wide set of environments such as introduced in [5,6,7,8,9] where a general description of multimodel environment and a general approach is provided. Some experiences have been reported referring to the adoption of several models such as [1, 2, 3]. This is a common situation in software process improvement (SPI) settings, and adopted in several contexts for enhancing not only organizations' revenues and performances, but in general their business.

During these recent years it has been discussed in literature such as in [10] about multimodel environments where authors introduced a framework basically based on mappings between process models at a high level. In fact the mapping between

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 121–132, 2012.

process models is the most used technique. This is the case for [1, 2, 3] as mentioned before. These works provide some valuable insights for people with a high knowledge on two specific standards, but they do not provide a roadmap on how to effectively apply to company settings especially in safety critical systems.

Safety critical environments context is similar to the rest of multimodels cases to some extent. However the nature of process models and standards vary significantly. Most of these environments deal with the application of regulations, standards for products [17] and even for organizational processes [18]. Several research works such as [16] provide experiences applying software process improvements initiatives and reporting their benefits. This is the case for CMMI-DEV [18], a widely known practices based process model. Due to the relevance of this model we have used it as a basis for a process based improvement model. In fact there is an extension of CMMI® called +SAFE for safety environments developments [19]. However all these models are defined with a specific purpose without taking into account the existence of other models.

Therefore the focus of this paper is based on the following research questions:

RQ1. Is it possible to define a multimodel framework not only covering practices based process models but also covering products characteristics based models?

RQ2. Can we reconcile this framework for safety critical systems?

This paper is structured as follows. Section 2 provides an overview of the standards and process models used for building this framework analyzing recent research works. Section 3 defines the harmonized framework. Section 4 describes an adaptation of this framework to a safety critical environment. Section 5 discusses some conclusions and future research lines.

2 Research Background

2.1 CMM to CMMI® Constellations and a Safety Adaptation Model

The literature reports several experiences applying Software Capability Maturity Model (CMM), the previous version of CMMI (1.1, 1.2 and currently v1.3), for software development successfully such as [22] where authors provided their experiences applying Software Process Improvement (SPI) based on software for CMM, and [23] reporting a high maturity case study. Nowadays software CMM has evolved to CMMI for development (CMMI-DEV) and there are also reports related to this application such as [24] where authors develop a SPI implementation maturity model. CMMI is described by practices with general and specific goals and grouped in process areas.

+SAFE [19] is an adaptation of CMMI for safety environments. As stated in [19] *+SAFE is an extension used for defining goals and for increasing levels of performance capability,* and in fact it is not used for complying with specific standards. Basically +SAFE adds safety management and safety engineering process areas and they define also some specific goals and practices to be performed by organizations.

Table 1. SEI's +SAFE 2007 process areas and goals

CMMI PA Category	Safety Process Area	Specific Goals
Project Management	Safety Management	SG1 Develop Safety Plans SG2 Monitor Safety Incidents SG3 Manage Safety Related Suppliers
Engineering	Safety Engineering	SG1 Identify Hazards, Accidents and sources of hazards SG2 Analyse hazard and Perform Risk Assessments SG3 Define and maintain safety requirements SG4 Design for safety SG5 Support safety acceptance

Table 1 describes briefly the +SAFE structure. However each process area also contains a set of generic goals as CMMI-DEV. Each goal in its turn it describes a set of best practices to be performed.

2.2 ISO26262

The last version of ISO26262 [21] has been released in 2011 and it is a standard covering all activities during the safety lifecycle of safety related systems in the context of road vehicles. ISO26262 consists of the following parts:

1. Vocabulary
2. Management of functional safety
3. Concept phase
4. Product development at the system level
5. Product development at the hardware level
6. Product development at the software level
7. Production and operation
8. Supporting processes
9. Automotive Safety Integrity Levels (SIL)-oriented and safety oriented analyses
10. Guideline on ISO26262

Basically this standard specifies the following automotive lifecycle: management, development, production, operation, service, decommissioning. All these phases and different parts of this standard define product development best practices for system, hardware and software levels.

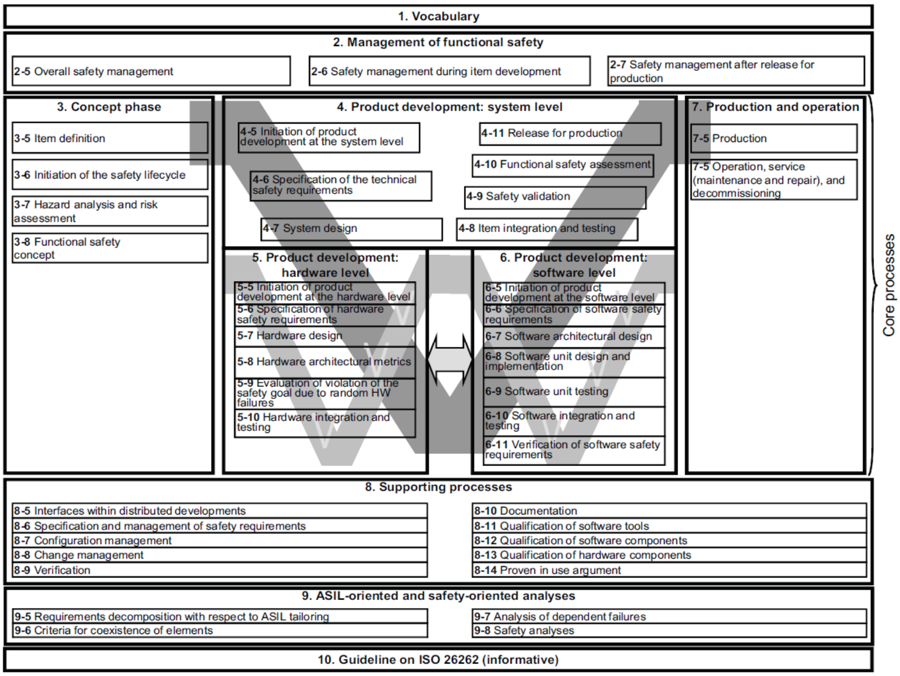


Fig. 1. Overview of the ISO26262

There are other management and concept phases and 10 supporting processes and ASIL levels specifications. An overview of this standard is described in Fig. 1. A general V-Model is the main backbone for an ISO26262 based safety product development, containing in its turn two consecutives v-models for hardware and for software levels.

2.3 Multimodels Approaches

As it is stated in the introduction section, during these recent years some multimodels approaches has been defined such as [5, 6, 7, 8, 9] where basically the multimodel concept is introduced and some general approaches and tools are suggested as a way to face up this kind of environments. Multimodel approaches deal with the fact that organisations need to satisfy a wide range of models/regulations/standards/laws at the same time. The aforementioned models do not deal with multimodel environment because they are just reference models or standards.

Other research works such as [10] defines a multimodel framework for mapping ISO9001 [20] to CMMI-DEV [18]. Basically they compare both models at a high level assuming a relationship between practices from CMMI to statements in ISO 9001. However they do not deal with a wider range of models where their granularity is different and their purpose is orthogonal. In fact this issue is even worse when dealing with safety models. And at the end it is only applicable for technology expert such as expert on ISO9001 or CMMI-DEV.

3 A Harmonized Multimodel Framework

Safety critical systems require a harmonized [5] multimodel framework for dealing with the aforementioned situation. An overview of the proposed framework is shown in Fig. 2. This framework defines the following three layers:

- **Realm of Reference Technologies:** this layer is used by technology experts having a deep knowledge on standards, laws, reference models and so on. For example the research work presented in [10] refers this layer as the framework for supporting multi-models Harmonization. In our approach we introduce new concepts and we extend this framework to the following two layers: organizational processes and evidences. In our approach all these technologies, standards and models are represented as "Quality model" and they are compared following a specific method such as [10]. Mappings between pairs of models are defined through drilling down into each model and comparing atomic pieces of each. As result of this comparison we extract an impact into the organizational processes that are running within a company or organization. In addition we evaluate safety integrity items and we place them in a separate column covering the three horizontal layers.
- **Realm of Organisational Processes:** this layer is mainly focused on the identification of ongoing processes inside an organization. All these processes can be explicitly defined or implicitly performed. Practitioners are interested in this layer and they need to have a clear overview of how the results of the mapping have an impact into their processes. They should take into account their business objectives defined previously and their safety integrity levels (SIL) that are related to specific practices.
- **Realm of Evidences:** there are a wide set of different evidences in a company and most of them can be assessed different times and with respect to different models or regulations. Some of them require that a specific set of work products should be generated for specific practices or they are just suggested. This is the case for the ISO26262 [21] where some work products are required or suggested. Safety integrity levels have an impact at this level in order to keep traceability between processes, artefacts and their SIL. This layer is controlled by practitioners and they need to have a clear understanding and trace of their current generated products, their evidences and of their originating process.

The left side of Fig. 2 describes the aforementioned three layers and the right side of this same figure describes the two different points of views: technology experts and practitioners. At each layer we need to perform a process containing a set of activities that should be carried out. This paper only covers the process inside the Realm of reference technologies for the sake of simplicity. It is relevant to mention that Safety Integrity Level vertical layer represents ASIL from ISO26262 [21], SIL from ISO61.508 [15] and DO178B [26] criticality levels.

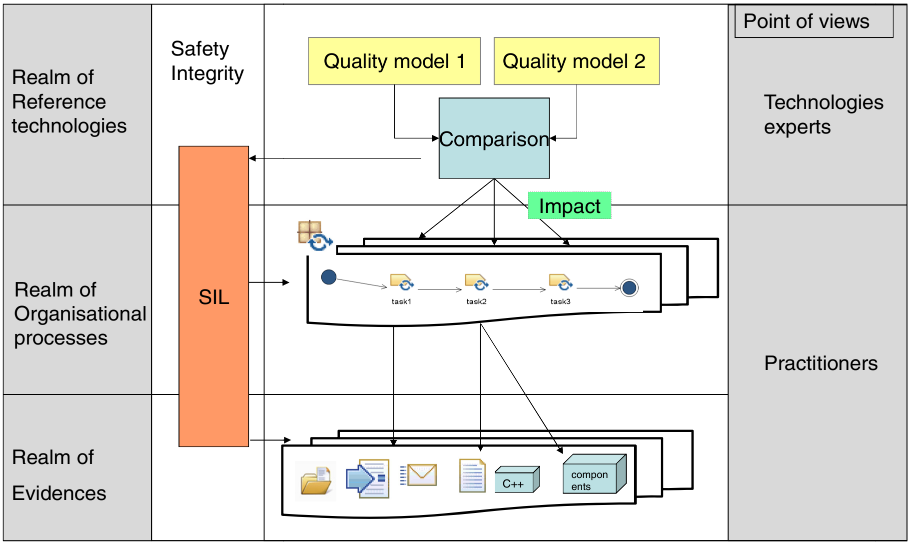


Fig. 2. Overview of the harmonized multimodel framework

This reference technologies process is described in Fig. 3 and it defines the following activities:

- Multimodel Analysis: This activity is focused on a detailed analysis of each reference model. Firstly of all we need to read carefully each model identifying requirements for each area defined within the model. Secondly we extract a set of required and suggested work products. Thirdly we identify potential activities. Finally we associate the appropriate SILs.
- Multimodel Comparison: the most widely used method is by mapping. Several works use this technique as described in [2, 3, 10].
- Multimodel Results: As result of this comparison we extract the areas that are and are not covered by one model with respect to the other.
- Multimodel Recommendations: Based on the previous results we can provide a set of recommendations containing potential work products, activities and SILs.

Fig. 3. Reference technologies harmonization process

4 From a Practical Point of View

The aforementioned framework has been developed and generalized from a road vehicle environment that it is a safety critical system. In this context different models and regulations are applied in order to assure functional and non-functional aspects of the components involved in a road vehicle construction process. ISO26262 is applied in our case and the safety integrity levels are adapted to Automotive Safety Integrity Levels. Our environment combines this standard and +SAFE and CMMI-DEV. Our harmonization process is described in Fig. 3 and in this section we tailor it to automotive settings.

4.1 Multimodel Analysis

As mentioned before this activity is focused on a detailed analysis of each reference model. Firstly we have identified for each model the appropriate granularity for the comparison. ISO26262 defines clauses and subclauses and they represent the requirements that should be fulfilled. +SAFE and CMMI are defined using specific and general goals. Each goal defines a set of practices in order to achieve each goal. Therefore goal is the concept used as requirement for fulfilling with a process area. Secondly we extract a set of required and suggested work products Fig. 4. Thirdly we identify potential activities. Finally we associate the appropriate SILs.

Clause	Work products
Overall safety Management	Organization-specific rules and processes for functional safety Evidence of competence Evidence of quality management
Safety Management during item development	Safety plan Project plan Safety case Functional safety assessment plan Confirmation measure reports
Safety Management after release for production	Evidence of field monitoring

Fig. 4. Work products extracted from ISO26262 Management of Functional Safety

4.2 Multimodel Comparison

The most widely comparison method used is mapping [2, 3, 10]. Several works use this technique as described in [10]. The comparison method is from ISO26262 to CMMI+SAFE. We compare not only clauses from ISO26262 versus goals and their practices from CMMI+SAFE. We also compare required/suggested work products between these models. A discrete scale combining practices and work products is defined in order to provide a clear understanding of how much CMMI+SAFE is covered by ISO26262.

Table 2. From ISO26262 to +SAFE work products coverage

			SAFETY MANAGEMENT				
	From ISO26262 to +SAFE	CMMI+SAFE goals	Develop Safety Plans				Monitor Safety Incidents
		specific practices	S.P.1.1	S.P.1.2	S.P.1.3	S.P.1.4	S.P.2.1
		Work products	1. Requirements source lists 2. Requirements categoris list 3. Safety requiremetns specification 4. Product requirements specification 5. Safety requirements trace	1. Safety criteria 2. Safety strategy	1. Project organisation chart and responsibility allocation matrix 2. Project safety plan	1. Safety plan 2. Certification plan 3. Safety verification plan 4. Safety validation plan 5. Independent safety assessment plan 6. Safety acceptance plan 7. Safety staff skills and experience matrix 8. Safety training plan	1. Minutes of meetings (e.g., of the safety management group) 2. Updated project safety plan 3. Updated hazard analysis 4. Updated safety case 5. Updated hazard log 6. Incident reports 7. Change requests
	Clause	Coverage	L	L	S	S	S
Management of Functional Safety	Overall safety Management	1.Organization-specific rules and processes for functional safety 2. Evidence of competence 3. Evidence of quality management	L	S	P	L	-
	Safety Management during item development	1. Safety plan 2. Project plan 3. Safety case 4. Functional safety assessment plan 5. Confirmation measure reports	P	L	P	L	-
	Safety Management after release for production	1. Evidence of field monitoring	P	-	-	L	S

These values result dividing the sum of expected work product in ISO26262 that can be matched to an expected work product in CMMI+SAFE by the total number of expected work products in CMMI+SAFE.

The scale is defined as follows in based on the existing evidences:

- Strongly (S) related (86–100%): There are several evidences between these two elements.
- Largely (L) related (51–85%): There are some evidences between these two elements. It is recommended to evaluate the existing gaps.
- Partially (P) related (16–50%): There are a few evidences between these two elements. It is recommended to evaluate seriously the existing gaps.
- Weakly (W) related (1–15%): There is minimal evidence. We need to reformulate how to fulfill with these work products.
- No evidence (-) (0%)

Table 2 shows a chunk of the mappings that have been done for comparing work products suggested/required from ISO26262 to those artifacts suggested by +SAFE model. This mapping only establishes a relationship between Management of Functional Safety described in ISO26262 and Safety Management described in +SAFE. It is interesting to note that there is a strong coverage in the management area. Table 3 establishes a relationship between clauses from ISO26262 and goals in +SAFE. The scale used is the same that for work products and the function used for calculating these values is the following: dividing the sum of fulfilled practices of CMMI+SAFE with respect to ISO26262 by the total number of expected practices in CMMI+SAFE This table shows percentage of coverage instead of "strong", "largely"," partially", "weakly" and "no evidence", in order to show the resulting percentages.

Table 3. From ISO26262 clauses to +SAFE goals coverage

			+SAFE							
		ISO26262 \ CMMI +SAFE	Project Management			Engineering - safety				
			Safety Management			Safety Engineering				
			Develop Safety Plans	Monitor Safety Incidents	Manage Safety Related Suppliers	Identify Hazards, Accidents and sources of hazards	Analyse hazard and Perform Risk Assessments	Define and maintain safety requirements	Design for safety	Support safety acceptance
			SP 1.1 Determine Regulatory Requirements, Legal Requirements, and Standards SP 1.2 Establish Safety Criteria SP 1.3 Establish a Safety Organization Structure for the Project SP 1.4 Establish a Safety Plan	SP 2.1 Monitor and Resolve Safety Incidents	SP 3.1 Establish Supplier Agreements That Include Safety Requirements SP 3.2 Satisfy Supplier Agreements That Include Safety Requirements	SP 1.1 Identify Possible Accidents and Sources of Hazards SP 1.2 Identify Possible Hazards	SP 2.1 Analyze Hazards and Assess Risk	SP 3.1 Determine Safety Requirements SP 3.2 Determine a Safety Target for Each Safety Requirement SP 3.3 Allocate Safety Requirements to Components	SP 4.1 Apply Safety Principles SP 4.2 Collect Safety Assurance Evidence SP 4.3 Perform Safety Impact Analysis on Changes	SP 5.1 Establish a Hazard Log SP 5.2 Develop a Safety Case Argument SP 5.3 Validate Product Safety for the Intended Operating Role SP 5.4 Perform Independent Evaluations
			75.00%	100.00%	75.00%	100.00%	100.00%	85.00%	100.00%	80.00%
		Coverage	S	S	S	S	S	S	S	S
ISO26262	Management of functional safety	Overall safety Management	75.00%							25.00%
		Safety Management during item development	25.00%	100.00%						25.00%
		Safety Management after release for production	25.00%		100.00%					25.00%
	Concept Phase	Item definition	25.00%	100.00%		100.00%			33.00%	25.00%
		Initiation of the safety lifecycle							33.00%	25.00%
		Hazard analysis and risk assessment					100.00%		33.00%	25.00%
		Functional safety concept	25.00%					100.00%		25.00%
	System Level	Initiation of product development at the System level	25.00%			100.00%	100.00%			25.00%
		Specification of the technical safety requirements	25.00%					100.00%		25.00%
		System design							100.00%	25.00%
		Item integration and testing				100.00%		33.00%	100.00%	25.00%
		Safety validation				100.00%		33.00%	100.00%	25.00%
		Functional safety assessment				100.00%		33.00%	100.00%	25.00%
		Release for production				100.00%		33.00%	100.00%	25.00%
	Hardware level	Initiation of product development at the Hardware level	25.00%			100.00%	100.00%		100.00%	25.00%
		Specification of hardware safety requirements	25.00%					100.00%	100.00%	25.00%
		Hardware design							100.00%	25.00%
		Hardware architectural metrics							100.00%	25.00%
		Evaluation of violation of the safety goal due to random HW failures				100.00%		33.00%	100.00%	25.00%
		Hardware integration and testing				100.00%		33.00%	100.00%	25.00%
	Software Level	Initiation of product development at the Software Level	25.00%			100.00%	100.00%		100.00%	25.00%
		Specification of software safety requirements	25.00%					100.00%		25.00%
		Software architectural design							100.00%	25.00%
		Software unit testing							100.00%	25.00%
		Software integration and testing							100.00%	25.00%
		Verification of software safety requirements				100.00%		33.00%	100.00%	25.00%
	Production and operation	Production							33.00%	25.00%
		Operation, service (maintenance and repair) and decommissioning			50.00%				33.00%	25.00%
	Supporting processes	Interfaces within distributed developments			100.00%				33.00%	
		Specification and management of safety requirements						100.00%	33.00%	25.00%
		Configuration management							33.00%	25.00%
		Change management							33.00%	
		Verification of software safety requirements							33.00%	25.00%
		Documentation							33.00%	25.00%
		Qualification of software tools							33.00%	
		Qualification of software components							33.00%	
		Qualification of hardware components							33.00%	
		Proven in use argument							33.00%	25.00%

4.3 Multimodel Results

As result of this comparison we extract the areas that are and are not covered by one model with respect to the other. We associated this activity to the traditional gap analysis. It is relevant to note that the results between "clauses versus goals" (Table 3) and "among work products" (Table 2) are approximately the same except for a small delta coming from the functions used for the comparison. The following Fig. 5 provides an overview of the results.

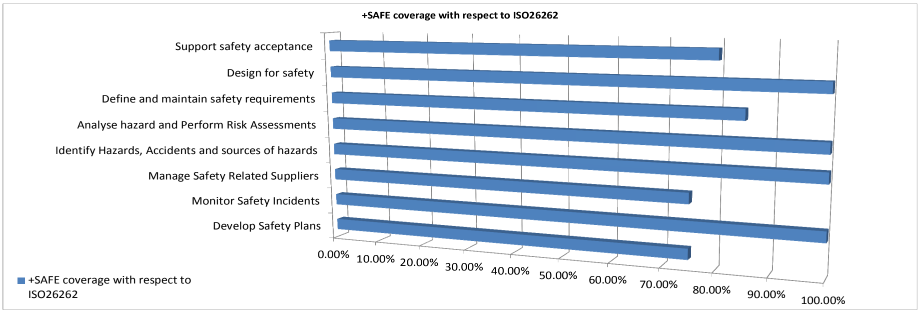


Fig. 5. +SAFE coverage with respect to ISO26262

4.4 Multimodel Recommendations

Based on the gap analysis we recommend a set of activities and work products with their associated ASIL. Most of the best practices and work products are in some way defined using ISO26262. In fact this ISO standard is a large a really complete standard for safety assurance. However in order to fulfill with +SAFE we need to explicitly address the following activities:

- Support safety acceptance
- Define and maintain safety requirements
- Manage Safety Related Suppliers
- Develop Safety Plans

For example +SAFE emphasizes the suppliers management. However ISO26262 does not cover it in the same way, so it is recommended to perform additional activities.

5 Conclusions and Further Work

Software process improvement (SPI) initiatives can also be applied in safety critical systems developments. These are complex environments where a wide set of models, regulations and standards coexist at the same time, and it is not always evident how to start a SPI initiative. This paper introduces a harmonized framework for safety environments analyzing ISO26262 to +SAFE model, and it helps organisations to launch this kind of initiatives. This approach is useful in several situations but especially if we want to identify what are the recommended activities or the process areas that are already fulfilled when we are using ISO26262 and we want to adopt +SAFE model. It is relevant to stress that this study has been done also for the vice versa situation: from +SAFE to ISO26262. In fact our framework does not just take into account practices but also suggested/recommended work products comparisons. We have left intentionally ASIL comparison aside because +SAFE does not specify this kind of safety integrity levels, and it is not relevant for our case to add figures representing a single column for ASILs. As further work a comparison between ASIL from ISO26262 [21], SIL from ISO61.508 [15] and DO178B [26] criticality levels will be performed using this framework. In addition we would like to mention that some works related to [25] will be also carried out in a near future. Therefore the research questions state in the introduction have been solved.

One limitation of this framework is that we have used a mapping technique for the comparison method. In this sense there are some problems associated with subjectivity when we want to compare elements whose granularity differs from each other. Further work it will be related also in this sense.

Acknowledgement. The research leading to these results has received funding from the FP7 programme under grant agreement n° 289011 and from the Basque Country government (programa de perfeccionamiento y movilidad del personal investigador - resolución MV-2011-2-42).

References

1. Cater-Steel, A., Tan, W.-G., Toleman, M.: Challenge of adopting multiple process improvement frameworks. In: 2010 Second International Conference on Computer Engineering and Applications (2010)
2. Latif, A.A., Din, M.M., Ismail, R.: Challenges in adopting and integrating ITIL and CMMi in ICT Division of a Public Utility Company. In: 2010 Second International Conference on Computer Engineering and Applications (2010)
3. Rozman, I., Horvat, R.V., Gyorkos, J., Hericko, M.: PROCESSUS – Integration of SEI CMM and ISO quality models. Software Quality Journal 6, 37–63 (1997)
4. Kirwan, P., Siviy, J., Morley, J., Marino, L.: Maximizing your Process Improvement ROI through Harmonization, `http://goo.gl/1oog7`
5. Siviy, J., Kirwan, P., Marino, L., Morley, J.: The Value of Harmonizing Multiple Improvement Technologies: A Process Improvement Professional's View (May 2008), `http://goo.gl/PIBwT`
6. Siviy, J., Kirwan, P., Marino, L., Morley, J.: Strategic Technology Selection and Classification in Multimodel Environments, `http://goo.gl/WJXtV`
7. Kirwan, P., Siviy, J., Marino, L., Morley, J.: Improvement Technology Classification and Composition in Multimodel Environments., `http://goo.gl/iSaZR`
8. Kirwan, P., Siviy, J., Marino, L., Morley, J.: Process Architecture in a Multimodel Environment, `http://goo.gl/l6a9cSa`
9. Kirwan, P., Siviy, J., Marino, L., Morley, J.: Implementation Challenges in a Multimodel Environment, `http://goo.gl/AcUlM`
10. Baldassarre, M.T., Caivano, D., Pino, F.J., Piattini, M., Visaggio, G.: Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application. Software Qual. J., doi:10.1007/s11219-011-9154-7
11. Voas, J.: Assuring Software Quality Assurance. IEEE Software (2003)
12. Harju, H., Lahtinen, J., Ranta, J., Nevalainen, R., Johansson, M.: Software Safety Standards for the Basis of Certification in the Nuclear Domain. In: 2010 Seventh International Conference on the Quality of Information and Communications Technology (QUATIC) (2010), ISBN: 978-1-4244-8539-0, doi:10.1109/QUATIC.2010.22

13. Maibaum, T., Wassyng, A.: A Product-Focused Approach to Software Certification. Computer 41(2), 91–93 (2008), doi:10.1109/MC.2008.37
14. Software Engineering Institute, +SAFE, V1.2 A Safety Extension to CMMI-DEV, V1.2 (March 2007), http://www.sei.cmu.edu/reports/07tn006.pdf
15. IEC 61508-. International Standard. Functional safety of electrical/electronic/programmable electronic safety-related systems Edition 2 (April 2010)
16. Tuohey, W.: Benefits and Effective Application of Software Engineering Standards. Software Quality Journal 10, 47–68 (2002)
17. Trienekens, J., Kusters, R., Van Solingen, R.: Product Focused Software Process Improvement: Concepts and Experiences from Industry. Software Quality Journal 9, 269–281 (2001)
18. Software Engineering Institute. Capability Maturity Model Integration for development v1.3, http://www.sei.cmu.edu/reports/10tr033.pdf
19. Software Engineering Institute. +SAFE. A Safety Extension to CMMI-DEV, V1.2, http://www.sei.cmu.edu/reports/07tn006.pdf
20. ISO9001:2008. Quality management systems – Requirements, http://www.iso.org/iso/catalogue_detail?csnumber=46486
21. ISO26262. Road vehicles – Functional safety – Part 1: Vocabulary, http://goo.gl/3w9mw
22. Hall, T., Rainer, A., Baddoo, N.: Implementing software process improvement: an empirical study. Software Process Improvement and Practice (2002)
23. Paulk, M., Weber, C., Curtis, B., Chrissis, M.: A High Maturity Example: Space Shuttle Onboard Software. In: The Capability Maturity Model: Guidelines for Improving Software Process. Addison-Wesley (1994)
24. Niazi, M., Wilson, D., Zowghi, D.: A maturity model for the implementation of software process improvement: an empirical study. Journal of Systems and Software - Special Issue: The New Context for Software Engineering Education and Training 74(2) (2005)
25. ISO/IEC 15504. Information technology – Process assessment, http://goo.gl/HSzLe
26. DO178 Software Considerations in Airborne Systems and Equipment Certification. RTCASC-167 EUROCAE WG-12
27. Paulk, M.: A History of the Capability Maturity Model for Software, http://goo.gl/q3CsK

Towards the Harmonization of Process and Product Oriented Software Quality Approaches

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. Software organizations are currently required to implement more than one software process improvement model concurrently. Several multimodel initiatives have appeared to support this situation, and existing proposals address integration from the process perspective, considering models such as CMMI, ISO 90003, ISO/IEC 12207, and ISO/IEC 15504. These efforts attempt to understand how to integrate process focused models in order to optimize resources and obtain the expected benefits. However, as the eventual aim of process improvement is to improve software product quality, it is also important to consider product quality models in harmonization efforts. In this paper, the result of mapping models based on both, (process and product) quality perspectives, is presented. The method used is also briefly described and applied to map ISO/IEC 25010 onto CMMI-DEV and ISO/IEC 12207. The result shows that process oriented improvement models consider product quality characteristics during the early stages of the software development life cycle, and that process improvement initiatives can therefore be driven by product quality improvement goals.

Keywords: product oriented quality approach, process oriented quality approach, harmonization, mapping, CMMI-DEV, ISO/IEC 12207, ISO/IEC 25010.

1 Introduction

Software organizations must confront diverse challenges if software products are to be developed efficiently. Market pressures, customers' needs, government regulations, and certifications are business drivers that organizations should consider when adopting quality standards, reference models, guidelines, or recommended practices [1]. Each improvement model has its particular goals, structure, granularity, and application domain. Software organizations must therefore deal with different improvement models simultaneously, and cope with the similarities and differences among them.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 133–144, 2012.

Various works addressing the multimodel problem [1, 2], i.e., when organizations use more than one quality approach simultaneously, have appeared. For instance, Pardo et al. [3, 4] propose a framework that provides the conceptual, methodological and technological support needed to facilitate the harmonization of multiple models. Harmonization is therefore an approach with which to integrate improvement models in order to achieve particular business objectives.

Harmonization initiatives have, to date, been focused on process oriented quality approaches. In the systematic review reported in [5], 60% of the primary studies selected (32 papers) analyze both reference models, such as ISO 9001, CMM, CMMI, and assessment models, such as ISO 15504 and SCAMPI. Kelemen et al. [2] analyze 78 papers, the majority of which deal with software process improvement (SPI) and quality approaches such as CMMI, SPICE, and ISO 9001. When models rely on the same quality approach, they may share vocabulary and structures that would ultimately assist in model integration.

Quality is a complex multidimensional concept that allows diverse research approaches [6] and software quality standards to be classified in two main categories: product and process [7]. However, few works have undertaken the integration of both process and product oriented perspectives from the viewpoint of harmonization.

A fundamental step in a harmonization strategy is how to compose the models [1, 2]. Model integration is usually carried out without coordination, and the organizational performance is overlooked when new technology is integrated. Quality approaches must be compared before a model can be composed. Various procedures with which to compare models as bilateral mappings or needs mappings exist [8], but low level comparison requires an understanding of both the structural differences of models, and the level of granularity, and this can make the comparison difficult [9].

In this paper we present the harmonization result of process and product quality models in order to support organizations when carrying out their process improvement programs driven by the desired product quality characteristics. A method with which to systematically carry out the mapping of models is also provided. This method has been applied in order to compare ISO/IEC 25010 [10] as a quality product software model with ISO/IEC 12207 [11] and CMMI-DEV [12], which are focused on the process perspective.

The remainder of this paper is structured as follows: Section 2 describes the works existing in literature that concern the integration of product and process quality perspectives. In Section 3 the method proposed to establish mapping in process and product oriented standards is presented. Section 4 describes how the method was applied to map ISO/IEC 25010 onto CMMI and ISO/IEC 12207; and finally, Section 5 shows our conclusions and future work.

2 Related Work

Few papers consider the alignment of process and product oriented quality approaches within an SPI effort. Therefore, there is insufficient evidence to establish causal relationships between the models used and the results obtained [13-15]. Balla et al.

[16] proposed a framework to show how quality standards and quality models can be used when a company is searching for enhanced software quality. The authors link process, product and resources with process definitions, quality properties and measures. However, there is a lack of information in the papers as to how the quality models were integrated or composed.

Some researchers have focused on the mappings of process oriented and product oriented standards. Ashrafi [17] analyzes the impact of CMM and ISO 90003 on software product quality characteristics. Her results suggest that SPI initiatives contribute towards enhancing different product quality characteristics, and she creates a decision tree to show the relationships between reference models and quality characteristics. Pardo et al. [18] carry out a mapping of CMMI-Dev 1.2 and ISO 90003 in a search for the coverage levels of software product characteristics and sub-characteristics as they are described in ISO/IEC 25010. These authors also propose a decision tree with which to choose a reference model when organizations consider particular software product quality. Finally, Al-Qutaish [19] analyzes the cross-reference between ISO/IEC 9126 and ISO/IEC 12207, and subsequently obtains a mapping of recommended measures, whilst noting some inconsistencies between standards.

Harmonization must deal with two main issues: improvement model selection and the implementation of the improvement model [2]. We focus on the first issue, given that comparing standards is a complex task owing to their terminology, structure and granularity [20]. In this context, mapping is a commonly used means to compare models. Although different types of mappings have been reported [8], some of them overlook finer granularity levels, which could result in requirements being met on one level of granularity, but not on another [21].

The benefit of low-level mapping is traceability back to source approaches, and could be considered as a basis for multimodel comparison. For instance, Pino et al. [22] define a mapping process with which to carry out a step-by-step comparison of diverse models, and they report the task required to achieve a detailed matching between models. Mappings are also useful for model composition [9].

3 Comparison Method

Such as stated in the previous section, the existing works in literature do not address how to systematically compare both product and process perspectives of quality models in order to support their integration. The purpose of the method proposed herein is to carry out a mapping between a product oriented quality model and process oriented quality model in order to identify the corresponding elements contained in them.

Improvement models are focused solely on a particular view of the process, and this view has a relationship with the process elements. Fig. 1 shows the elements usually considered when modeling a process [23]. Processes, activities and tasks are concepts related to the actions or best practices that an organization could implement.

Process models can be represented at diverse levels of granularity and usually express a specific view of interest, such as an activity model or a product model [23]. In process oriented models, activities are highlighted while other process elements are only listed or briefly described. In contrast, product quality approach models are interested in the description of the characteristics that are considered relevant to enhance product quality, without addressing how they can be achieved during software development or operation. Others analyses of different quality perspectives can be found in [6, 7, 24].

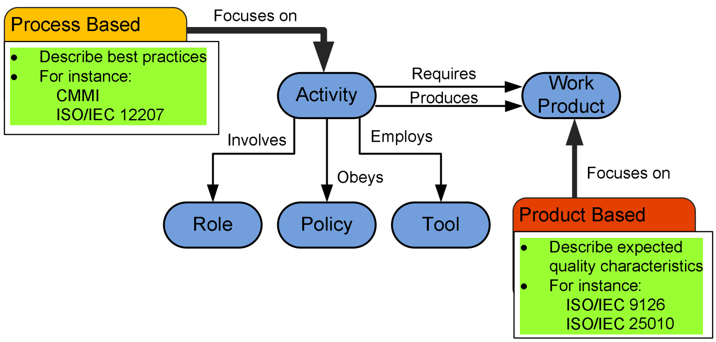


Fig. 1. Quality approaches linked to process elements (Adapted from [23])

With regard to the above, it was necessary to compare quality models that are focused on different entities: activities and products. This was done by carrying out a mapping, which was undertaken by adapting the method described by Pino et al.[22]. The update obeys the evaluation of two different quality perspectives. Fig. 2 shows the activities and tasks that are necessary to map quality models. The activities are:

1. Analyze models. The purpose of this is to understand the improvement models' goals, structure, and requirements. Since we are working with two quality perspectives, one of them can be the source for search terms, whilst the other can be the target of the mapping. Task required: Describe product oriented model and Describe process oriented models.

2. Design mapping: The purpose of this is to set out a comparison procedure and design mapping templates. Task required: Establish comparison levels and Define comparison scale.

3. Execute mapping: The purpose of this is to apply a comparison procedure in order to achieve mapping results. Task required: Carry out comparison.

4. Assess mapping results: The purpose of this is to report issues that must be considered in a harmonization initiative. Task required: Analyze findings.

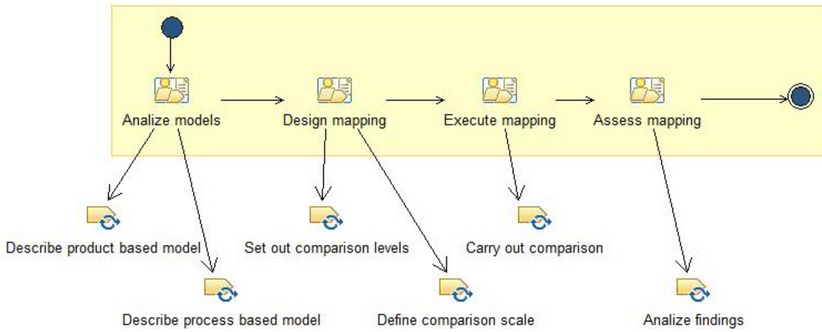


Fig. 2. Mapping method used to compare process oriented and product oriented quality approaches

4 Mapping ISO/IEC 25010 onto Both CMMI and ISO/IEC 12207

This section summarizes the results obtained after carrying out the mapping process described above. It is applied to ISO/IEC 25010, CMMI-DEV and ISO/IEC 12207.

4.1 Analyze Models

ISO/IEC 25010

The ISO/IEC 25010 [10] has a software product quality model that is hierarchical in nature, and which documents the most relevant characteristics and sub-characteristics that software must have. One of the principles leading to its development was that a product comprehensive specification and evaluation are essential to produce high quality software. It is also supposed that software systems have different audiences, and that each of them perceives the level of product quality differently.

The ISO/IEC 25010 quality models can be used to identify relevant quality characteristics which can then be used to establish the requirements, criteria of achievement, and measurement. The quality in use model is composed of five characteristics: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. These could be applied to software products in the operation stage. The product quality model is, meanwhile, composed of eight characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. These can identify the static or dynamic properties of software at the development stage.

CMMI for Development, Version 1.3

The CMMI focuses on improving processes in an organization [12]. The CMMI-DEV contains 22 process areas. Each process area has set goals and expected practices.

CMMI model components are grouped into three categories that reflect how to interpret them. The components required must be visibly implemented to achieve specific and generic goals. The expected components guide the implementation of improvements and are related to specific and generic practices. Lastly, informative components assist users to understand either required or expected components.

The process areas are organized into four categories: process management, project management, engineering and support. In this paper we analyze the key areas of the engineering category, since they describe the practices associated with the development of software products. The process areas that are included in this category are: requirements development (RD), technical solution (TS), product integration (PI), verification (VER) and validation (VAL).

ISO/IEC 12207:2007

The ISO/IEC 12207:2007 [11] contains a specification of the processes, activities and tasks that are applied throughout the life cycle of a software product, from acquisition to retirement. The requirements are marked with the word "shall", the recommendations with "should" and the permitted practices with "may".

In this paper we shall focus on the technical processes of system context and software implementation sub-processes since they are associated with the tasks, context and capabilities that should show how the software is developed and deployed in a target environment. The technical processes are used to define the requirements for a system, the product development, use, maintenance and system retirement. Software implementation processes are used to develop a software element. This includes those processes that transform the specified behavior, interfaces and implementation constraints in implementing actions which result in a software element that satisfies the requirements derived from system requirements.

4.2 Design the Mapping

The ISO/IEC 25010 product quality model for software systems is made up of eight characteristics. Overall, the product quality model has 30 sub-characteristics, whilst the quality in use model consists of five characteristics and 11 sub-characteristics. The names of all of these characteristics and sub-characteristics, and their respective descriptions, will be the keywords searched for in selected parts of the CMMI and ISO/IEC 12207.

It should be noted that both the CMMI and ISO/IEC 12207 models have mechanisms that the organization can implement in order to choose the methods, tools and techniques that are suitable for software development. Moreover, when addressing the requirements specification, the software quality characteristics are called non-functional requirements. This revision therefore focuses solely on the explicit statements which mention quality characteristics or categories of such characteristics. These statements are included in the description of activities, processes and practices of the selected process categories of both models.

The procedure used to map software quality characteristics is based on the objective identification of the concepts associated with them in the process and practices descriptions. Furthermore, the matched statement is reviewed to set out its localization in the hierarchical structure of the process model. The steps needed to evaluate the models are:

1. Identify the search terms. In this case we wish to know the software quality characteristics. This can be achieved by referring to the following terms: features, dimensions, characteristics and quality attributes. Secondly, other concepts that are relevant in the search are those associated with the concepts of non-functional requirements and quality requirements. ISO/IEC 25010 additionally provides a list of software quality characteristics and sub-characteristics.
2. Localize the statement in the structure of the process models and categorize the relevance of the practice. The process model statements which contain the search terms are identified and classified as: required (R), important (I), or auxiliar (A). The assigned category relies on the process model element type.
3. Fill in the mapping matrix. For each match, add an item to the mapping matrix linking the process elements and set out the relevance of the statement. Finally, add the contribution of model elements in each category. An example of this is shown in Table 1.

4.3 Execute the Mapping

This process was applied to set out the mapping of the product-based quality approach, ISO/IEC 25010, and the process-based quality approach of both the CMMI-DEV1.3 and ISO/IEC IS 12207:2007 models. Only those components in the technical category were analyzed.

The search terms identified in the previous section allowed statements to be discovered in CMMI with regard to the model structure. For instance, Table 1 shows process area Product Integration, and lists the type of statement, description and the score assigned. In this case, the three references to software product quality can be found in sub-practices and informative notes.

As can be observed in the score in Table 2, the key process areas with the greatest number of components containing quality characteristics are requirements development and technical solution. There is only one specific practice in requirements development that expresses the importance of product quality characteristics. No goal, as stated in the model, includes references to software quality attributes. The informative notes and some sub-practices address the quality attributes. However, they do not have the relevance of a specific goal. For the technical solution process area, informative notes emphasize the need for architectural design and evaluation based on relevant quality attributes.

The ISO/IEC 12207, meanwhile, has 18 processes in the technical category which are grouped into two subcategories: technical processes and implementation processes. For instance, Table 3 shows the mapping of software quality characteristics onto the system requirements analysis process. There are two references. The first shows that the system specification must have a section of non-functional requirements that are relevant to the problem to be solved. The second is a requirement for a formal description of quality requirements.

Table 1. Statement linked to software quality characteristics of Integration Product process area. Labels: (R)equired, (I)mportant, (A)uxiliary

Element type	Description	R	I	A
Informative note SP 1.3. Establish procedures … product integration	Criteria can be defined by their **behavioral performance(functionality and quality attributes**)	0	0	1
Subpractice SP 2.1. Review … for completeness	Subpractice 1. Review interface data for completeness and ensure **complete coverage of all interfaces**.	0	0	1
Informative note SP 3.4. Package … product or product component	Sub-practice 3. Satisfy the applicable requirements and standards for packaging and delivering products. Examples of requirements: **safety,…, security, …**	0	0	1
Total		**0**	**0**	**3**

Table 2. Software product quality in CMMI

Process Area	Required	Important	Auxiliary
Validation (VAL)	0	0	1
Verification (VER)	0	0	1
Product integration (PI)	0	0	4
Technical solution (TS)	0	0	18
Requirements development (RD)	0	1	22
Total	0	1	46

In the technical processes, only two clauses were found that were considered as relevant quality requirements for certification purposes. In contrast, the implementation processes include multiple references to quality characteristics, of which we can highlight the software requirements analysis process with four clauses (Table 4). The ISO/IEC 12207, in the notes associated with certain tasks, recommends using appropriate standards to achieve the desired level of quality characteristics. For example, if usability is an important aspect of the development project, then the notes point to standards such as ISO/IEC 9241, ISO/IEC 13407 and ISO/IEC 18152.

4.4 Assess Mapping Results

After carrying out the comparison of these models, various facts can be noted. First, the practices and processes which point to tasks that directly describe quality characteristics correspond to the system and software analysis stage (Fig. 3 and 4). Second, the system design stage in CMMI only has a few notes regarding the consideration of the quality characteristics that have an impact on the software architecture (Fig. 3). In the case of ISO/IEC 12207 the consideration is lower in frequency. The remaining stages of the life cycle contain few references to the quality characteristics. Third, there were few specific references to key aspects of individual quality characteristics. Safety, security, usability, maintainability and performance are explicitly addressed, mainly in the descriptions of the examples. Other cases only address general terms, such as quality attributes, quality attribute requirements or non-functional requirements. Lastly, the structure and level of abstraction of the models studied make the comparison of findings difficult.

Table 3. Statement linked to software quality characteristics of system requirements analysis process. Labels: (R)equired, (I)mportant, (A)uxiliary.

Element type	Description	R	I	A
Note 6.4.2.2. Outputs	A defined set of system functional and **non-functional requirements** describing the problem to be solved are established.	0	0	1
Requirement 6.4.2.3.1.1 The specific intended use of the system to be developed.	The system requirements specification shall describe: ….**user requirements; safety, security, human-factors engineering, interface, operations, and maintenance requirements**…	1	0	0
Total		**1**	**0**	**1**

Table 4. Software product quality in ISO/IEC 12207

Processes: Technical context and Software implementation	Required	Important	Auxiliary
Stakeholder reqs. definition 6.4.1	1	0	4
System requirements analysis 6.4.2	1	0	1
System architectural design 6.4.3	0	0	3
Implementation 6.4.4	0	0	0
System integration 6.4.5	0	0	0
System qualification testing 6.4.6	0	0	0
Software Installation 6.4.7	0	0	0
Software acceptance support 6.4.8	0	0	0
Software operation 6.4.9	0	0	0
Software maintenance 6.4.10	0	0	1
Software disposal 6.4.11	0	0	1
Total	**2**	**0**	**10**
Software implementation 7.1.1	0	1	0
Software requirements analysis 7.1.2	4	0	3
Software architectural design 7.1.3	0	0	0
Software detailed design 7.1.4	0	0	0
Software construction 7.1.5	0	0	0
Software integration 7.1.6	0	0	1
Software qualification testing 7.1.7	0	0	0
Total	**4**	**1**	**4**

The comparison is the first step in identifying improvement areas in a multimodel SPI initiative. CMMI and ISO/IEC 12207 show that they have practices related to enhancing software product quality early in the software development cycle. However, there is a lack of activities and tasks focused on quality software characteristics at the end of development. Moreover, the majority of the references are explications concerning the requirements of improvement models. It may therefore be necessary to set out practices throughout the entire development life cycle which will allow the correspondence elements between both quality approaches to be identified.

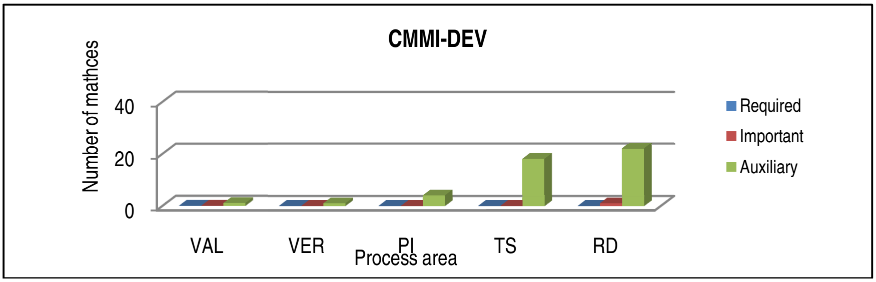


Fig. 3. Findings from CMMI-DEV. Labels: requirements development (RD), technical solution (TS), product integration (PI), verification (VER) and validation (VAL)

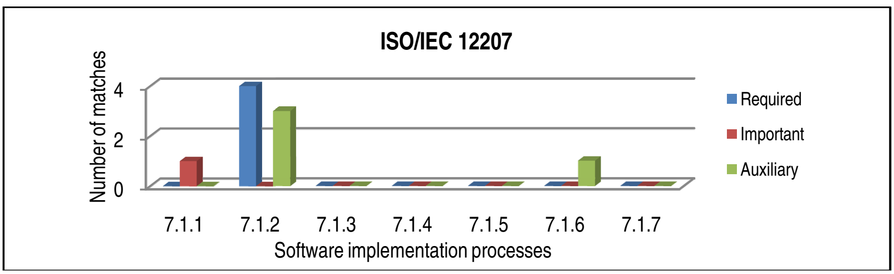


Fig. 4. Findings from ISO/IEC 12207

With regards to the above, a software organization that wished to enhance software product quality would focus on analysis tasks to define the desired quality product characteristics. This could be considered as the first step in the integration of both quality perspectives. It would then be possible to study software analysis methods, as regards organizational process and the desired quality product characteristics, in order to choose the most appropriate practices.

5 Conclusions

Software development organizations are interested in improving their product quality. Some research proposals have addressed both software product quality and process improvement quality approaches when deploying a multimodel SPI initiative. In order to systematically support the comparison of both product and process oriented quality perspectives, a method for mapping has been briefly described, which has been applied to carry out the mapping of a software product quality model (ISO/IEC 25010) onto process oriented quality models (CMMI-DEV and ISO/IEC 12207).

The mapping relies on explicit references to the software quality characteristics and takes into account the structure of the reference model. As a result of this research it was found that the majority of references are located in those processes related to the analysis stage. In the design, CMMI informative notes address software quality characteristics within the definition and evaluation of software architecture. However, in the models studied, there are few references to the quality attributes at the final stages of the life cycle.

Suitable mappings between process and product quality approaches permit integrated quality models harmonizing both perspectives to be built. More successful multimodel SPI initiatives can therefore be conducted in companies, since these companies can focus their process improvement efforts on the practices or areas that may influence the product quality characteristics desired. The cost of SPI programs, and the effort involved in them, can therefore be reduced, and a better focused product improvement can be obtained.

As future work, this comparison method must be validated with other reference models and software product models. In addition, the identification of correspondence parts among models is a step towards the integration of these quality perspectives. An implementation strategy which optimizes resources could then be developed to deploy a successful multimodel SPI initiative which must also be validated in companies by means of case studies.

References

1. Siviy, J., Kirwan, P., Morley, J., Marino, L.: Maximizing your Process Improvement ROI through Harmonization. Software Engineering Institute, Carnegie Mellon (2008)
2. Kelemen, Z., Kusters, R., Trienekens, J.: Identifiyng criteria for multimodel software process improvement solutions - based on a review of current problems and initiatives. Journal of Software Maintenance and Evolution: Research and Practice (2011), doi:10.1002/smr.549
3. Pardo, C., et al.: An ontology for the harmonization of multiple standards and models. Computer Standards & Interfaces 34(1), 48–59 (2012)
4. Pardo, C., Pino, F.J., García, F., Piattini, M., Baldassarre, M.T., Lemus, S.: Homogenization, Comparison and Integration: A Harmonizing Strategy for the Unification of Multi-models in the Banking Sector. In: Caivano, D., Oivo, M., Baldassarre, M.T., Visaggio, G. (eds.) PROFES 2011. LNCS, vol. 6759, pp. 59–72. Springer, Heidelberg (2011)
5. Pardo, C., Pino, F.J., García, F., Piattini Velthius, M., Baldassarre, M.T.: Trends in Harmonization of Multiple Reference Models. In: Maciaszek, L.A., Loucopoulos, P. (eds.) ENASE 2010. CCIS, vol. 230, pp. 61–73. Springer, Heidelberg (2011)
6. Allen, J., Kitchenham, B., Konrad, M.: Theme Q. The relationships between processes and product qualities. In: Forrester, E. (ed.) A Process Research Framework, pp. 19–28. Software Engineering Institute, Carnegie Mellon (2006)
7. Voas, J., Laplante, P.: Standards confusion and harmonization. Computer 40(7), 94–96 (2007)

8. Halvorsen, C.P., Conradi, R.: A Taxonomy to Compare SPI Frameworks. In: Ambriola, V. (ed.) EWSPT 2001. LNCS, vol. 2077, pp. 217–235. Springer, Heidelberg (2001)
9. Ferreira, A., Machado, R., Paulk, M.: Quantitative Analysis of Best Practices Models in the Software Domain. In: Asia-Pacific Software Engineering Conference, APSEC, pp. 433–442 (2010)
10. ISO, ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQauRE) - System and software quality models (2010)
11. ISO, ISO/IEC FDIS 12207 Systems and software engineering - software life cycle processes (2007)
12. SEI. CMMI-DEV, V1.3, Improving processes for developing better products and services. Technical Report, CMU/SEI-2010-TR-033, ESC-TR-2010-033 (2010)
13. Järvinen, J., Komi-Sirviö, S., Ruhe, G.: The PROFES Improvement Methodology - Enabling Technologies and Methodology Design. In: Bomarius, F., Oivo, M. (eds.) PROFES 2000. LNCS, vol. 1840, pp. 257–270. Springer, Heidelberg (2000)
14. Trienekens, J., Kusters, R., van Solingen, R.: Product focused software process improvement: concepts and experiences from industry. Software Quality Journal 9, 269–281 (2001)
15. van Latum, F., van Uijtregt, A.: Product Driven Process Improvement PROFES Experiences at Dräger. In: Bomarius, F., Oivo, M. (eds.) PROFES 2000. LNCS, vol. 1840, pp. 232–243. Springer, Heidelberg (2000)
16. Balla, K., Bemelmans, T., Kusters, R., Trienekens, J.: Quality through Managed Imprvement and Measurement (QMIM): Towards a Phased Development and Implementation of a Quality Management System for a Software Company. Software Quality Journal 9, 177–193 (2001)
17. Ashrafi, N.: The impact of software process improvement on quality: in theory and practice. Information and Management 40, 677–690 (2003)
18. Pardo, C., Pino, F., García, F., Piattini, M.: Harmonizing Quality Assurance Processes and Product Characteristics. IEEE Computer 44(6), 94–96 (2011)
19. Al-Qutaish, R.: Measuring the Software Product Quality Durng the Software Development Life-Cycle: an International Organization for Standardization Standards Perspective. Journal of Computer Science 5(5), 392–397 (2009)
20. Yoo, C., et al.: A unified model for the implementation of both ISO 9001:2000 and CMMI by ISO-certified organizations. Journal of Systems and Software 79, 954–961 (2006)
21. Kirwan, P., Siviy, J., Marino, L., Morley, J.: Improvement Technology Classification and Composition in Multimodel Environments. Software Engineering Institute, Carnegie Mellon (2008)
22. Pino, F., Baldassarre, M.T., Piattini, M., Visaggio, G.: Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504. Journal of Software Maintenance and Evolution: Research and Practice 22, 279–296 (2010)
23. Montangero, C., Derniame, J.-C., Kaba, B.A., Warboys, B.C.: The Software Process: Modelling and Technology. In: Derniame, J.-C., Kaba, B.A., Wastell, D. (eds.) Promoter-2 1998. LNCS, vol. 1500, pp. 1–13. Springer, Heidelberg (1999)
24. Kitchenham, B., Pfleeger, S.: Software quality: the elusive target. IEEE Software 13(1), 112–121 (1996)

Improvement of Task Management with Process Models in Small and Medium Software Companies

Jakub Miler[1] and Hanna Wesołowska[2]

[1] Gdansk University of Technology, Department of Software Engineering, Gdansk, Poland
jakubm@eti.pg.gda.pl
[2] Blue Services Sp. z o. o., Quality Assurance Department, Sopot, Poland
hwesolowska@gmail.com

Abstract. Small and medium software companies exhibit many special features that give reason for a dedicated approach to process improvement. They often cannot afford implementing maturity models or quality standards both in terms of time and money. Instead, they expect simpler solutions that can allow to run projects in more systematic and repeatable way, increase quality and knowledge management. In this paper, we present a method focused on improvement of task management using the process models. The method proposes the integration of modeling and task management tools, where models become templates of enacted projects. We applied the method in two case studies with SMEs, where sample process models were built and enacted in adapted task management tools, followed by a survey. The survey resulted in 82.5% of positive answers. The case studies show considerable potential of our method in solving some improvement problems of SMEs.

Keywords: software process improvement, software process model, SPEM, model enactment, project management, task management, SMEs.

1 Introduction

Many researchers [1-6] and organizations [7-8] acknowledge that software process improvement (SPI) in small companies requires a dedicated approach different from simple tailoring of standards and maturity models such as CMMI [9], ISO 12207 [10], and ISO 15504 [11]. Small and medium enterprises (SMEs) with less than 250 employees [12] and, in particular, very small entities (VSEs) with less than 25 employees [13] explore their advantages such as flexibility, innovativeness, market reaction and managerial agility [14] to achieve their specific key business goals identified in [15]. Application of reference models was perceived by SMEs as harnessing their potential as it involves high costs, long term investment, considerable staff and overall change of organizational culture [16]. Success of SPI in SMEs comes from taking advantage of their specifics which should be carefully taken into consideration [17-19].

According to maturity models such as CMMI, first step of process improvement requires manageability, repeatability and reuse of good practices [9]. To achieve this, large enterprises employ effective but complex and expensive software development

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 145–156, 2012.

and process management tools such as IBM Rational tool suite [20]. In turn, small and medium enterprises (SMEs) commonly use simple marketed or proprietary project and task management tools such as [21] which do not offer any process abstraction and reuse. SMEs follow agile-inspired development culture [22] and define tasks manually from scratch in every new project. Even if SMEs describe their processes, these descriptions often remain ignored or bypassed and end up stored in a handbook such as the Quality System [23] or a piece of software called Electronic Process Guide [24] that nobody (or most of people) bothers to study or use in practice.

We identify a common problem of SMEs as a gap between the definitions of processes and the daily management of tasks related to these processes. As a result, the considerable effort on process definition is wasted, which increases negative attitude to future improvement initiatives. The people and the entire organization learn slower, when the reactive approach to daily work cannot be successfully superseded with a proactive approach based on managed processes. The projects fail to meet the success criteria such as schedule, budget and the quality of products.

The research question is how to facilitate introducing more systematic and repeatable way of project realization and practical usage of good practices in SMEs. Our research is aimed at designing a method of using process definitions in the management of actual projects' tasks in software SMEs. The solution would be to introduce such technique of process definition and their integration with task management that is affordable to SMEs and possible to smoothly introduce into current culture, practice and toolset. This solution should use modern and popular approach to process definition which is supported by easily achievable software tools. Additionally, the definitions should be possible to apply as templates of individual and team tasks specified in currently used task management tools. Such approach facilitates evolutionary change in organizational culture and daily routine, making the process improvement more likely to succeed.

2 Related Work

The systematic reviews of SPI for SMEs presented in [3] and [19] reveal the need for automated tools support to facilitate SPI, but the actual SPI models and techniques discussed do not cover tool-assisted enactment of tasks from the process models. Mishra and Mishra [5] discuss several methods of SPI dedicated to SMEs: Self-diagnosis, SPM Model, ASPE-MSC, PRISMS, and MESOPYME. These methods allow to identify the improvements and specify them as process guides or action packages, but none of them mentions how to implement these recommendations in daily task management. Savolainen et al. [25] present a simplified way of process modeling dedicated to SMEs. They apply wall-charts which are documented in an 'electronic version'. This is used more as an analytical and training tool within SPI rather than a process model for project management. Additionally, Savolainen et al. do not propose any specific software tools.

Friedrich and Bergner [26] define a formal method of deriving actual project's actions from plans in plan-driven process enactment. Although they place process models atop the project plans, the translation of process models to project's actions and the implementation in a software tool remains future work. Yaeli and Klinger [27] focus on enacting only one aspect of a process definition, namely the responsibility assignments of roles to work definitions and work products. Gonzalez-Perez and Henderson-Sellers [28] propose an approach to methodology definition and enactment focusing on work products instead of processes and tasks. As a support, they do not use commonly used tools but provide a proprietary MethodComposer toolset, both for process definition and enactment. The method is intended to improve the software development processes, however no application is discussed, in particular in the context of SMEs. Valiente et al. [29] propose model and tool integration schemes based on ontology covering both software engineering and IT service management processes. However, they do not discuss the application of their approach by SMEs and mention only MS Project as a project management tool overlooking the internet-based agile-like task management tools often used by SMEs.

3 The Method

The proposed method involves description of software development processes with models and using these models in process instantiation and daily management of projects and their tasks. The method is assisted with software tools both for process modeling and task management. It consists of the following elements:

- process modeling,
- process modeling tool and task management tool integration,
- process model enactment,
- process model improvement.

We use process models as templates and guidelines for tasks in actual projects. Among several process modeling paradigms discussed in [30] activity-oriented metamodels satisfy best this goal, as they build upon concepts close to task management. From the activity oriented-metamodels we chose Software Process Engineering Metamodel (SPEM) [31] which is commonly used in description of software processes and is supported by a number of modeling tools.

SPEM defines basic elements of process structure such as disciplines, activities, artifacts and roles. Disciplines group activities in a common area of knowledge e.g. business analysis or testing. Activities describe elements of work to do in the process. Artifacts are work products processed by activities on input, output or both. Roles define skills and competencies of performers of the activities as well as their responsibilities for activities and artifacts. Additionally, the aforementioned model elements are supplemented with guidelines, among which artifact templates and practice descriptions are most important.

Process models are built with dedicated software process modeling (SPM) tools supporting SPEM such as Rational Method Composer [32] or Eclipse Process Framework Composer [33]. To fit in our approach, an SPM tool must offer saving or exporting models into formats that allow to access the model's data (e.g. XML) and creating easily navigated web pages with description of process elements (disciplines, activities, artifacts, roles). We recommend the Eclipse Process Framework Composer (EPF Composer) which is freely available for commercial use and satisfies the above requirements. It addresses the needs of SMEs in terms of affordability, ease of use, flexibility and compatibility with current toolset. EPF Composer does not require complex deployment and substantial training.

Our method assumes that process models are enacted in a task management (TM) tool currently used in a company. Our focus is to respect current practice and people's knowledge and smoothly overlay process definition on daily tasks. As a TM tool we understand a software tool that allows, among others, to add projects, add tasks and subtasks, assign tasks and subtasks to projects, add roles, assign tasks to roles, assign persons to roles, and add descriptions to tasks. JIRA is an example of such tool [21].

To enact the process models from an SPM tool in a TM tool, these tools must be integrated. We integrate tools on the data level. To understand the data model of an SPM tool, it is helpful to create a sample model of a process, save it to the text file format (e.g. XML) and inspect its structure. Then, the data model of the TM tool should be analyzed compared to the SPM tool model structure. As a result, a mapping between objects that store similar information in both data models is built (e.g. tasks in the task model are able to store similar information as activities in the process model). The mapping should also include missing objects (e.g. persons in task model have no equivalent in the process model). Equivalent objects can be transferred from the SPM tool to the TM tool automatically with a translator or import filter; missing ones will have to be added to the TM tool manually. We assume that appropriate translator is already available or can be easily developed as a stand-alone application or an extension to the TM tool. Table 1 presents the mapping of SPEM elements and their representation in EPF Composer models on the concepts used in TM tools.

Table 1. Mapping of elements from SPEM, EPF Composer model and TM tool data model

SPEM	EPF Composer	TM tool
Role	<ContentElement xsi:type="uma:Role">	Role/group
-	-	Employee/user
Task	<ContentElement xsi:type="uma:Task">	Subtask
Discipline	<ContentCategory xsi:type="uma:Discipline">	Task
Artifact	<Attachment> in <ContentElement xsi:type="uma:Template"> in <ContentElement xsi:type="uma:Artifact">	Template
Guideline	subelements of <ContentElement>	Task description

The instances of activities, disciplines and roles created in the TM tool must be supplemented with their descriptions, templates and guidelines. To achieve this, the process model should be published as a website. Task descriptions can then link to this site. This way an employee assigned to a task can easily access guidelines for this task, download templates of documents or read his/her role description. The integration of SPM and TM tools is shown in Fig. 1.

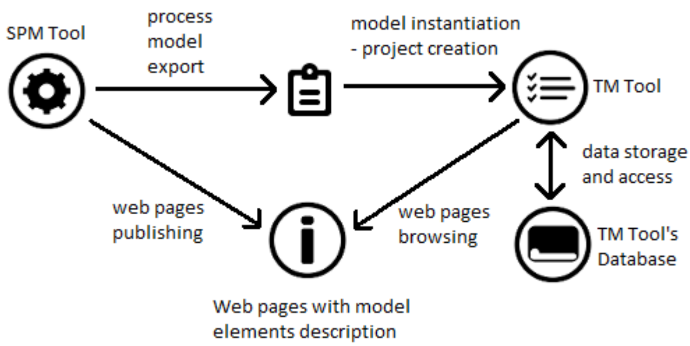


Fig. 1. SPM and TM tool integration

The application of the process model to task management in an actual project requires model enactment. Model is enacted by employees that perform tasks assigned to their roles [26]. To support this, the process model must be instantiated in the TM tool. The TM tool administrator takes the model file and runs the translator which creates an new project where generic model elements become parts of this project (e.g. tasks are created based on modeled activities). These tasks are linked to a process guide exported from the SPM tool, which is made available to employees. Once people are assigned to their roles, they learn their tasks and can start using the TM tool in daily task management. If necessary, they can modify existing tasks or define new ones that can possibly be incorporated back into the model.

The process models must be constantly improved to ensure new projects learn from the past ones in a knowledge management cycle. This requires collecting comments on the process from employees and management, analyzing processes and identifying the needs for change, designing changes, and making changes to models. These activities should be conducted periodically, based on observations of realization of projects carried out in accordance with the models and taking account of changing organization's business needs.

4 Case Studies

The research method to validate our approach involved case studies in representative SME software companies, where we used survey and participant observation to collect the data. The survey was chosen to obtain comparable data from both companies, while the participant observation allowed us to gain better insight and internal knowledge. The goal of the case studies was to assess the suitability and accessibility of the approach to SME software companies and, in particular, to verify the following research hypotheses:

1. It is possible to combine process definitions and task definitions;
2. Such combination can be done with commonly used software;
3. Such combination meets the needs of SMEs as a tool for process improvement.

The case studies involved two companies that fit the profile of potential users – the SME software companies. The first one appears as Company A, the second one – Company B. One of the authors had an opportunity to conduct a participant observation in both, making familiar with their way of working and the problems. This helped to propose the sample process model content and better present how the solution works and what could be the key benefits. It should be noted that both companies had some common needs addressed by solution. Both companies were interested in standardizing the implementation of projects, increasing the quality of processes and products, and knowledge management.

The Company A is a typical representative of SME – with low budget, tight deadlines and short term strategy constraints. Its strategy can be characterized by high sensitivity of the market situation and the need to acquire many new customers for relatively short-term contracts. It carries out orders for 11 years for external clients, often for the public sector, and has about 30 employees. It has close to functional structure and market organizational culture. Because of the relatively small size, it was easy to make changes in the company and communicate directly. The participant observation in company A took 17 months.

The Company B is a rapidly growing organization, which exists for 2 years, but derives from another organization with 12-years experience. It's clients are often from banking and insurance sector but also telecommunication and others. It has close to project structure and hierarchy culture. As for now, it has about 50 employees. It runs longer contracts with smaller financial constraints. Far-reaching strategy is being determined and the company is establishing position in the market and its customer segment. This is watershed moment, the last and best opportunity to streamline and standardize processes. The participant observation in company B covered 7 months.

In both companies, the tool support for automatic creation of tasks in a TM tool based on the process models made in an SPM tool was designed and implemented to verify the hypotheses 1 and 2. The sample content of the process model was developed and converted to web pages. The tools and models were presented to employees and feedback was collected in a survey to verify the hypothesis 3. The survey included the following questions starting with "Can this solution…":

- Q1: facilitate project realization in a more systematic and repeatable way?
- Q2: have positive impact on process and product quality?
- Q3: provide employees with access to company's knowledge, information and templates needed in the project?
- Q4: have positive impact on your comfort with the work because of instructions, templates, clearly assigned responsibility, and the list of tasks in the project known in advance?
- Q5: be applied in practice in the company's projects?

The possible responses were: definitely yes, rather yes, rather no, definitely no. Answers were selected so that respondents should decide about conviction they have (yes or no) and its strength (definitely or rather). We assume that to verify the hypothesis 3 positively we should obtain at least 60% of the answers "rather yes" and "definitely yes" for all questions.

4.1 Company A Case Study

EPF Composer was selected as an SPM tool for company A. The process model was limited to technical documentation department and covered all stages of project realization. The descriptions of model elements and the artifacts were based on those currently used. Two sample process models for two project types (a tender and an ordering) were build. The former consisted of 4 disciplines, 6 tasks, 2 roles, and 3 artifacts, while the latter comprised 3 disciplines, 3 tasks, 2 roles, and 3 artifacts. EPF Composer screenshot with some elements of the process model is shown in Fig. 2.

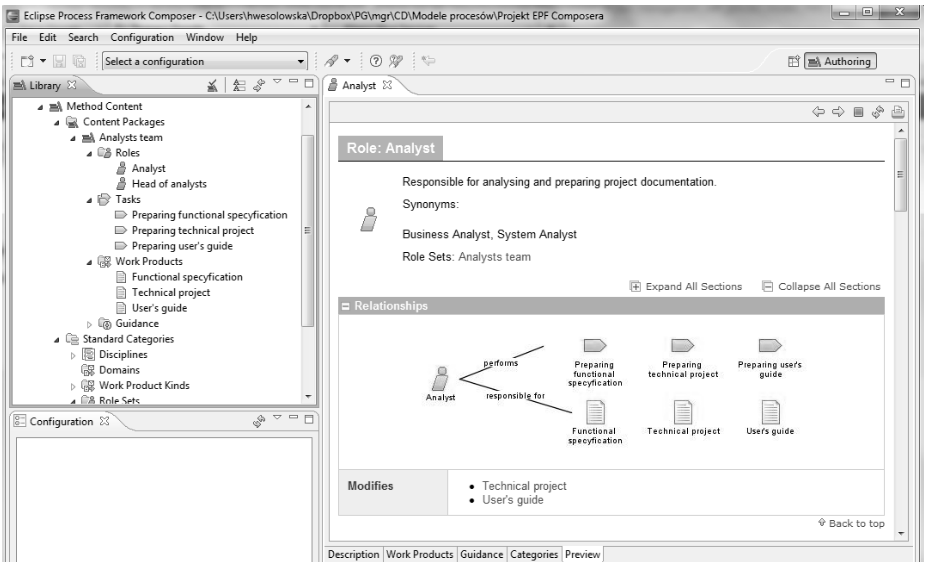


Fig. 2. EPF Composer with part of the process model in company A

In company A, each employee used daily a self-made company's TM tool, coded here as the Projects-Tool. This tool was adequately modified to support the translation of the EPF Composer's process models into its data model. Links to artifact templates and pages with process model elements generated from EPF Composer were included in the description of tasks. The implementation of the extensions to the Projects-Tool is shown in Table 2 (compare to Table 1).

Table 2. Implementation of model to task translation in TM tool of company A

TM tool element	Projects-Tool database table
Role	TRoles (added)
User	TUsers (modified)
Subtask	TTaskDetails (modified)
Task	TTasks (modified)
Template	Link to artifact template in Description field of TTaskDetails
Task description	Link to intranet webpage in Description field of TTaskDetails

Extension to the TM tool took about 80 working hours. This included learning both tools, data model analysis, design and programming the integration code. Process modeling took about 16 working hours. Most of this time was used to prepare detailed description of elements as well as artifact templates.

The anonymous survey involved 4 employees: an analyst, a technical documentation specialist, a tester and a project manager. This group was representative to the company's structure and the processes covered by our SPI initiative which was introduced bottom-up from the employees' level. In total, the survey achieved 100% positive answers (50% "Definitely yes" and 50% "Rather yes"). For Q1, Q3 and Q4, 3 respondents answered "Definitely yes" and 1 "Rather yes". For Q2, 1 person answered "Definitely yes" and 3 "Rather yes". All respondents believed that our solution could be ("Rather yes" answers) applied in practice in the company (Q5).

Participant observation revealed strong positive attitude of employees to this solution which we attribute to lack of similar SPI initiatives. Our approach was appreciated as a possible way to introduce some order into daily work and achieve better quality of project realization. So far, the solution was not used in any projects because of lack of strong support from the board. This can be due to market centered organizational culture, where improvement of internal processes seems to have lower priority.

4.2 Company B Case Study

The company B is interested in improving the process and hired a consultant, whose task was to develop a standard of projects realization. The consultant developed the process model and its content with the participation of one of the authors and taking account of guidance of project managers and the board. Similarly to company A, EPF Composer was selected as an SPM tool. The process model covered all phases of project realization (offer, analysis, project, realization, deployment) and included guidance for roles, tasks, and artifacts (the latter only for the analysis phase). The model consisted of 3 disciplines, 111 tasks, 12 roles, and 20 artifacts.

In company B, each employee uses every day a popular TM tool – JIRA [21]. The translation of EPF Composer's process models to the JIRA data model was achieved with a plug-in to JIRA. The roles, users, tasks, subtasks and task descriptions of the TM tool data model (see Table 1) are directly implemented by JIRA. The templates were mapped onto JIRA attachments. The process content stored as a website exported from EPF Composer was linked to from the task descriptions. Integration and adaptation of the tools took about 80 working hours, while process modeling took about 24 working hours. Model elements were described briefly and used already developed artifact templates.

The anonymous survey involved 4 employees - 1 member of the board and 3 project managers. This was the first target group of our SPI initiative, which was introduced top-down from the board and managerial level of the company. In total, the survey achieved 65% positive answers (5% "Definitely yes" and 60% "Rather yes") and 35% negative answers (10% "Definitely no" and 25% "Rather no"). In detail, questions Q1, Q3, Q4 and Q5 received 75% positive answers, while question Q2 received 25% positive answers. The latter is attributed to small testing sample and the answers of one respondent being significantly below the average (only negative answers). Due to survey anonymity we cannot assign this to any particular person or role.

In company B there was a positive attitude to our solution, but not as strong as in company A. Participating observer supposes that the company is open to new solutions because it is young and still developing. Nonetheless there were already some failed initiatives to standardize the processes, which could result in slightly less enthusiastic reception of our approach. The quality of project management in B contributed to more successful projects than in A. Company B also exhibited greater confidence in its own competencies. Our approach is decided to be used in everyday practice. In this case, it received strong support from the board. Company B has good position on the market and can afford investing in SPI.

4.3 Assessment of Results

In both case studies presented above it was possible to integrate process definitions and task definitions. Processes were defined in terms of models and content, which were then used to define tasks of particular projects following the models and using the content. This positively verifies the hypothesis 1 of our research.

We used EPF Composer as a process modeling tool in both case studies. It was selected owing to its popularity, free license, support for SPEM metamodel and website based process guides. In task management, we integrated process models with the proprietary TM tool in company A and a common marketed JIRA tool in company B. Both tools were adapted with little effort. In our opinion, the hypothesis 2 of our research can also be positively verified.

The summarized results of the survey from both case studies are shown in Fig. 3.

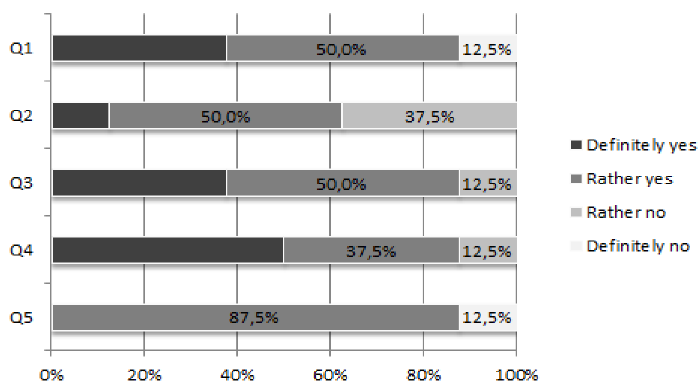


Fig. 3. Summarized survey results in companies A and B

In total, four questions received 87.5% ‘yes’ answers and one question received 62.5% ‘yes’ answers. It allows to verify positively the hypothesis 3 at this stage of the research. In detail, our method has the potential to facilitate project realization in more systematic and repeatable way (Q1). Also support for knowledge management

(Q3) was rated highly. Possible increase on the comfort of work was perceived even better (Q4). What is the most important, the survey shown that our solution could be applied in practice to support the companies' projects (Q5). Lower positive impact of our solution was perceived on process and product quality (Q2), particularly in company B, but the repeatable processes (Q1) are good starting point (first and the most important step) for improvement and achievement of better quality.

5 Conclusions

In this paper, we presented a method of improvement of project management which employs process models to enact project tasks with more guidance and repeatability. The method was applied in two case studies. Because results of process improvement require long time, resources and iterations, current research was limited to sample process modeling, tool integration and a survey on approach assessment complemented with participant observation. The case studies allowed to verify positively all the research hypotheses. In particular, we combined process modeling and task management in two different environment using commonly available tools. The results show that our solution can help small and medium companies in implementing the projects in a systematic and repeatable way, managing the company knowledge and increasing the employees work comfort. The applicability to practice was also appreciated. Based on the above, we can claim that our solution is a valuable contribution to address the research question. A more robust conclusion might be drawn from a larger sample or some more case studies in comparable organizations.

The main benefit from our solution is the systematization of processes – they have to be modeled, this means – presented in a certain form. This affects the consolidation of knowledge and enables access to it. The whole organization can start to work in the same, repeatable way. Models can be improved from version to version which requires only updating the process model in a PM Tool. Our approach contributes to facilitation of using process models in practice of task management. Process models are enacted in a task management tool already used in an organization. This can reduce the resistance of people as process models turn out to be feasible and useful. Employees are assigned tasks in the TM tool just like before and get specific support during their execution immediately. A model will ensure the consistency of all tasks and enable the project (and the whole organization) to move in the right direction. Our solution can be progressively implemented in selected parts of organization. It is also characterized by low cost and ease of implementation.

Despite many opportunities offered by defined processes and support for their enactment, the success of SPI with our approach still depends on many factors: scope of implementation in a company (insufficient training, missed opportunities), selection of roles, tasks, artifacts (inadequacy to company needs), model content (little valuable roles, tasks, artifacts descriptions), model improvement. The case studies carried out so far provided opinions on the improvement potential of our approach. Future work is planned to enact actual projects based on defined process models and assess the benefits to the process improvement. The observation of actual SPI results to task management with our approach should take into account long time to assess changes (more difficult if the company has previously worked differently for each project), dynamic SMEs changes, strategy, strong dependence on market, organizational structure, employee's roles, and competence needs.

Detailed specification of our approach and the complete description of company A case study is presented in [34].

References

1. Basri, S., O'Connor, R.V.: Understanding the Perception of Very Small Software Companies towards the Adoption of Process Standards. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 153–164. Springer, Heidelberg (2010)
2. Pino, F.J., Pardo, C., Garcia, F., Piattini, M.: Assessment methodology for software process improvement in small organizations. Information and Software Technology 52(10), 1044–1061 (2010)
3. Pino, F.J., Garcia, F., Piattini, M.: Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review. Software Quality Journal 16(2), 237–261 (2008)
4. Pino, F.J., García, F., Piattini, M.: An Integrated Framework to Guide Software Process Improvement in Small Organizations. In: O'Connor, R.V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 213–224. Springer, Heidelberg (2009)
5. Mishra, D., Mishra, A.: Software Process Improvement Methodologies for Small and Medium Enterprises. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 273–288. Springer, Heidelberg (2008)
6. Von Wangenheim, C.G., Weber, S., Hauck, J.C.R., Trentin, G.: Experiences on Establishing Software Processes in Small Companies. Information and Software Technology 48(9), 890–900 (2006)
7. Software Engineering Institute, International Process Research Consortium, CMU/SEI-2006-SR-001 (2006)
8. ISO/IEC JTC 1/SC7 Working Group 24, `http://profs.etsmtl.ca/claporte/English/VSE/`
9. CMMI Product Team: CMMI for Development, Version 1.3, Software Engineering Institute, CMU/SEI-2010-TR-033 (2010)
10. ISO, ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes (2008)
11. ISO, ISO/IEC 15504-4:2004: Information technology - Process assessment - Part 4: Guidance on use for process improvement and process capability determination (2004)
12. European Commission: The New SME Definition, User guide and model declaration (2005)
13. ISO/IEC 29110-4-1:2011 Software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 4-1: Profile specifications: Generic profile group (2011)
14. Richardson, I., Von Wangenheim, C.G.: Why are Small Software Organizations Different? IEEE Software 24(1), 18–22 (2007)
15. Clarke, P., O'Connor, R.V.: The Meaning of Success for Software SMEs: An Holistic Scorecard Based Approach. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 72–83. Springer, Heidelberg (2011)

16. Johnson, D.L., Brodman, J.G.: Tailoring the CMM for small businesses, small organizations, and small projects. In: El Emam, K., Madhavji, N.H. (eds.) Elements of Software Process Assessment and Improvement, pp. 239–259. IEEE CS Press, New York (1999)
17. Kautz, K.: Software Process Improvement in Very Small Enterprises: Does it Pay Off? Software Process – Improvement and Practice 4, 209–226 (1998)
18. Dyba, T.: An empirical investigation of the key factors for success in software process improvement. IEEE Transactions on Software Engineering 31(5), 410–424 (2005)
19. Sulayman, M., Mendes, E.: An extended systematic review of software process Improvement in small and medium web companies. IET Seminar Digests (1), 134–143 (2011), doi:10.1049/ic.2011.0017
20. IBM, Rational software, `http://www.ibm.com/software/rational/`
21. Atlassian, JIRA - Track bugs, tasks, and projects for software development, `http://www.atlassian.com/software/jira/overview`
22. Manifesto for Agile Software Development, `http://agilemanifesto.org`
23. ISO, ISO 9001:2008 Quality management systems — Requirements (2008)
24. Dingsoyr, T., Moe, N.B.: The Impact of Employee Participation on the Use of an Electronic Process Guide: A Longitudinal Case Study. IEEE Transactions on Software Engineering 34(2), 212–225 (2008)
25. Savolainen, P., Sihvonen, H.-M., Ahonen, J.J.: SPI with Lightweight Software Process Modeling in a Small Software Company. In: Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.) EuroSPI 2007. LNCS, vol. 4764, pp. 71–81. Springer, Heidelberg (2007)
26. Friedrich, J., Bergner, K.: Formally Founded, Plan-based Enactment of Software Development Processes. In: Raffo, D., et al. (eds.) Proceedings of the 2011 International Conference on Software and Systems Process, ICSSP 2011, Waikiki, Honolulu, HI, USA. ACM, New York (2011)
27. Yaeli, A., Klinger, T.: Enacting Responsibility Assignment in Software Development Environments. In: Dubinsky, Y. (ed.) Proceedings of the 1st International Workshop on Software Development Governance, SDG 2008, Leipzig, Germany. ACM, New York (2008)
28. Gonzalez-Perez, C., Henderson-Sellers, B.: A work product pool approach to methodology specification and enactment. The Journal of Systems and Software 81(8), 1288–1305 (2008)
29. Valiente, M.-C., Garcia-Barriocanal, E., Sicilia, M.-A.: Applying Ontology-Based Models for Supporting Integrated Software Development and IT Service Management Processes. IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications and Reviews 42(1), 61–74 (2012)
30. Hug, C., Front, A., Rieu, D.: Henderson-Sellers. B.: A method to build information systems engineering process metamodels. The Journal of Systems and Software 82(10), 1730–1742 (2009)
31. Object Management Group: Software & Systems Process Engineering Meta-Model Specification, Version 2.0 (2008), `http://www.omg.org/spec/SPEM/2.0/PDF`
32. IBM, Rational Method Composer, `http://www-01.ibm.com/software/awdtools/rmc/`
33. The Eclipse Foundation: Eclipse Process Framework Project (EPF), `http://www.eclipse.org/epf/`
34. Wesołowska, H.: Tool and procedural support for software development processes of SMEs (in Polish), MSc thesis, supervised by J. Miler, Department of Software Engineering, Gdansk University of Technology, Gdansk (2011)

Towards a Maturity Model for IT Service Management Applied to Small and Medium Enterprises

Renato Ferraz Machado, Sheila Reinehr, and Andreia Malucelli

Graduate Program in Computer Science (PPGIa) of
Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, Brazil
rfmachado@gmail.com, sheila.reinehr@pucpr.br,
malu@ppgia.pucpr.br

Abstract. Despite the discussions about Information Technology (IT) management, models are constantly on the agenda. A lack of maturity models that meet the needs of IT management service providers which conform to the reality of small and medium-sized companies can be observed. This paper presents a proposal for a maturity model for IT service management, called MM-GSTI. The proposed model is compliant to ISO / IEC 20000 and models CMMI for services (CMMI-SVC) and MPS.BR, and uses practices described in ITIL. Its goal is to help service providers in the implementation of improvements for the management of IT services.

Keywords: ITSM, Maturity Model, ISO 20000, MPS.BR, ITIL.

1 Introduction

The importance of internal services and their impact on the quality of manufactured products, according to [16], was the principle of the Total Quality Management (TQM) approach, developed in the '80s by Deming [6]. "Now there also appears to be a common acceptance that the internal service quality is an influence and a key quality collaborator of external services" [16]. Examples of this acceptance are the initiatives of development and use of models and international standards with the intent of setting a standard for service management processes, mainly for the services related to Information Technology (IT) [2],[3], [4] and [7].

The service industry is a significant driving force in the growth of the world economy. The development and the improvement of service practices are the keys for greater performance, an increase in client satisfaction, and profitability within the sector [24].

According to [8], Brazilian software companies that provide IT services had generated 25.8 billion dollars of operational net income; this includes 50 thousand companies, employing more than 367 thousand people, and paying more than 6.3 billion dollars in salaries. In 2010, the Brazilian IT service market achieved almost 19 billion dollars in net income [1]. A study conducted by SOFTEX [28] revealed that

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 157–168, 2012.

there are more than 50 thousand small and medium-sized software businesses and others working in the IT service sector in Brazil. This number represents about 98% of the total of 52 thousand companies in the sector. Another statistics shown in the study revealed that more than 20 thousand of these companies are related to IT services.

Models and standards have been developed for the organization, implementation, and evaluation of the processes involving ITSM [17]. Publications such as ITIL (Information Technology Infrastructure Library) [20] and COBIT (Control Objectives for Information and related Technology) [14], have grown around the world in terms of acceptance and usage. As reported in [33], European companies already have a broad acceptance of the practices of ITSM and that an estimate made in consultation with 167 directors of IT departments within American organizations point out that 90% of these organizations have one or more ITSM processes already implemented. The launching of the CMMI model for services - CMMI-SVC [24] and ISO/IEC 20000-2 [12] standard, reinforces this tendency and the concerns in this area.

ITIL is a framework of structured best practices for IT service management. ITIL is a de facto standard that has been introduced and distributed by the British Government Ministry Of Commerce (OGC). It contemplates practices of service management for the several aspects of IT within organizations [20]. “On its third edition, ITIL being the most accepted approach for IT services” [20]. It focuses on the: Strategy, Design, Transition, Operation and Continual Improvement dimensions.

The ISO/IEC 20000 standard consists of five parts under a general title: Information Technology - Service Management. The ISO/IEC 20000-1 specifies the requirements to plan, establish, implement, operate, monitor, revise, maintain, and improve the ITSM for the service provider. The ISO/IEC 20000-2 represents a consensus within the sector concerning quality standards in processes of ITSM and describes the best practices for these processes [12]. The ISO/IEC TR 20000-3 provides orientation, explanations, and recommendations for the definition of scope, applicability, and demonstration of the conformity to ISO/IEC TR 20000-1 by using practical examples. The ISO/IEC TR 20000-4 goal is to provide a guideline to the development of an assessment model according to the standard ISO/IEC 15504 [13]. The ISO/IEC TR 20000-5 presents an example of an implementation plan.

The CMMI for Services - CMMI-SVC [24], launched in 2009 and updated in 2010, is the most recent model of the SEI series. This model concerns to the application of best practices for process improvement in companies that provide IT services. The best practices of this model focus on the activities to provide quality services for the client and end-users. The CMMI-SVC integrates a body of knowledge that is essential for the service provider [24].

Other studies like the research of [22] defines a maturity model as the implementation of practices of ITIL specifically to define a scale of implementation. The model proposed by the authors provides 5 maturity levels, such as CMMI-SVC, in both the staged and continuous versions.

The concepts of ISO/15504 [13] were used in the project named Tudor´s ITSM Process Assessment - TIPA [30] of the Henri Tudor Public Research Centre. Two models were developed for the evaluation of ITSM processes. The first is a reference model of the processes based on the practices of ITIL, named Process Reference Model (PRM) and the other is a model for the processes evaluation based on ISO/IEC 15504, named Process Assessment Model (PAM).

Some Brazilian initiatives in the development of ITSM models are restricted to an academic scope. An example is the doctoral thesis of [5] where the author proposed a method for the creation of the model of IT service management based on existing models. Another research paper related to ITSM can be seen at [25], where the author had made a study of a framework for the optimum ITSM practices, ITIL and its alignment with business. The creation of a framework divided in layers, based on ITIL, which leads to a process to assist in ITSM, is the proposal of a master's dissertation of [18]. Another research related to ITSM is a master's dissertation of [26] that makes an analysis of the implications of the premising of the eSourcing Capability Model (eSCM) [15] for its suitability in the context of Brazilian institutions.

The challenge, according to [17] and supported by [3],[4] and [29], is bringing together the best practices of these models and standards in a unique model that can be implemented in the shortest possible time and in small and medium-sized businesses.

In this context, we found an opportunity to propose a maturity model for ITSM, mainly focused in small and medium-sized businesses, taking into account that [17]: (i) The market is composed basically of small and medium-sized software businesses and IT services; (ii) The need of a model that may serve as reference for services process improvement based on the concept of maturity levels, allowing the gradual adoption of processes in accordance with the availability of the organization; and, (iii) The difficulty in adopting a very large number of practices simultaneously by a small and medium-sized company.

This paper describes the proposal of a maturity model based on the research of [17], called Maturity Model for IT Services Management (MM-GSTI). It aims at guiding process improvement programs for IT services companies, especially small and medium-sized businesses. The proposed model is based on Reference Model of Brazilian program for software process improvement - MPS.BR [27].

The other sections of this paper are organized as follows: Section 2 presents the MPS.BR concept. In Section 3 the research methodology is shown. In section 4 are presented and discussed the results of the research. Section 5 describes the research with ITSM experts and section 6 concludes this article.

2 The Brazilian Program for Software Process Improvement - MPS.BR

The Brazilian program for software process improvement (MPS.PR) is an initiative coordinated by the Association for Promoting the Excellence of Brazilian Software (SOFTEX). The program, which has the support of official government entities, universities and organizations, led to the development of a process improvement

Reference Model (MR) for software process improvement(MR-MPS) [27]. Created for the Brazilian reality, the novelty of the project was the adopted strategy for its implementation that sought to increase the maturity of the processes of the Brazilian software companies, mainly the micro, small, and medium-sized companies at an accessible price, according to [32]. The MR-MPS (MPS Reference Model) of the MPS.BR defines seven maturity levels: A (In Optimization), B (Managed Quantitatively), C (Defined), D (Broadly Defined), E (Partially Defined), F (Managed), and G (Partially Managed). The scale of maturity begins at the level G and progresses up to level A.

3 Research Methodology

The research for the development of MM-GTSI combines qualitative and quantitative procedures and in its execution, the following phases were used: i) Comprehension of the ITSM scenario, specifically regarding to the models and standards used (bibliography research); ii) A conception of the proposed model including: definition of a structure for the elaboration of the model containing the identification of the maturity levels and the attributes of capability (selected existing models and identification and selection of the ITSM processes that are applicable); iii) Field research to identify the most used sequence of implementation in the organizations that uses ITSM processes; iv) The conception of the proposed model including the classification of the proposed processes at certain maturity level; and v) Survey with experts for the evaluation of the proposed model.

For the definition of processes in the maturity levels of MM-GSTI, research was done through a semi-structured interview with six small and medium-sized Brazilian companies, that were chosen using the following criteria: i) the organization should be a micro, small, or medium-sized IT company (considering the criteria of the allocated people used by [9]), ie up to 250 employees; ii) the organization should provide some type of IT service; iii) the percentage of their participation of services should be more than 50% of the total income of the company; iv) the organization should operate within the Brazilian market; and v) the organization should have implemented, at least partially, some process related to ITSM.

The semi-structured interview investigated how the improvement programs in ITSM were implemented and the sequence in which the processes were implemented. At the end of the interviews the interviewees were asked to set up a desirable sequence for the implementation of ITSM processes suggested by ISO-IEC 20000-1 [11].

To evaluate the proposed model, a survey was conducted with ITSM experts from several institutions, with broad experience in implementation, training or evaluation of ITIL, ISO / IEC 20000 and CMMI-SVC. Next, a questionnaire was sent to evaluate the processes in the maturity levels. Thirteen questionnaires were sent by electronic mail, six of them were returned completed.

4 The Proposed Model Conception

The general structure of MM-GSTI was based on the structure of the Reference Model of MPS.BR (MR-MPS) [27] since it was a model already focused on micro, small, and medium-sized companies in the software development area [32].

According to [27], the division into 7 stages aims at enabling an easier implementation and a more proper assessment of small and medium-sized companies. The chance to perform assessments, considering more levels, also allows a greater visibility of the results of process improvement in a shorter time frame, according to [27].

The same concepts of process capability defined in MR-MPS [27] and ISO/IEC 15504 [13] were also used. For each maturity level, there is a set of Process Attributes (AP) and their results which define the degree of institutionalization of each process.

Both, the ISO/IEC 20000-1 [11] standard and CMMI-SVC v1.3 [24] model were took into account to define the new processes related to ITSM which were included in the proposed model. As many ITIL process have been incorporated in ISO/IEC 20000, we did not use specific process of ITIL for the MM-GSTI conception. For the verification of the adherence of the processes, two mappings of the processes, practices, and attributes of proposed processes with the processes and practices of CMMI-SVC and ISO/IEC TR 20000-4 were developed.

To keep the compliance with the CMMI-SVC model, some processes of the MR-MPS were excluded and incorporated into a single process called Service System Development. The Reuse Management process and the Development for Reuse process were excluded from level E of the MR-MPS. Once these two processes are mainly related to software development, they were not considered to take part into the MM-GSTI.

To define the new processes, the following factors were considered: i) The processes and the results of the processes must be compliant to the processes and practices of ISO/IEC TR 20000-4 [10]; and ii) The processes and the results of the processes must be compliant to the processes and the specific practices of CMMI for Services (CMMI-SVC), version 1.3 [24].

We keep the compliance with CMMI-SVC in the same way that MR-MPS is already in compliance with CMMI-DEV.

The new processes related to ITSM and defined for the MM-GSTI followed the processes that were already defined in the ISO/IEC/20000-1 [11]. The objectives and results of the processes were based on the ISO/IEC TR 20000-4 [10]. The only exceptions were the processes from the Service System Development and the Service Delivery that were added to the MM-GSTI to ensure compliance to the processes of the CMMI-SVC [24].

The results of the processes and practices mapping generated the new processes of MM-GSTI which are: Capability and Availability of Services Management; Information Security Management; Incident Management; Release Management; Service Level Agreement Management; Problem Management; Budget and Accounting for IT Services; and Service Report.

Some updates were made in the already existing processes of MR-MPS [27] in order to keep the compliance with the processes of the ISO/IEC TR 20000-4 [10] standard and the CMMI-SVC model [24], in the ITSM context.

To define the ISTM processes in the maturity levels of MM-GSTI, the following factors were taken into account: i) The results of the field research with Brazilian IT companies; ii) The results of the research of Magalhães and Brito [19]; Cater-Steel and Pollard [4]; and Cater-Steel, Toleman and Tan [3], which show experiments of the sequences of implementation of ITIL practices; iii) The research of Pereira and Silva [22], and iv) The distribution of the processes on the levels of CMMI-SVC [24].

4.1 Results of the Field Research

For field research, with the selected companies, we used the concept of Points of Analysis (PAN), used by [23]. The points of analysis were used to determine how the data obtained in this research might help to define the process maturity levels in the MM-GSTI.

During the field research, other themes related to ITSM were approached, however, for the effectiveness of the definition of the processes in the maturity levels, we used two points of analysis: **PAN 1** - The process implementation sequence followed by the organization and what process implementation sequence would be followed if the implementation was started again; and **PAN 2** - Suggested sequence, considering the ITSM processes of ISO/IEC/20000-1 [11].

Therefore, a tabulation of the consolidated data of the answers from all the companies that had been interviewed was generated, considering each process and its possible sequence of implementation. For each process, a sequence number was appointed, being 1st for the first process to be implemented, 2nd for the second and so on.

Figure 1 summarizes the amount of votes for each process and position in the implementation sequence. For example, the Incident Management process received one vote for the 1st position, three votes in the 2nd position and two votes in the 3rd position, and so on for the other processes.

All the values identified in PAN 1 and PAN 2 (Fig. 1) composed a consolidated sum that defined a sequence of the implementation of each process. This helped to indicate the maturity level of MM-GSTI was more suitable.

Process	Number of votes by position – PAN1 + PAN2									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Incident Management	3	5	4	-	-	-	-	-	-	-
Problem Management	1	6	3	2	-	-	-	-	-	-
Service Level Agreement Management	6	-	-	3	2	-	-	-	-	-
Change Management	-	-	2	2	-	-	2	-	-	-
Budget and Accounting for IT Services	2	-	1	3	1	-	2	-	-	-
Continuity and Availability Management	1	-	1	1	1	3	-	-	-	-
Capacity Management	3	2	-	-	2	-	-	-	-	-
Information Security Management	-	-	-	-	2	2	-	-	1	1
Release Management	-	1	1	-	-	1	-	3	-	-
Services Reporting	-	1	-	-	1	1	-	1	2	-

Fig. 1. Sum of votes of PAN 1 and PAN 2

4.2 Research Results from Other Authors

In order to validate and consolidate an appropriate process implementation sequence, research from others authors were taken into account. These works helped to compose the final ranking for the definition of the implementation sequence of MM-GSTI processes.

In first place, the research of Magalhães and Brito [19] was considered. The authors suggested an implementation sequence of the ITIL model based on a practical experiment. Afterwards, the research conducted by Catter-Steel and Pollard [4] was considered. It also shows a sequence of implementation of the ITIL model in Australian companies. The research done by Cater-Steel, Toleman and Tan [3] also presented a sequence of implementation of the ITIL model in American and Australian companies. Finally, the research of Pereira and Silva [22] proposed a maturity model with 5 levels contemplating the practices described in ITIL v3 [21].

The same procedure used to analyze PAN 1 and PAN 2 was used in the process implementation sequence analysis. The amount of these votes composed the general consolidated counting of these three factors, as shown in Fig. 2.

The CMMI-SVC [24], in its staged representation, were also used for the definition of the MM-GSTI processes, because it is a maturity model focused on services and which is already structure in maturity levels. Some processes of CMMI-SVC have different names from those of MM-GSTI. Due to this, it was necessary to reach a consensus before arriving at a final result. The result of CMMI-SVC analysis generated a sum of votes by position to this model. To map CMMI-SVC, it was considered that the maturity level 2 process areas were the first to be implemented (position one) and the maturity level 3 process areas were the second ones to be implemented (second position).

4.3 Result Consolidation and Discussion

For the definition of the processes of the MM-GSTI model in terms of maturity levels, it was generated a consolidation of the results, as seen in the previous items. With this final tabulation, it was possible to generate Fig. 2, which shows the sum of the 3 factors, emphasizing the most voted processes in each position.

According to the results shown in Fig. 2, it was generated a distribution of the processes by maturity levels, represented in Fig. 3, that identifies the most voted processes in each position. The Incident Management process and Service Level Agreement Management process had the highest number of votes in the first position. The Problem Management process had the highest number of votes in the second position and so on for other processes.

The MM-GSTI follows the same structural definition of levels proposed by the MR-MPS [27] model (levels scaling up from G to C, remaining on levels B and A only with the attributes of the processes and their results for the definition of measurement, control, improvements, and continual optimization of the process).

As can be seen in Fig. 3, the 10 new processes of ITSM can be distributed in the 5 first maturity levels of MM-GSTI (G until C). For the G level, the Incident Management and the Service Level Agreement Management processes were defined, once these processes had the highest amount of votes. Following the same criteria, the Problem Management process was placed in level F and so on for the other processes until level C.

Process	Number of votes by position									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Incident Management	9	9	6	-	-	-	-	-	-	-
Problem Management	3	14	5	2	-	-	-	-	-	-
Service Level Agreement Management	9	2	1	3	2	-	-	-	-	-
Change Management	7	6	2	2	-	-	3	-	-	-
Budget and Accounting for IT Services	4	1	1	3	1	-	1	1	-	-
Continuity and Avalilability Management	1	3	1	1	1	3	-	-	-	-
Capacity Management	3	4	-	-	2	-	-	-	-	-
Information Security Management	-	-	1	-	2	2	-	-	1	1
Release Management	-	4	1	-	-	1	1	2	-	-
Service Report	-	2	1	-	1	1	-	1	2	-

Fig. 2. Amount of votes for the consolidated position of the three factors for defining the processes in maturity levels in MM-GSTI

Process	Position	Maturity level
Incident Management	1st	G
Service Level Agreement Management		
Problem Management	2nd	F
Change Management	3rd	E
Budget and Accounting for IT Services	4th	D
Capacity Management	5th	C
Information Security Management		
Continuity and Availability Management	6th	
Release Management	7th	
Service Report	8th	

Fig. 3. Processes most voted in each position in the consolidated sum of the three factors for defining the processes at maturity levels in MM-GSTI

With the definition of new processes of ITSM in each maturity level, it was possible to define the MM-GSTI with all processes, as shown in Table 1.

The Service Delivery and Service System Development processes were included in G and D levels, respectively, to be compliant to MR-MPS and CMMI-SVC. The others processes, that were not directly linked with ITSM, that appear in Table 1 were kept or adopted from MR-MPS. The process attributes (AP) were also maintained from MR-MPS. They define the attributes for the capability of each process in their related maturity levels.

The proposed ITSM maturity model, with seven maturity levels, becomes a promising tool for implementing improvement programs on IT service providers.

Some initiatives in this area were responsible for the proposal of ITSM models, although none of them presented a maturity model structured in maturity and capability levels, with defined processes for each of them, as presented in this work.

The main difference between the CMMI-SVC model [24] and the MM-GSTI model is that the later provides a larger amount of maturity levels, which can facilitate its implementation, mainly in small and medium-sized companies. The basic difference of MM-GSTI in relation to the other models is a major scaling of the maturity levels that could drive the proposed model to the same way of MR-MPS that has a good acceptance in the small and medium-sized software companies' scenario, according to [31]. The division into seven stages is more suitable to small and medium-sized companies [27].

The proposed model is different from the practices related in ITIL [21] and ISO/IEC TR 20000-4 standard [10], because it points out a processes implementation sequence, based on maturity levels and a set of processes on each level.

Table 1. The MM-GSTI with all process at maturity levels and the process attributes

Maturity Level	Process	Capacity Process Atributes (AP)
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2 , AP 5.1 e AP 5.2
B	Work Management (evolution)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Capacity Management	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Continuity and Avaialbity Management	
	Decision Management	
	Release Management	
	Information Security Management	
	Risks Management	
	Service Report	
D	Service System Development (Additional)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Budget and Accounting for IT Services	
E	Evaluation and Improvement of Organizational Process	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Organizational Process Definition	
	Change Management	
	Human Resources Management	
	Work Management (evolution)	
F	Aquisition	AP 1.1, AP 2.1 e AP 2.2
	Quality Assurance	
	Configuration Management	
	Project Portfolio Management	
	Problem Management	
	Measurement	
G	Service Delivery	AP 1.1 e AP 2.1
	Incident Management	
	Service Level Agreement Management	
	Requirements Management	
	Work Management	

5 Evaluation Using Experts Opinion

In order to evaluate the proposed distribution of the processes in such maturity levels, an expert opinion were used. To achieve this, the Likert scale was used to measure the level of agreement or disagreement for each statement. The following scale was used: "Strongly Agree" (2), "Agree" (1), "Neither Agree nor Disagree" (0), "Disagree" (-1) and "Strongly Disagree" (-2). For each option the scale sets a value that decreases from 2 to -2. These weights were used to calculate the weighted average of total responses for each of the questions answered by experts. To measure the degree of acceptance of the proposed model, five questions were presented to the experts to verify their acceptance in relation to the processes at each maturity level of MM-GSTI.

The first question concerns the acceptance of the processes defined in the first sequence of implementation, which is equivalent to level G of MM-GSTI and so on until level C. The overall result pointed to an average impact of 1.07, which, using the weights of the Likert scale leads us to a value near to 1. This means that the experts agree with the processes and the sequence presented in the proposed model.

6 Conclusion

This paper presented a maturity model for ITSM, the MM-GSTI model, which was evaluated by a set of experts in software process improvement and service models. The MM-GSTI model can become an appropriate path to service management improvement programs, especially for small and medium-sized companies. It has also provided the opportunity to enrich the debate on models and standards related to ITSM as well as those showing new implementation possibilities of ITSM practices described in ITIL and ISO/IEC 20000, in staggered maturity levels. Additionally, we demonstrated that the MR-MPS [27] could be adapted to other forms so that the model may be used as a basis for generating other maturity models.

An important contribution of this research is that the proposed model was chosen to be used as a basis for creation of the maturity model service-oriented of Brazilian Program for Software Process Improvement (MPS.BR). The trial phase started on January 2012.

Some limitations of this study should be highlighted. Even though, in an earlier phase, a scenario of several companies from different sizes and lines of business were used, this research is limited to the companies surveyed, though it comprises a small sample among the large amount of IT service providers established in Brazil. Since few reports of experiments in the ITIL model implementation are available, they may not be sufficient to represent the broad sample required for making decisions regarding the order of implementing the processes. As the amount of ISO/IEC TR 20000-4 [10] ITSM practices is so broad, there is a chance that these rules do not cover all aspects necessary to meet the reality of all companies that provide IT services. Finally, another limiting aspect is the debate about other processes not directly linked to ITSM that were kept back in the MM-GSTI model. Even though these processes have already been established and widely used in the MR-MPS [27], they may require some tailoring to better match the improvement programs in the ITSM context.

As a continuation of this work, in order to enhance and improve this research, we highlighted: the practical application of the proposed model as a pilot in small and medium-sized IT services providers to validate their applicability, as well the effort required to implement the proposed model for each maturity level (this is being carried out since January 2012); implementation of the proposed model for different types of IT services and different areas of customer service activities in order to evaluate their suitability for different types of businesses; creation of other complementary guides, such as implementation guide, evaluation guide, and acquisition guide, similar to those already in place for the MPS.BR model; further debate on the compatibilities and differences among the processes and practices from the CMMI-SVC, ISO/IEC TR 20000-4 and ITIL.

References

1. ABES - Brazilian Association of Software Companies. Brazilian Software Market: Scenarios and Trends 2011, São Paulo, 1st edn. (2011)
2. Ayat, M., Sharifi, M., Sahibudin, S., Ibrahim, S.: Adoption Factors and Implementation Steps of ITSM in the Target Organizations. In: Proc. of Third Asia International Conference on Modeling & Simulation. IEEE Computer Society (2009)
3. Cater-Steel, C., Toleman, M., Tan, W.: Transforming IT service Management – the ITIL Impact. In: 17th Australasian Conference on Information Systems, Adelaide, Australia, December 6-8 (2006)
4. Cater-Steel, C., Pollard, C.: Justifications, Strategies, and Critical Success Factors in Successful ITIL Implementations in U.S and Australian Companies: An Exploratory Study. Information Systems Management (26), 164–175 (March 2008)
5. Clementi, S.: The GSS-COBITIL model for managing support services of information technology. Doctoral Thesis. Escola Politécnica da Universidade de São Paulo, São Paulo (2007)
6. Deming, W.E.: Get out of the crisis: the final 14 lessons for quality control, p. 503. Futura, São Paulo (2003)
7. Hochstein, A., Tamm, G., Brenner, W.: Service-Oriented IT Management: Benefit, Cost and Success Factors. In: 15th European Conference on Information Systems, Regensburg, Germany, May 26-28 (2005)
8. IBGE - Brazilian Institute of Geography and Statistics. Annual Survey of Services – 2008, Rio de Janeiro, vol. 10, pp. 1–205 (2008)
9. IBGE - Brazilian Institute of Geography and Statistics. 2006 Business demography (2006), http://www.ibge.gov.br/home/estatistica/economia/demografiaempresa/2006/demoempresa2006.pdf (access in May 14, 2010)
10. ISO – International Standard Organization. ISO/IEC TR 20000-4: Information Technology – Service Management – Part 4: Process reference model, Switzerland, p. 30 (2010)
11. ISO – International Standard Organization. ISO/IEC 20000-1: Information Technology – Service Management. Part 1: Service management system requirements, Switzerland, p. 17 (2004)
12. ISO – International Standard Organization. ISO/IEC 20000-2: Information Technology – Service Management. Part 2, Code of practice, Switzerland, p. 34 (2005)
13. ISO – International Standard Organization. ISO/IEC 15504-1: Information Technology – Process Assessment – Part 1: Concepts and vocabulary, Switzerland, p. 19 (2004)
14. ITGI - IT Governance Institute. COBIT 4.1 – Control Objectives Management Guidelines Maturity Models. Rolling Meadows, IL, USA (2007)
15. ITSQC – Information Technology Service Qualification Center. The eSourcing Capability Model for Services Providers (CMU/ITSQC-06-006), Version 2.1., p. 104. Carnegie Mellon University, Pittsburg (2006)
16. Johnston, R.: Internal Service – barriers, flows and assessment. International Journal of Service Industry Management 19(2), 210–231 (2008)

17. Machado, R.F.: MM-GSTI: A proposal for a Maturity Model for IT Service Management with a focus on small and medium companies. Master Dissertation. Department of Informatics, Pontifical Catholic University of Paraná – PUC-PR, Curitiba, p. 257 (2011)
18. Martins, M.M.G.: IT Service Management: A Proposal for Integration of Process Improvement and Management Services. Dissertation. Department of Electrical Engineering at the University of Brasilia, Distrito Federal (2006)
19. Magalhães, L., Brito, W.: It service management in practice: an approach based on ITIL. Includes ISO/IEC 20000 and it flex. Novatec, São Paulo (2007)
20. OGC - Office of Government Commerce. Service Strategy Book, `http://www.best-management-practice.com/Publications-Library/IT-Service-Management-ITIL/ITIL-Version-3/Service-Strategy/?DI=582325` (access in January 16, 2010)
21. OGC - Office of Government Commerce. ITIL - Service Strategy. The Stationary Office, Belfast, United Kingdom (2007)
22. Pereira, R.F.S., Silva, M.M.: A Maturity Model for Implementing ITIL v3. In: Proc. of 6th World Congress on Services, pp. 399–406. IEEE Computer Society (July 2010)
23. Reinehr, S.: Systematic software reuse and software product lines in the financial sector: case studies in Brazil. Doctoral Thesis. Department of Production Engineering of the Polytechnic University of São Paulo, São Paulo (2008)
24. SEI - Software Engineering Institute. CMMI for Services (CMU/SEI-2010-TR-034), Version 1.3., p. 520. Software Engineering Institute, Pittsburg (2010)
25. Silva, R.F.: A convergent approach to the mapping of ITIL processes similar to the eTOM model. Dissertation. Department of Electrical Engineering at the University of Brasilia, Distrito Federal (2007)
26. Soares, H.F.: Importation of managerial technology for managing IT service delivery: a study on the applicability of the model to the eSourcing Capability Model Brazilian institutional context. Dissertation. Department of Management College of Economics, Administration and Accounting from the University of São Paulo, São Paulo (2008)
27. SOFTEX - Society for Promoting Brazilian Software Excellence (SOFTEX). MR-MPS.BR – General Guide – v1.2, `http://www.softex.br/mpsbr/_guias/default.asp` (access in September 13, 2009)
28. SOFTEX - Society for promoting Brazilian Software Excellence (SOFTEX). Software and IT services: The Brazilian industry in perspective - n.1. SOFTEX Observatory, Campinas (2009)
29. Tan, W., Caeter-Steel, A., Toleman, M., Seaniger, R.: Implementing centralized IT services Management: Drawing Lessons from the Public Sector. In: 18th Australasian Conference on Information Systems, Toowoomba, Australia, December 5-7 (2007)
30. TIPA - Tudor's ITSM Process Assessment. An ITIL Assessment, `http://www.tipaonline.org/tipa/about-tipa` (access in April 2, 2011)
31. Travassos, G.H., Kalinowski: IMPS 2009: characterization and variability in performance of organizations that adopted the MPS model. Softex, Campinas (2009)
32. Weber, K., Rocha, A.R., Nascimento, C.J.: Software Quality and Productivity, 4th edn. Makron Books, São Paulo (2001)
33. Winniford, M., Conger, L., Erickson-Harris, L.: Confusion in the Ranks: IT Service Management Practice and Terminology. Information Systems Management (26), 153–163 (March 2008)

Towards Configurable ISO/IEC 29110-Compliant Software Development Processes for Very Small Entities

Quentin Boucher[1], Gilles Perrouin[1],
Jean-Christophe Deprez[2], and Patrick Heymans[1,3]

[1] PReCISE Research Centre, University of Namur, Belgium
[2] Centre of Excellence in Information and Communication Technologies, Belgium
[3] INRIA Lille-Nord Europe, Université Lille 1 – LIFL – CNRS , France
{qbo,gpe,phe}@info.fundp.ac.be,
jean-christophe.deprez@cetic.be

Abstract. Using ISO/IEC 29110, very small entities (VSEs) can perform a stepwise increment of their software process by switching between the different ISO/IEC profiles. However, ISO/IEC 29110 provides no guidance on how to switch between profiles incrementally, other than resorting to costly software development process experts unaffordable for VSEs. To address this shortcoming, this paper shows how to model the variability of currently available ISO/IEC 29110 profiles in an integrated and configurable workflow with illustration on the Requirements Engineering (RE) activity. This workflow is linked to a questionnaire used to support automated process configuration. Thereby, the user can easily derive the ISO/IEC-compliant processes to switch between profiles incrementally. The feasibility of this approach is shown using open-source workflow management tools Synergia and YAWL.

Keywords: Workflow Management, Configuration, ISO/IEC 29110, Requirements Processes.

1 Introduction

From the first versions of Software CMM [1] and ISO/IEC 12207 [2], reference process models for software development have attracted a lot of interest over the past decade. However, they have failed to become accepted by Very Small Entities (VSEs), i.e., enterprises, organizational units or projects composed of 25 people or less [3]. The major criticism of the aforementioned standards relates to their excessive complexity, and thus their inapplicability to contexts where resources are extremely limited. The second recurring criticism is the technical jargon used in standards [4]. Nevertheless, VSEs remain interested by ISO certification [4]. To fill this gap, ISO/IEC recently published the ISO/IEC 29110 standard [3], which provides adapted development processes for VSEs.

Despite notable effort to make ISO/IEC 29110 more applicable to VSEs than its predecessors, considerable time and resources to understanding and applying the standard is still needed often requiring intervention from software process consultants. Moreover, if certification is the target —which seems to be the case for many VSEs [4]— records of systematic application of the standard must be kept and exhibited upon demand to certification authorities. The adoption barrier thus remains high.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 169–180, 2012.

To alleviate the difficulties of applying the ISO/IEC 29110 standard, the concept of *deployment package* has been introduced [5]. Deployment packages are additional documents, task cards [6], detailed process descriptions and templates [7]. Although they improve the understanding of the standard, they are of little help for the concrete realization of reference models, and do not guarantee compliance throughout the project lifecycle. Furthermore, reference models still need to be tailored to the specific operational needs of the adopting VSE.

In this paper, we propose to address the above problems by applying *configurable workflow*[1] [8] concepts and technology. First, we model the *Basic* and *Entry* profiles of the ISO/IEC 29110 *Software Requirements Analysis* sub-process as a single, integrated, workflow. This workflow exhibits the commonalities and differences between the two profiles and allows a fine-grained tailoring of the activities and artefacts. Tailoring is realized by simply asking the user to answer a set of questions [9]. Answers to these questions are used to automatically derive a correct and ISO/IEC-compliant workflow that can be integrated within widespread workflow engines. The approach is currently being implemented using the Synergia tool suite [10].

The paper is organized as follows. Section 2 presents the ISO/IEC 29110 standard. It also relates the configurable workflow approach to principles of method engineering and introduces our research methodology. Section 3 describes our application of the latter concepts and the implementation. In Section 4, our contribution is discussed. Finally, Section 5 wraps up the paper and presents some on-going and future developments.

2 Background and Related Works

2.1 ISO/IEC 29110

In 2011, ISO/IEC published a first version of a five-part standard, named "ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities" [3]. Parts 1 and 5 of ISO/IEC 29110 target a VSE audience. Part 1 presents an overview of the standard whereas Part 5 describes the major software development lifecycle processes for VSEs. The three other parts present mandatory information for all standards but do not necessarily need to be understood by VSEs. Part 2 introduces the general framework for developing current and future profiles; Part 3 discusses the assessment dimension; Part 4 establishes a correspondence between elements of ISO/IEC 29110 and their counterparts in ISO/IEC 12207.

From inception, the intent is to create multiple profiles to define a process improvement ladder made of four rungs. Based on its current process maturity, a VSE can then start at the appropriate rung (ranging from *Entry* to *Advanced*) by setting up the software process for that rung before climbing to the next rung. In the 2011 version of ISO/IEC 29110, only one profile was included in the standard: the *Basic* profile (rung 2). At the moment, an internal version of the *Entry* profile (rung 1) also exists and will be officially published in 2012.

The profiles defined in ISO/IEC 29110 use the following main concepts:

[1] We use the term "workflow" to denote an executable process model.

– A *software process* is composed of a set of objectives. An enterprise is said to have a software process in place if an auditor determines that the process' objectives are reached by the work performed by the enterprise.
– A *process task* describes the work that staff members with particular roles need to perform using a specified set of inputs to create the expected set of outputs.
– A *process activity* groups a set of related process tasks. A process activity only appears in one process and a task only appears in a single process activity.

The *Entry* and *Basic* profiles of ISO/IEC 29110 are composed of two processes: project management (PM) and software implementation (SI). The PM-*Entry* and SI-*Entry* are actually lighter than their counterparts in the *Basic* profile. Pragmatically, the *Entry* profile was created by editing out portions of the *Basic* profile. In some cases, tasks were removed altogether. In others, the work required to perform a task was simplified and eventually simplified tasks were merged.

2.2 Method Engineering

Method Engineering (ME) is the discipline of designing, constructing and adapting methods, techniques and tools for the development of information systems [11]. *Situational* method engineering refers to the customization of methods for the particular project and context at hand. Situational method engineering generally proceeds by combining method fragments reused from a common "methodbase" [12]. Sometimes only one fragment of the method is changed, or *incremental* ME. Naturally, this fragment is usually changed to improve the performance of the overall method by finetuning it to a specific situation. This makes the link between incremental ME and software process improvement obvious [13,14,15]. Another way to deal with flexibility is to introduce variation points in the method itself. This is the approach followed by ISO/IEC with their lifecycle profiles. Then, it is possible to perform *method configuration* [16] to tailor an existing base method. Thus, ME and method configuration form a conceptual framework to define and tailor software engineering processes such as defined by ISO/IEC.

Over the years, several Computer Assisted Method Engineering (CAME) tools, such as MetaEdit+ [17], have been developed to assist method engineers tuning up method to the needs of a project. However, as noted by Cervera et al. [18], besides MetaEdit+, very few tools reached industrial maturity. Furthermore, industry usually thinks that the costs of applying situational ME is perceived as being larger than using an off-the-shelf method [12].

If we focus on VSEs, the target of our approach, further issues arise. One is the cost of such CAME tools. VSEs generally have limited financial resources and prefer to invest in software development tools than in tools guiding their development process. As a result, certification and self-assessment of ISO/IEC 29110 compliant VSEs is challenging. Providing an affordable approach to software lifecycle management is currently the focus of the NAPLES project[2]. Additionally, VSEs hardly have ME skills internally: subtleties of ISO/IEC 29110 profiles' combinations may be overlooked (unnecessarily forbidding relevant method configuration) or, on the contrary, important constraints may be ignored allowing incorrect application of the standard.

[2] `http://www.cetic.be/NAPLES,1162`

Therefore, our research relies on open-source solutions and strives to provide a reusable method configuration approach based on the ISO/IEC 29110 to reduce the ME effort to its minimum. Our approach relies on workflow configuration which is detailed in the next paragraphs.

2.3 Configurable Workflows

Workflow Management Systems (WFMS) help realizing processes in an automated way. Powerful open-source WFMS, such as YAWL [19] or Bonita[3] make workflow-based applications development accessible to all. A first, naïve approach, to workflow-based ISO/IEC 29110 compliance would be to model each profile as an independent workflow and let the users pick the one they want to apply. There are three drawbacks to this. First, several activities and tasks of the ISO/IEC 29110 are common between profiles. Duplicating them across multiple workflows makes development and maintenance more costly. Second, as mentioned in Section 2.1, profiles are not mutually exclusive: an *Entry* level development process consists of activities and tasks also specified in the *Basic* profile. Finally, even if WFMS simplify the task of developing and customizing workflows, relying on a purely manual configuration would also be error-prone and would necessitate a thorough knowledge of the standard to guarantee the compliance of the resulting configured workflow. A more flexible and user-friendly approach should therefore improve the adoption by VSEs.

The concept of *configurable workflow* [8,10] proposes a product-line [20] approach to workflow modelling. The principle is simple: a *domain* workflow model representing all legal workflow variants is defined expressing all variation points explicitly. Those variation points are then used during configuration where the *product* workflow model is produced by activating and deactivating activities. To facilitate configuration, questionnaire-driven configuration is being used [9]. User-oriented questions are associated to variation points, offering an additional abstraction layer on top of the variation points. The consistency of the answers given by the user is enforced by the configuration engine, in the same spirit as feature-based configuration in software product lines. Based on the answers, the initial all-variants domain workflow model is pruned from the deselected elements through an *individualisation* process. The individualisation process consists of transforming the domain workflow model into a valid product workflow model. This questionnaire-driven individualisation process is supported by the Synergia tool suite [10].

2.4 Research Methodology

Our goal was to explore the feasibility of configurable workflows to assist users applying the ISO/IEC 29110 standard. For this purpose, we first read the documentation of available profiles, *Entry* and *Basic*, and represented common as well as variable tasks into a single configurable workflow, so introducing variability points. This workflow was then validated by a member of subcommittee 7 of ISO/IEC-JTC1 defining the ISO/IEC 29110 standard. Then, we defined a question for each variation point of

[3] http://www.bonitasoft.com/

the workflow. Those questions and their mapping with tasks of the workflow were also validated by the the same ISO/IEC-JTC1 member. Remaining steps, i.e. product configuration and individualisation, are supported by the Synergia tool suite mentioned hereabove.

3 Automating ISO/IEC 29110 RE via Process Configuration

In this section, we explain how we modelled the ISO/IEC 29110 *Software Requirements Analysis* activity as a configurable workflow, and illustrate one of its possible individualisations, i.e. a tailored method based on a specific combination of ISO/IEC 29110 profiles. We chose to illustrate our approach on the RE process since it was highlighted as a priority by industry partners in the NAPLES.

3.1 A Domain Workflow for ISO/IEC 29110 RE Profiles

Although four profiles are anticipated for ISO/IEC 29110, only two of them are currently available (either publicly or internally) in their final or nearly final version: *Entry* and *Basic*. Thus, we first combined the tasks from the two available profiles, common as well as variable ones. Extracting this variability was conducted manually, by comparing profiles' task lists in their respective documents. Our comparison task was somewhat simplified by the fact that both profiles use the same task numbering scheme. However, it is important to mention that, even if task's IDs are the same, their description

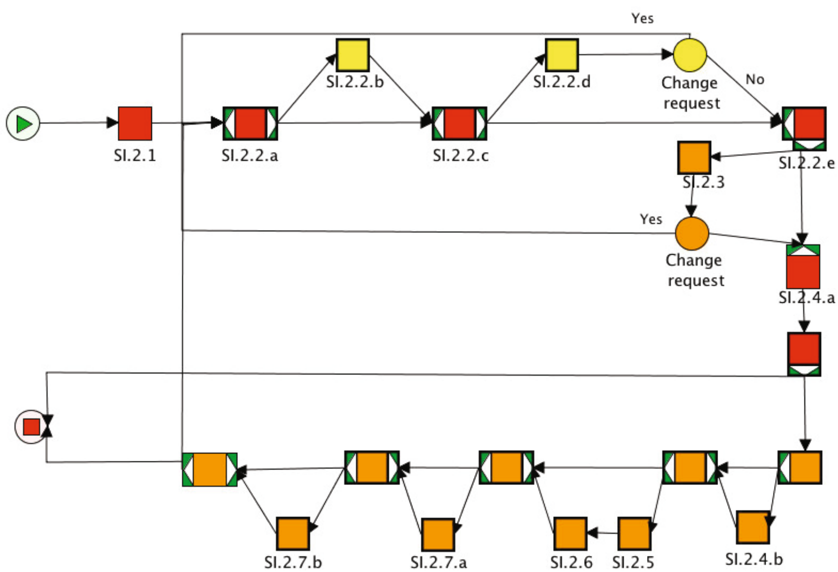


Fig. 1. ISO/IEC 29110 – Workflow of the software requirements analysis activity

can differ as the *Basic* profile might require more detailed information than the *Entry* one. Extracting the variability from the different profiles could thus be automated using tasks' IDs but should be validated by a human being to ensure that task's descriptions are the same. A complete description of the tasks appears in Table 1 where italic text denotes input and output products. The inclusion of a task in a profile is also displayed in the last two columns on the right side of the table. The corresponding YAWL workflow is depicted in Figure 1. There, tasks that are common to both profiles have a *red* (dark grey) background while *yellow* (light grey) tasks are specific to the *Entry* profile, and *orange* (grey) ones to the *Basic* profile. Triangles associated to tasks represent either a xor-join (mutually exclusive incoming transitions) or a xor-split (mutually exclusive outgoing transitions).

ISO/IEC 29110 allows combining tasks from both profiles. This means that a VSE can decide to follow the *Entry* workflow while borrowing some tasks from the *Basic* workflow, e.g. when a customer needs to perform some subprocesses in a different or more detailed manner. As long as all tasks of a given profile are included, compliance with this profile is guaranteed. Moreover, some tasks can be defined as optional in the standard for all or some profiles. For example, tasks `SI.2.5`, `SI.2.6` and `SI.2.7.b` are optional because *Software user documentation* is optional in the ISO/IEC 29110 standard. To assist users with process configuration, we identified five questions which allow to select tasks from Figure 1:

- Q1 - *Are the requirements gathered into a repository without automated versioning (e.g. in a spreadsheet) ?* – **"Yes"** or **"No"**
- Q2 - *When do the correctness and testability of the requirements specification and its consistency with the product description have to be verified?* – **"Once, when all requirements have been identified"** or **"Iteratively, until fully approved"**
- Q3 - *Do you wish to keep track of the requirements verification and validation process?* – **"Yes"** or **"No"**
- Q4 - *Do you plan to produce a Software User Documentation?* – **"Yes"** or **"No"**
- Q5 - *Do you wish to incorporate the requirements specification as a baseline in the software configuration ?* – **"Yes"** or **"No"**

Those questions have been defined with the *Questionnaire Designer* tool of Synergia [10] (see Figure 2). There, green (dark grey) boxes correspond to our five questions and yellow (light grey) boxes to the answers, called "facts". Each fact is associated to a question through a "MapQF" relationship. Dependencies between questions/facts, i.e. a question/fact is available if and only if another question/fact has been answered/selected, can also be defined but were not used in our example. The tool can then save the questionnaire in a XML format.

Then, Synergia's *C-mapper* tool is used to link questionnaire files and processes to allow configuration. Questions are then processed by *Quaestio* which interacts with the user and saves the answers. A negative answer to question *Q1* implies the selection of the task `SI.2.2.b`. The next question, *Q2*, has two possible answers which are mutually exclusive. The first answer is linked to task `SI.2.3` as all requirements are collected before verifying them, while the second one is associated to task `SI.2.2.d`. Question *Q3* is directly mapped to task `SI.2.4.a`. Question *Q4* is associated to all

Table 1. ISO/IEC 29110 – List of tasks of the software requirements analysis activity

Task ID	Task List	Entry	Basic
SI.2.1	**Assign *Tasks* to the Work Team members in accordance with their role, based on the current *Project Plan*.**	X	X
SI.2.2	**Document or update the *Requirements Specification*.**	X	X
a	Identify and consult information sources (customers, users, previous systems, documents, etc.) in order to get new requirements.	X	X
b	Gather the identified requirements.	X	
c	Analyze the identified requirements to determine the scope and feasibility.	X	X
d	Verify the correctness and testability of the *Requirements Specification* and its consistency with the *Product Description*.	X	
e	Generate or update the *Requirements Specification*.	X	X
SI.2.3	**Verify and obtain approval of the *Requirements Specification*.** Verify the correctness and testability of the *Requirements Specification* and its consistency with the *Product Description*. Additionally, review that requirements are complete, unambiguous and not contradictory. The results found are documented in a *Verification Results* and corrections are made until the document is approved by analysts. If significant changes were needed, initiate a *Change Request*.		X
SI.2.4	**Validate and obtain approval of the *Requirements Specification*.**	X	X
a	Validate that the *Requirements Specification* satisfies the needs and agreed upon expectations, including the user interface usability.	X	X
b	Document the results found in SI.2.4.a in a *Validation Results* and corrections are made until the document is approved by the customer.		X
SI.2.5	**Document the preliminary version of the *Software User Documentation* or update the present manual, if appropriate.**		X
SI.2.6	**Verify and obtain approval of the *Software User Documentation*, if appropriate.** Verify consistency of the *Software User Documentation* with the *Requirements Specification*. The results found are documented in a *Verification Results* and corrections are made until the document is approved by analysts. If significant changes were needed, initiate a *Change Request*.		X
SI.2.7	**Incorporate the *Requirements Specification*, and *Software User Documentation* to the *Software Configuration* in the baseline.**		X
a	Incorporate the *Requirements Specification* to the Software Configuration in the baseline.		X
b	Incorporate the *Software User Documentation* to the Software Configuration in the baseline.		X

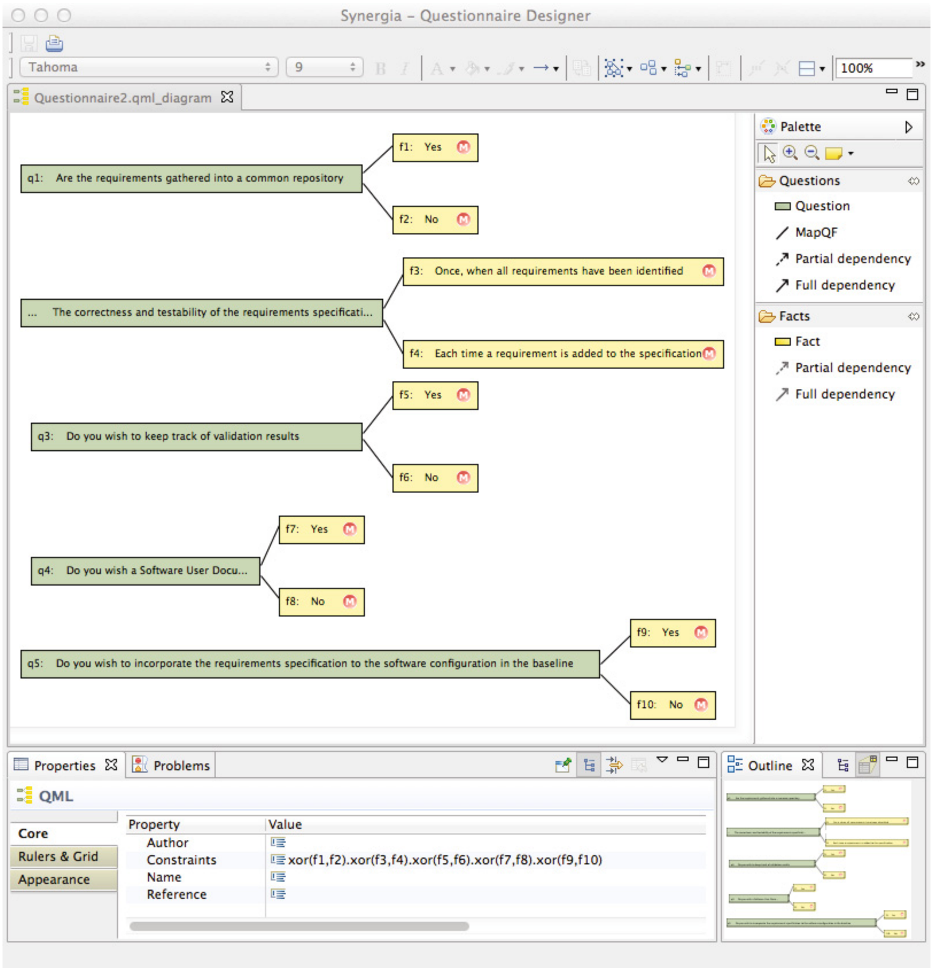


Fig. 2. Questionnaire Designer GUI

tasks that include the software user documentation, namely `SI.2.5`, `SI.2.6` and `SI.2.7.b`. The answer to the last question, *Q5*, will determine the selection (positive answer) or non selection (negative answer) of task `SI.2.7.a`. Finally, process individualisation is achieved directly from *Quaestio* which derives the desired YAWL process model according to the answers as described hereafter.

3.2 Deriving an Application Workflow through Individualisation

The Synergia *Quaestio* tool is the single tool used by a final user who wants to configure her workflow. Figure 3 depicts the tool's GUI. The first task of the user is to load the XML questionnaire produced by the *Questionnaire Designer*. Then, she can answer the different questions (see upper left part of Figure 3) in a random order. Upon question

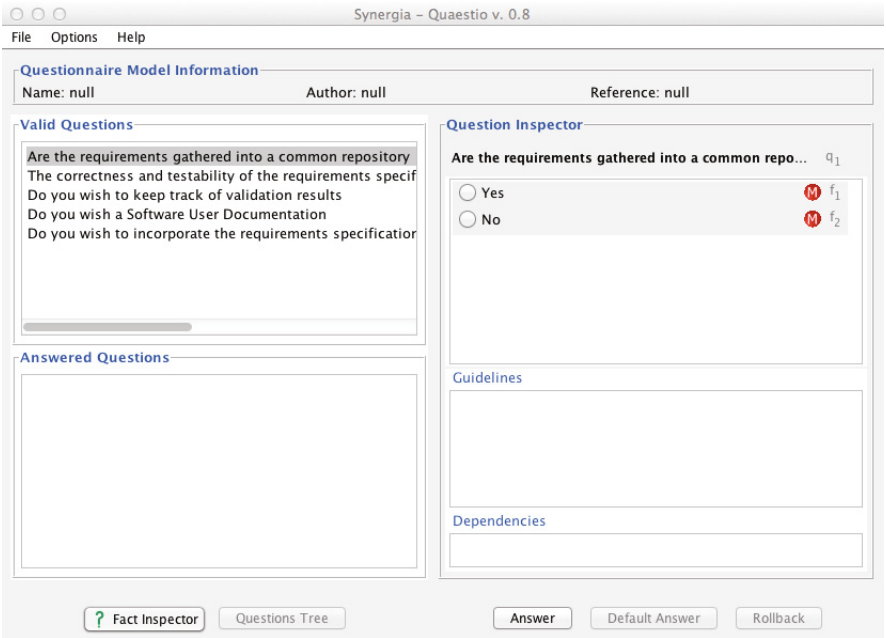


Fig. 3. Quaestio GUI

selection, possible answers appear in the *Question Inspector* part of the tool. Once a question has been answered, it will appear in the *Answered Questions* part of the tool. The user always has the opportunity to "rollback" her decisions for answered questions (*rollback* button). When all the questions have been answered, the user can individualise her process, i.e. prune the complete workflow according to her answers. For this purpose, she has to provide the tool with the paths to the workflow (.yawl) and the mapping (.cmap) previously defined. *Quaestio* will then individualise the process and save a "new" workflow containing only required tasks. A demo screencast of *Quaestio* usage is available online: `http://www.info.fundp.ac.be/~qbo/ISO29110.mov`.

4 Discussion

4.1 Lessons Learnt While Interpreting ISO/IEC 29110

Our main goal is to provide VSEs means to configure their methods according to ISO/IEC 29110. This implies that we need a deep understanding of the standard to be able to model finely a domain workflow. Even if ISO/IEC 29110 targets VSEs, its understanding still requires some basic knowledge about ISO/IEC standards which might not be available. As an illustrative example, we can mention the optionality of the *Software User Documentation* in the *Basic* profile. What is the scope of this optionality? Can the user decide to implement documentation for a given task and not others? Or is it a "global variable"? In our case, interpreting such peculiarities of the standard was not a problem as one of the authors is a member of subcommittee 7 of ISO/IEC-JTC1 defining the ISO/IEC 29110 standard. For this reason, we believe that the workflow implemented in this paper matches the standard and its intent. However, such problems could impede the adoption of the standard by its target audience, i.e. VSEs.

Based on our modelling of ISO/IEC 29110, we actually have several points to highlight and to comment back to the national bodies contributing to ISO/IEC 29110.

First, we found that VSEs used to the *Entry* profile and wishing to upgrade to the *Basic* profile may have trouble to understand how they need to perform certain tasks when going from *Entry* to *Basic*. For instance, task *SI.2.2* of the *Entry* profile contains more subtasks than the same task of the *Basic* profile. In particular, the subtask *SI.2.2.d* of *Entry* ("Verify the correctness and testability of the *Requirements Specification* and its consistency with the *Product Description*) is only present in the *Entry* and not in the *Basic* profile. After a more thorough analysis, the VSE may notice that the subtask *SI.2.2.d* of *Entry* is promoted to a full task in *Basic*, i.e. *SI.2.3.*. However, it is not explicitly mentioned in neither of the two profiles.

Second, given that the task order is defined through input/output relationships, it means that task numbers do not necessarily reflect task ordering. VSEs may not directly understand this fact. It would therefore be helpful not only to present workflows for the activity level but also for the task level. Furthermore, providing task workflows would help to disambiguate or verify the coherence of the proposed profile before publication. For instance, when modelling the configurable workflow for *Software Requirement Analysis*, we noticed that for the *Basic* profile, inputs to task *SI.2.6* on software documentation can already be obtained right after task *SI.2.2* since only the "Requirements Specification" is needed to conduct task *SI.2.6*. We are just wondering if this was intended or if instead, Task *SI.2.6* should require the "Requirements Specification [Validated]".

Overall, this preliminary experience makes us confident that providing semi-automated ways to configure a process complying to ISO/IEC 29110 is a necessity to ease VSEs uptake of the standard, especially if process improvement is targeted. Indeed, understanding the differences between the profiles was the most difficult part of the work, which paramount to decide whether a given VSE will climb to the next rung.

4.2 Threats to Validity

The first threat is related to the scope of our workflow and questionnaire models: we only focused on the requirements engineering process of the SI component of the standard. Thus, there is a risk that the approach could not scale when extended to the full SI component and/or to additional profiles (*Intermediate* and *Advanced*) to come. This risk is mitigated by the fact that the configurable workflow approach/tooling on which our research relies upon has already been applied on quite complex case-studies [21].

The second threat to validity relates to the maturity of the standard itself. Indeed, we worked on a preliminary version of the *Entry* profile. Therefore, the proposed workflow may not be fully accurate and some identified variability points may evolve. Collaborating concretely with a subcommittee 7 of ISO/IEC-JTC1 member helps us to foresee future developments of the standard and the model-oriented approach to workflow configuration we have chosen eases evolution of both workflow and questionnaire models.

Finally, our approach is currently at an early stage and thus needs more development. Furthermore, it should be trialled in the field by our partners in the NAPLES project as well as others in order to check if the proposed approach is valid. Evaluation by people familiar with the ISO/IEC 29110 standard would also be worthwhile.

5 Conclusion

In this paper, we described an approach based on configurable workflows to assist VSEs in adopting ISO/IEC 29110 compliant processes. It is implemented using the Synergia tool suite [10]. More specifically, we illustrated it with the RE process of the ISO/IEC 29110 standard. For this purpose, we first identified the variability of the RE process in the standard and represented it in a YAWL workflow. Second, we defined a user-oriented questionnaire and mapped the different answers to their corresponding tasks in the previously defined YAWL workflow. The last step is conducted by the final user who can answer those questions and the *Quaestio* tool individualises the workflow depending on her choices. The output is a YAWL workflow free of variability.

The approach is meant to lower the adoption barrier of ISO/IEC 29110 by substituting a methodology expert by a user-friendly (questionnaire-based) interface. Although, the tool might not fully replace the expert, it is likely to make ISO/IEC 29110 affordable for a wider public. The requirements workflows produced through configuration can either be followed manually (and thus used as mere documentation) or used to drive workflow which will thus enforce ISO/IEC 29110 compliance. This approach is in line with method engineering [22] techniques but is innovative in that it applies recent developments of process modelling research.

There is room for improvement. First, we would like to evaluate the approach empirically in VSEs trough pilot projects that will soon start in the context of the NAPLES project. We would like to determine to what extent our approach fosters process improvement and how it can be measured. Second, we will extend the approach to new profiles as they become available as well with the project management process of the ISO/IEC standard. So far, we have assumed a sequential ordering of development tasks as an interpretation of the ISO/IEC 29110 “input/output” approach to task ordering. Although our preliminary experience shows this works for the RE process, it may be different for others. Thus, we finally would like to investigate the suitability of “declarative workflows” [23] as an alternative approach to perform ME on ISO/IEC 29110.

Acknowledgements. This work was partially funded by the Walloon Region under the NAPLES project, the IAP Programme, Belgian State, Belgian Science Policy under the MoVES project, the BNB and the FNRS.

References

1. Humphrey, W.S.: Managing the Software Process. Addison-Wesley (1989)
2. Institute of Electrical and Electronics Engineers, Electronics Industry Association: IEEE/EIA 12207 - industry implementation of international standard iso/iec 12207 (1998)

3. ISO/IEC: ISO/IEC 29110 – Lifecycle Profiles for Very Small Entities (VSEs) (2010)
4. Laporte, C.Y., Alexandre, S., Renault, A.: Developing international standards for very small enterprises. IEEE Computer 41, 98–101 (2008)
5. Laporte, C.Y.: Contributions au génie logiciel et au développement et déploiement de normes internationales en génie logiciel pour de très petites organisations. PhD thesis, UBO, France (2009)
6. Saliou, P., Ribaud, V.: Iso-standardized requirements activities for very small entities. In: Requirements Engineering in Small Companies, pp. 145–157 (2010)
7. Alexandre, S., Laporte, C.Y.: Deployment package: Software requirements analysis - basic profile. Technical report, CETIC (Belgium) and ETS (Canada) (2007)
8. Gottschalk, F., Van Der Aalst, W., Jansen-Vullers, M.H., La Rosa, M.: Configurable workflow models. International Journal of Cooperative Information Systems 17(2), 177–221 (2008)
9. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.H.M.: Questionnaire-driven Configuration of Reference Process Models. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 424–438. Springer, Heidelberg (2007)
10. La Rosa, M., Gottschalk, F.: Synergia - Comprehensive Tool Support for Configurable Process Models. In: BPM, Demos (2009)
11. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
12. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: state-of-the-art review. Journal of Universal Computer Science 16(3), 424–478 (2010)
13. Van De Weerd, I., Versendaal, J., Brinkkemper, S.: A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management. In: Proceedings of REFSQ 2006, pp. 97–112. Springer (2006)
14. Mirandolle, D., van de Weerd, I., Brinkkemper, S.: Incremental Method Engineering for Process Improvement - A Case Study. In: Ralyté, J., Mirbel, I., Deneckère, R. (eds.) ME 2011. IFIP AICT, vol. 351, pp. 4–18. Springer, Heidelberg (2011)
15. 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)
16. Karlsson, F., Agerfalk, P.: Method configuration: adapting to situational characteristics while creating reusable assets. Information and Software Technology 46(9), 619–633 (2004)
17. Kelly, S., Lyytinen, K., Rossi, M.: Metaedit+ A Fully Configurable Multi-user and Multi-tool Case and Came Environment. In: Constantopoulos, P., Vassiliou, Y., Mylopoulos, J. (eds.) CAiSE 1996. LNCS, vol. 1080, pp. 1–21. Springer, Heidelberg (1996)
18. Cervera, M., Albert, M., Torres, V., Pelechano, V.: A Methodological Framework and Software Infrastructure for the Construction of Software Production Methods. In: Münch, J., Yang, Y., Schäfer, W. (eds.) ICSP 2010. LNCS, vol. 6195, pp. 112–125. Springer, Heidelberg (2010)
19. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N.: Modern Business Process Automation - YAWL and its Support Environment. Springer (2010)
20. Pohl, K., Böckle, G., van der Linden, F.J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc. (2005)
21. Gottschalk, F., Wagemakers, T.A.C., Jansen-Vullers, M.H., van der Aalst, W.M.P., La Rosa, M.: Configurable Process Models: Experiences from a Municipality Case Study. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 486–500. Springer, Heidelberg (2009)
22. Ralyté, J., Mirbel, I., Deneckère, R. (eds.): ME 2011. IFIP AICT, vol. 351. Springer, Heidelberg (2011)
23. van der Aalst, W., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science - Research and Development 23, 99–113 (2009)

Using Functional Defect Analysis as an Input for Software Process Improvement: Initial Results

Tanja Toroi[1], Anu Raninen[1,2], and Hannu Vainio[1]

[1] University of Eastern Finland, School of Computing, Kuopio, Finland
{tanja.toroi,anu.raninen,hannu.vainio}@uef.fi
[2] Lero – The Irish Software Engineering Research Centre, University of Limerick, Ireland

Abstract. In this paper we present how functional defect analysis can be applied for software process improvement (SPI) purposes. Software defect data is shown to be one of the most important available management information sources for SPI decisions. Our preliminary analysis with three software companies' defect data (11653 defects in total) showed that 65% of all the defects are functional defects. To better understand this mass, we have developed a detailed scheme for functional defect classification. Applying our scheme, defects can be classified with accuracy needed to generate practical results. The presented scheme is at initial stages of validation and has been tested with one software company's defect data consisting of 1740 functional defects. Based on the classification we were able to provide the case organization with practical improvement suggestions.

Keywords: functional defects, defect data analysis, process improvement.

1 Introduction

Software defect analysis is recognized as an effective and important approach to software process improvement (SPI) [1]. Robert Grady has stated that defect analysis, tracking and removing the major sources of defects offer the greatest short-term potential for improvements [2]. However, despite its importance the defect data is rarely utilized in process improvement efforts of software companies [3].

Previous research has shown that the classification of defects is important when aiming at measurement-based process and product improvement [4]. In addition, the defect classifications can be used to identify product and process problems [5] and to improve the testing and/or inspection activities [6]. There are numerous defect classification schemes available in the literature. To name a few, IEEE provides a Standard Classification for Software Anomalies [7] and IBM has generated Orthogonal Defect Classification (ODC) [8]. In addition, Beizer [9] and Humphrey [10] have presented their defect classification schemes. Unfortunately, for our purposes, defect classification schemes published are too general in nature and classify defects at a rough level.

Our preliminary analysis with three software companies' defect data (11653 defects in total) showed that 65% of the defects stored in the companies' databases are functional defects, i.e. defects in computation and/or functional logic [11]. In order to

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 181–192, 2012.

be able to use the defect data for process improvement purposes the functional defects had to be understood in more detail. To accomplish this, a more detailed defect classification of the functional defects was necessary to be conducted. However, there are not many defect classification schemes available for this purpose.

Beizer has defined a defect taxonomy [9] in which functional defects are divided in seven subclasses. We applied Beizer's functional defect classification for one software company's defect data consisting of 1740 functional defects. After applying Beizer's classification we noticed that over half of the functional defects (58%) were situated in one defect subtype, Feature/Function correctness. These results were not very useful in practice; over half of the defects remain in a single class. Hence, the functional defect classification was not detailed enough to identify the main problem areas.

In this paper, we present a detailed scheme for the classification of functional defects. The detailed classification scheme is an initial version based on analyzing defect data from one software company, including 1740 functional defects. Applying the classification scheme was encouraging: the result was easily recognizable inputs for process improvement. It appears that applying our scheme, the problems areas of software development and testing processes can be identified. Hence, testing can be focused on certain major issues. In addition, process improvement actions can be justifiably targeted to the problematic areas identified based on the defect data classification.

The aim of this paper is to present the initial results of applying our functional defect classification scheme and make the scheme available for other researchers and practitioners. We have already received feedback and improvement suggestions from our first case organization and are currently validating the scheme with the more data from other companies.

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

2 Research Setting

It is shown that software defect data is one of the most important available management information sources for software process improvement decisions [2]. 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 [11]. The study was conducted using defect data from three software companies consisting of 11653 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 in order to provide detailed information to be applied for process improvement purposes.

The initial study presented in this paper was conducted in one software company in the beginning of 2012. The case organization of the study is a Finnish software company with 18 employees. The organization has 9 employees in development and maintenance and 4-6 in testing. The company produces commercial off-the-shelf (COTS) products. An open source, web-based defect tracking system Mantis[1] is used in the company.

The results of the preliminary study conducted in 2011 showed that over half of the defects stored in the defect databases are functional defects (65%). In order to utilize defect data for process improvement purposes, functional defects had to be understood in more detail. Hence, the research problem of the study is: How functional defects should be classified so that the result provides practical inputs for software process improvement? In addition, we wanted to test our functional defect classification scheme and make the scheme available for other researchers and practitioners.

3 Functional Defect Classification

In this section we present the general defect distribution scheme applied in our previous study [11]. In addition, we present the functional defect classification by Beizer [9] and our own more precise initial scheme based on it.

3.1 General Defect Distribution Scheme

The defect distribution scheme applied in our preliminary study [11] is presented in Table 1. The scheme is a combination of the schemes by Beizer [9] and Humphrey [10]. It divides defects in ten types. We applied the scheme for three software companies defect data consisting of 11653 defects (see Section 4.1). The most common defects in every company were functional defects (65%), i.e. defects in computation and/or functional logic. In order to find out the real problems behind these functional defects they had to be investigated in more detail.

Table 1. General defect distribution scheme applied

ID	Defect Class	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

[1] http://www.mantisbt.org/

3.2 Beizer's Taxonomy for the Functional Defects

In the literature, only a few functional defect taxonomies exist. In order to classify the functional defects in a more detailed manner, we applied Beizer's taxonomy [9] which has seven subcategories for functional defects. In addition to the functional defects, Beizer's taxonomy also includes structural defects. Structural defect type includes "Control Flow and Sequencing" (e.g. path left out, unreachable code, improper nesting loops) and "Processing" (algorithmic, arithmetic expressions, initialization) defects. We added the structural defect types to the classification because control flow and sequencing, and processing defects are actually quite similar to functional defects. Often failures that are caused by a sequencing defect appear as erroneous system functionality. Hence, the failure is entered into the defect database as a functional defect. Based on our experience, the defect types or descriptions of the defects are seldom altered after being entered to the database. The Beizer's taxonomy of the functional and structural defects is presented in Table 2.

Table 2. Taxonomy of the functional and structural defects [9]

ID	Defect type	Description
21xx	Feature/ Function correctness	Feature not understood, feature interaction
22xx	Feature Completeness	Missing feature, duplicated, overlapped feature
23xx	Functional Case Completeness	Missing case, duplicated, overlapped case, extraneous output data
24xx	Domain bugs	Domain misunderstood, boundary location error, boundary closure
25xx	User Messages and Diagnostics	False warning, failure to warn, wrong message, spelling, formats
26xx	Exception Condition Mishandled	Exception conditions are not correctly handled, wrong exception-handling mechanisms used
29xx	Other functional bugs	Other functional bugs that are not mentioned in the previous rows.
31xx	Control Flow and Sequencing	(Structural bug) Path left out, unreachable code, improper nesting loops, loop termination criteria incorrect
32xx	Processing	(Structural bug) Algorithmic, arithmetic expressions, initialization, cleanup, precision

3.3 Improved Functional Defect Classification Scheme

The main problem with applying Beizer's taxonomy was that it is not detailed enough to identify the practical targets for process improvement. The defect type "Feature/Function correctness" included most of the defects in the end. In addition, a

“Feature completeness” defect is often hard to distinguish from a “Function/Feature correctness” defect. Further, due to the nature of the defect data analyzed, a “Functional case completeness” defect was quite impossible to detect.

To avoid the problems stated above, we developed a more detailed scheme in which a “Feature/Function correctness” defect type is divided into subtypes. In addition, we added “Control flow and sequencing” and “Processing” defect types to our functional defect scheme. Further, in our scheme “Domain bugs” refer to application domain defects not value ranges of the variables. Our initial functional defect classification scheme is presented in Table 3.

Table 3. Initial functional defect scheme

6	**Functional 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	Domain Bugs	Application domain bugs, subcategories vary between companies (e.g. taxes, allowances, materials)
6.3	Exception condition mishandled	Defects in exception handling.
6.4	Feature Completeness	Feature is executed inadequately.
6.5	Feature / Function Correctness	Implementation of feature / function is incorrect.
6.5.1	Copying data	Defects in copying data between systems / databases. Difficulties in making backups.
6.5.2	Default values and initial states	Defects in programs default values e.g. programs default selection causes failures in software.
6.5.3	Installation	Problems during installation of the developed program.
6.5.4	Retrieval, update and removal of data	Relates to refreshing the screen. Data inputs from user doesn’t update properly to the screen.
6.5.5	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.
6.5.6	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.
6.6	Processing	Defects in processing, calculations.
6.7	User messages and diagnostics	User messages are incorrect. Printing on screen / paper, defects in reports.

4 Applying Functional Defect Analysis in Process Improvement

In this Section we present the results of the defect classification after the first general classification, after applying Beizer's taxonomy, and after applying our initial functional defect scheme. In addition, improvement suggestions collected from the case organization are discussed.

4.1 General Defect Distribution

In our preliminary study [11] we applied the general defect classification scheme presented in Table 1 for three software companies defect data consisting of 11653 defects. The result of the defect classification is presented in Figure 1. From the Figure, it can be seen that by far the most common defect type in every company is "Function" defect type (total of 7574, 65%). The second most common defect types are "User Interface" (total of 1870 defects, 16%), "Assignment" (total of 700 defects, 6%) and "Checking" (total of 688 defects, 5.9%). "Requirements" (total of 24 defects, 0.2%) and "Documentation" (total of 47 defects, 0.4%) are the rarest defect types.

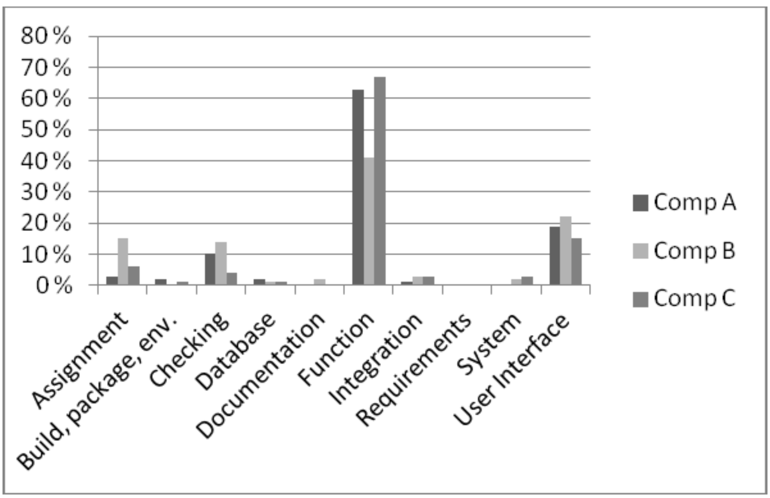


Fig. 1. Defect distribution after the first classification

4.2 Functional Defects Classified According to Beizer's Taxonomy

In order to make the defect classification data usable in practice, we needed to better understand what the mass of functional defects consisted of. Hence, we applied functional defect taxonomy by Beizer [9] to classify the defects in a more precise manner. The preliminary classification was conducted for one software company's defect data consisting of 1740 functional defects.

The defect distribution is presented in Figure 2. In practice, we ended up formatting Beizer's taxonomy. We did not include "Other functional defects" because this is too vague to tell anything about the defects nature. In addition, we did not identify a single "Functional case completeness" defect. This may be due to the cursory description of the defect type. From the Figure 2, it can be seen that the defect type "Feature/Function correctness" is remarkably more common than the other defect types. "Feature/Function correctness" includes 58% of the defects. The rest of the defect types include evenly from 4 to 13 percent of the defects. One exception is "Exception condition mishandled" type which includes only 0.29% of the defects.

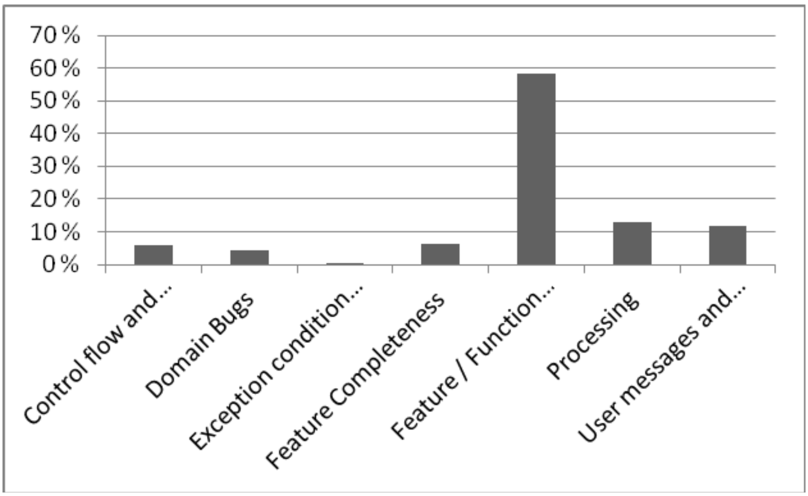


Fig. 2. Functional defect distribution classified according to Beizer's taxonomy

4.3 Functional Defects Classified According to Our Own Defect Scheme

The defect classification according to Beizer's taxonomy was still not detailed enough for process improvement purposes of the case organization. They 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 defined a more detailed scheme for the functional defects. The scheme is presented in Table 3. We applied the scheme for the same 1740 defects as with the Beizer's taxonomy. The results can be seen in Figure 3.

The distribution of the defects is notably more even applying our scheme. The most common functional defect type is "Retrieval, update and removal of data" (24% of the defects). The second most common defect types are "Processing" (13%) and "Default values and initial states" (13%). "Exception condition mishandled" is the most uncommon defect type (only 0.3% of the defects).

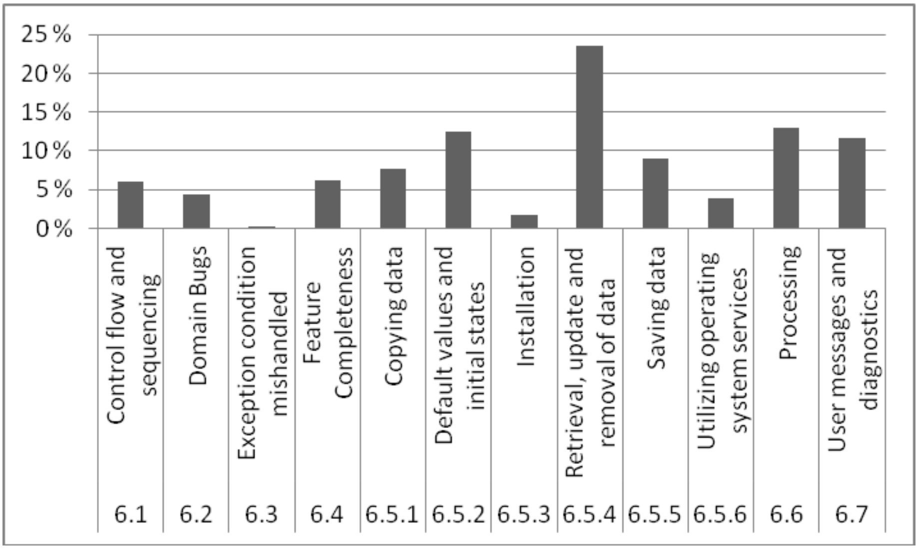


Fig. 3. Functional defect distribution classified according to our own scheme

4.4 Further Development of the Functional Defect Scheme

The functional defect scheme presented in this paper is an initial version and applied only to one software company's defect data. The goal of the classification scheme is to be general enough to be applied in most software companies. To achieve this, the classification scheme must be documented unambiguously. In addition, the defect types must be general enough to be applied in several companies. Yet, the defect types must be particular enough to get significant results out of the classification exercise. In order to improve the functional defect classification scheme, we collected feedback from our case organization.

The feedback was collected in a three hour workshop organized in the premises of the case organization. In the workshop the classification scheme and result of the classification were presented for the participants from the case organization and then discussed in detail.

Mainly the case organization was happy with the classification scheme and the result of the classification. However, there were two defect types that caused confusion. The case organization had problems understanding the defect types "6.5.6 Utilizing operating system services" and "6.5.2 Default values and initial states". The representatives of the case organization questioned whether these two types were already included in the higher level of the classification. They wanted to know what was the difference between "6.5.6 Utilizing operating system services" and "10 User Interface" defect types. Also, they wanted to know why "6.5.2 Default values and initial states" defects were not included in the "4 Data" type.

Based on the feedback the description of these two defect types was written in more detail. The improved descriptions that highlight the difference between these defect types are presented in Table 4.

Table 4. The difference between defect types Data & Default values and initial states and User Interface & Utilizing operating system services

Defect Type	Description
4 Data (see Table 1) vs.	Database structure and content. For example; bug due to error in the structure of the database, bug due to the availability of the data, bug due to difficulties in obtaining the data from the database.
6.5.2 Default values and initial states (see Table 3)	Defects related to default values and initial statuses of the software. Default values or initial states that prevent the user from using the system as intended. For example; the user is presented with wrong and/or wrong sized screens as a default.
	6.5.2 Distinguishes from type 4: Default values and initial states are different from data defects as they are regarded different by nature. All default values do not necessarily derive from database.
10 User Interface (see Table 1) vs.	Procedure calls and references, I/O, user formats. For example; incorrect output data from the user point of view, a problem with usability and/or trivial defect in layout (e.g. overlapping windows)
6.5.6 Utilizing operating system services (see Table 3)	Defects related to utilizing the services of the operating system of the computer on which the software is installed. For example; defects due to applying the monitors, printers and other peripherals. A defect related to Windows system (e.g. tab-order).
	6.5.6 Distinguishes from type 10: User Interface defects are more often cosmetic defects, for example, typing errors in user interface.

5 Process Improvement Suggestions Based on Functional Defect Data Analysis

Based on the functional defect data classification, the case organization is able to see their software engineering problem points from the defect point of view. The classification shows that the most troublesome issues are related to retrieving, updating and removing data, default values of the variables and forms, processing i.e. calculation, and user messages and diagnostics. The most common functional defect types of the case organization are presented in Figure 4.

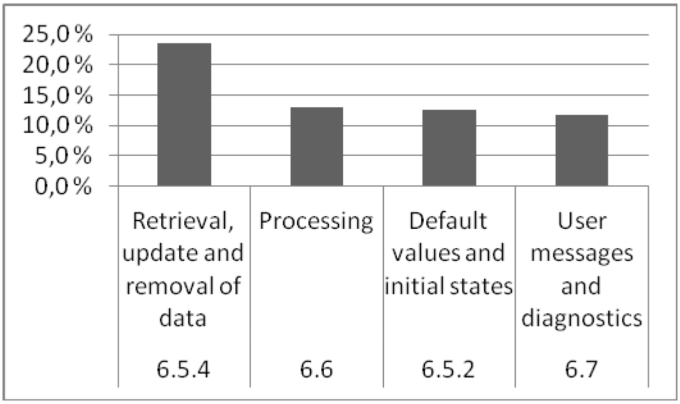


Fig. 4. The most common functional defect types in the case organization's defect data

Based on the results of the defect classification improvement suggestions were given to the case organization. The suggestions are presented in Table 5.

Table 5. Improvement suggestions related to the most common defect types

Defect Type	**Improvement suggestions**
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 [12].
Processing	Conduct **code inspections** in order to reduce the amount of bugs due to carelessness. Processing bugs are often due to the software engineer not being careful enough while coding the calculation rules to the software. Inspection is proved to be effective at identifying defects [13].
Default values and initial states	Conduct **code inspections and pair programming**. **Take test automation in use.** Test automation does not prevent the defects but would make it easier and more cost-effective to detect them from the code [14].
User messages and diagnostics	**Conduct usability testing.** This could help to find defects in user messages. **Inspect** end user reports in order to find anomalies and bugs in them.

6 Discussion

Our preliminary analysis with three software companies' defect data (11653 defects) showed that 65% of the defects were functional defects [11]. 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 software process improvement decisions [2]. Yet, defect data is rarely utilized properly in process improvement efforts [3].

However, in the literature, there are only a few functional defect classifications available. Beizer has developed a defect taxonomy which has subcategories for functional defects [9]. We applied Beizer's taxonomy for functional defect data (1740 functional defects). However, the results were not satisfying, over half of the defects still remained of one defect type, "Feature/Function correctness". Beizer's taxonomy was not able to properly make the problem areas of the process visible.

The main problem with applying Beizer's taxonomy was that it is not detailed enough to identify the tangible targets for process improvement. Namely, the "Feature/Function correctness" defect type is so general that far too many of the defects are of this type. In addition, "Feature completeness" type is often impossible to distinguish from the defect type "Function/Feature correctness". When the defect has been entered to the database it cannot often be known whether the feature causing a defect has been properly completed or incorrectly coded. Further, a "Functional case completeness" defect is quite difficult to identify from the defect data.

To avoid the problems stated above and to better identify the problem areas of the processes we defined a more detailed functional defect classification in which the defect type "Feature/Function correctness" is refined in more detail. We applied our scheme for one software company's functional defect data and received a more diversified defect distribution. Based on the functional defect analysis, practical process improvement suggestions could be provided. It was suggested that the company should conduct code inspections to identify simple errors earlier. In addition, they should stress the importance on unit testing to the programmers.

Further, the results of the functional defect data classification can be utilized in making decisions on whether testing should be automated. It is important for the test team to manage automated testing expectations and to outline the potential benefits of automated testing [14]. Overall, the functional defect analysis can be used in justification when more resources for verification and validation processes are required.

7 Conclusion

In this paper we have presented how functional defect classification can be applied as an input for process improvement. A functional defect classification scheme is presented and applied for one software company's defect data (1740 functional defects). Based on the results of the defect analysis, process improvement suggestions are provided. Applying our scheme, the problems areas of the development and testing processes can be identified and testing can be focused on certain major issues. In addition, process improvement actions can be targeted to the areas identified based on the defect data classification.

Our scheme is an initial version. Due to promising results reached applying it to one company's data we are currently validating it via applying it to additional companies' defect databases and collecting feedback from the companies.

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 and supported, in part, by Science Foundation Ireland grant 03/CE2/I303 1 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).

References

1. Vinter, O.: Experience-Based Approaches to Process Improvement. In: Proceedings of the 13th International Software Quality Week, San Francisco, USA (2000)
2. Grady, R.B.: Practical software metrics for project management and process improvement. Prentice Hall, New Jersey (1992)
3. Fredericks, M., Basili, V.: Using Defect Tracking and Analysis to Improve Software Quality. DoD Data & Analysis Center for Software (DACS) (1998)
4. El Emam, K., Wieczorek, I.: The repeatability of code defect classifications. In: Proceedings of the Ninth International Symposium on Software Reliability Engineering, pp. 322–333 (1998)
5. 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)
6. Freimut, B.: Developing and using defect classification schemes. Fraunhofer IESE IESE-Report No. 72 (2001)
7. IEEE standard classification for software anomalies. IEEE Std 1044-2009 (Revision of IEEE Std 1044-1993), pp. C1–C15, 7 (2010)
8. Chillarege, R., Bhandari, I., Chaar, J., Halliday, M., Moebus, D., Ray, B., Wong, M.-Y.: Orthogonal defect classification – a concept for in-process measurements. IEEE Transactions on Software Engineering 18(11), 943–956 (1992)
9. Beizer, B.: Software Testing Techniques. International Thomson Computer Press (1990)
10. Humphrey, W.: A discipline for software engineering. Addison-Wesley (2007)
11. Raninen, A., Toroi, T., Vainio, H., Ahonen, J.J.: Defect Data Analysis as Input for Software Process Improvement. In: TBP 13th International Conference on Product-Focused Software Development and Process Improvement, PROFES 2012 (2012)
12. Cockburn, A., Williams, L.: The Costs and Benefits of Pair Programming. In: Succi, G., Marchesi, M. (eds.) Extreme Programming Examined, pp. 223–243 (2001)
13. Gilb, T., Graham, D.: Software inspection. Addison-Wesley, Great Britain (1993)
14. Dustin, E., Rashka, J., Paul, J.: Automated Software Testing: Introduction, Management, and Performance. Addison-Wesley (1999)

Documenting Evolutionary Process Improvements with Method Increment Case Descriptions

Peter van Stijn[1], Kevin Vlaanderen[1],
Sjaak Brinkkemper[1], and Inge van de Weerd[2]

[1] Utrecht University, Department of Information and Computing Sciences,
P.O. Box 80.007, 3508 TA, Utrecht, The Netherlands
pcestijn@cs.uu.nl, {k.vlaanderen,s.brinkkemper}@uu.nl
[2] Vrije Universiteit Amsterdam, Faculty of Economics and Business Administration,
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
i.vande.weerd@vu.nl

Abstract. Evolutionary process improvement is a common approach to manage the complexity and risk of large software process improvement efforts. Performing SPI through a sequence of small steps allows organizations to reflect and steer the effort often and avoid failed improvements. However, few methods currently exist to structure improvement paths in a clear and concise manner. In this paper, we present a template for such a structuring method, based on Use Case Descriptions and method engineering techniques. A concise description of improvement paths allow organizations to reflect on their implementation and to guide similar improvement efforts. A case study of two large improvements within a small Dutch software company is used for evaluation.

1 Introduction

The quality of software product management processes strongly influences the productivity of the organization and the quality of its products [13]. Process improvements tend to be complex and are often implemented evolutionary [31], consistent with the definition of Software Process Improvement (SPI), which is formulated as "a continuous and *evolutionary* approach to improve a software organization's capability to develop quality software in response to customer requirements" [22]. This approach emphasizes *stepwise improvement* of the software processes, systematic assessment of an organization's current operations, and application of normative models for organizing a software operation [18].

Several methods exist to model processes within companies. A well know method in the method engineering domain is MAP [25]. In the business process management domain, BPML is a common approach [26]. Another approach from the method engineering domain is the Process Deliverable Diagram (PDD), proposed by van de Weerd et al. [28]. A variation of this method has been used to model the individual increments [31]. However, no convenient methods have been

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 193–204, 2012.

found in literature to describe a sequence of increments in a clear and concise way. Such a method would be useful for reporting and reviewing process changes. Furthermore, these overviews can prove valuable for companies that are considering a similar process improvement. By building a knowledge base of process improvement paths, those companies can use the reference improvement paths as guideline to implement their own process improvements. Therefore, the research question this article aims to answer is *How can multiple increments that are part of a larger evolutionary process improvement be documented conveniently?* In order to answer this question, we apply the use case description technique [9], as it fits with the high-level attributes that are important for describing process improvements.

This paper is organized as follows: first, an overview of the most relevant literature on software process assessment and improvement is provided. Then, the research method is described in more detail. After this, section 3 describes our approach for describing process improvement steps, after which the approach is evaluated with data obtained through a case study which is described in section 4. The paper concludes with a conclusion and discussion of further research.

2 Research Context

2.1 Related Literature

Software Process Improvement (SPI) has been researched for over twenty years, after the publication of the frequently cited work of Humphrey [17]. Process improvements generally improve the quality of the software product directly, although the impact per quality factor can differ depending on which methodology is used [3]. Many specific methods, usually in the form of maturity models, have been developed for these software process improvements. The most well-known of these are Bootstrap [20], the Capability Maturity Model (CMM) [24], Software Process Improvement and Capability dEtermination (SPICE) [14] and the Quality Improvement Paradigm (QIP) [23]. A widespread idea in SPI approaches is the use of an evolutionary approach. Changes are not implemented through one single (big-bang) transformation but by a sequence of changes (steps) over a period of time [2]. An example of a method based on such an evolutionary approach is Business Process Reengineering [11].

Measuring the current state of any (software) process is the first and a very important step for any process improvement [24,32]. Many different measurement methods exist, such as the Goal/Question/Metric (GQM) measurement method, statistical process control, practical software measurement and the balanced score care (BSC) [19]. In an earlier study, van de Weerd et al. [31] have investigated how large improvements can be separated into small increments. When doing so, process assessment can be performed between the different steps. Although van de Weerd et al. proposed a way to visualize a single method increment, no solution is provided to visualize multiple increments (steps) within one improvement. Furthermore, the drivers of the increment are not adopted in the visualization. No other publications have been found covering these aspects.

2.2 Research Approach

The aim of this research is to fill the gap identified above by providing an approach for documenting sequences of steps that form a larger software process improvement effort. This approach should provide all necessary information on individual steps needed for their reproduction, while maintaining a high level overview to quickly assess the whole improvement. Furthermore, it should allow for an easy comparison of two comparable process improvements within different companies.

In terms of the design science research framework by March and Smith [21], we *build* and *evaluate* a *model* that allows for the description of process improvements. As a basis for the design of the new documentation approach, we use the use case description template by Cockburn [9]. The result is a template in which process improvements and their steps can be described.

The proposal is evaluated in the sense of functional testing as defined by Hevner et al. [16], i.e. to discover failures and shortcomings. For this purpose, data from an earlier performed retrospective case study have been filled in the template. The case study was performed as part of a multi-case study to unravel large process improvements into smaller steps. During a set of interviews, in combination with a document analysis, data was gathered regarding several process improvements performed within a small Dutch software company during the past two years. These process improvements were then unraveled in sequences of small steps (increments), which resulted in useful information to test the template on.

During the case described in this paper, process improvements in the domain of Software Product Management (SPM) were investigated. SPM is defined as "the process of managing requirements, defining releases, and defining products in a context where many internal and external stakeholders are involved" [29]. The involved key process areas, together with the (internal and external) stakeholders and their relations are summarized in the SPM competence model [29]. As the quality of the software product management processes enhances the quality of the software product [13], improving these (software) processes is often required.

For measuring the maturity of the organization's SPM process, we employed the Situational Assessment Method (SAM) [4]. The SAM combines multiple descriptions of an organization and its product management process by performing an analysis of the organization's situational factors [5] and its product management capabilities. The results of the approach are described as solution oriented and realistic, allowing for incremental growth and requiring little effort to obtain. This assessment has resulted in a filled in SPM maturity matrix [6]. The SPM key processes are represented by the rows and are divided into four groups (the business functions). The columns 0 to 10 represent the maturity levels (where zero is low and ten is high). The letters A to F represent the capabilities. Each focus area has its own unique capabilities; the amount of capabilities within a focus area varies from two (A- B) to six (A-F). The CM suggests the best implementation order for the capabilities (from left to right). The placement of the capabilities is based on a series of interviews with experts from both the scientific world and the field of practice, and questionnaires among product managers [4]. The SPM Maturity Matrix [6] is very useful for our purpose as it allows us to quantify the process changes in the method increment case description.

3 Method Increment Case

3.1 Method Increment Case Description

A use case specifies the behavior of a system or part of a system, and is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor [7]. Use cases capture the intended behavior of the developed system, with no specific emphasis on how this behavior is implemented. In many senses, this is very comparable to a process improvement when seen as a sequence of actions that is performed [18] and that results in an observable result of value to stakeholders: an improved capability to develop quality software [22].

These similarities have been the reason to employ Use Cases as a basis for the semantic framework to document process improvements. Cockburn [9] has defined a template for the description of use cases, providing a textual overview that supports the visualization. This description consist of the title, goal, scope, pre-conditions, success end conditions, failed end conditions, actors and triggers. Furthermore, a part for the descriptions of the different steps in the case exists, and for the extensions and variations.

Many of the use case description elements can be directly applied to the domain of SPI. This has resulted in the template shown in Table 1. To emphasize the relation of this approach with the UML's Use Case Descriptions, it is named *Method Increment Case Description*. Most elements from the original use case description template were directly applicable to the method increment case description. However, some adjustments were made.

Table 1. Template of a Method Increment Case Description

Name		**Goal as a short active verb phrase**
Goal in context	#	Extended description of the goal
Scope		Identifier of the process under consideration
Primary and Secondary Stakeholders	#	Stakeholders [1..n]
Trigger		Brief description of the original problem
Pre-Conditions	#	Conditions [1..n] applicable before the process improvement
Post-Conditions	#	Conditions [1..n] applicable after the process improvement
Increment Path	#	Path steps [1..n] - *Driver, Stakeholders, Affected Capabilities*
Unordered Increments	#	Increments not part of the sequence or executed during multiple steps. - *Driver, Stakeholders, Affected Capabilities*
Failed Paths	#	Steps that were undone afterwards. - *Driver, Stakeholders* - *Failure Reason*: Why was this failed and what were the consequences?
Reference to PDD		PDD Increment relevant to this method increment case description.

The *Scope* element is used to denote the sub-domain to which the process improvement applies. This can be described in terms of an existing process framework, such as *product planning* in the case of SPM, or as a brief description. In use case descriptions, the *Level* is used to describe whether it is related to a summary, a primary task, or a sub-function. As we don't have a similar way of categorizing our process improvements, we have left it out of the template.

The *Trigger* in the context of use case descriptions is the action upon the system that starts the use case. Similar to this, we interpret the trigger as a description of the problem or dissatisfaction that initiates the process improvement effort. The elements *Preconditions* can serve as a more detailed elaboration of the driver. If possible, these preconditions should be described in a (semi-)formal manner, such as with the capabilities of the SPM Maturity Matrix [6] in the case of the SPM domain. When no formal method is available, a short textual description can be provided.

To allow for a description of the change of state after the successful implementation of a process improvement, we have renamed the element *Success End Condition* to *Post-Conditions*. The changes in process maturity should also be described in a (semi-)formal manner, e.g. a low maturity in the area of portfolio management or low employee satisfaction amongst developers. When no formal method is available, a short textual description can be provided.

For describing the steps within a software process improvement, we use three elements. Successful improvements paths are described in the element *Increment Path*, renamed from *Description*. Paths that failed, i.e. did not result in the desired end state, can separately be described in the element *Failed Paths*, renamed from *Failed End Condition*. As some steps cannot chronologically be placed inside the main *increment path*, we have added *Unordered Increments*.

All steps of the improvement path elements described above should be recorded in a clear and concise manner in order to allow a good overview. Besides a one-line description of the step, the following three attributes should be described.

Driver: The driver(s) or rationale of the increment need to be described briefly to allow for later analysis of the improvement. Furthermore, this allows other companies that consider the same improvement to determine whether or not they also need the specific increment.

Stakeholders: As with use cases, the stakeholders involved in the specific step should be described. This can aid when determining shifts in roles.

Affected Capabilities: For every step in the improvement path, the affected capabilities are quantified to show the exact consequences of an individual improvement step on the processes.

As with use case descriptions, the element of time is not included in the variant method for SPI descriptions. The rationale behind this is that the time needed for the process improvement depends largely on the available (human) resources, the company culture (e.g., resistance to change) and other situational factors. Furthermore, the exact time and dates of a increment are often difficult to discover after a large process improvement. Additionally, even if the start and end date of a large process improvement can be discovered, this will still prove to be very hard (if not impossible) for individual increments. By not including the element of time, these issues are acknowledged and it is emphasized that the different steps are more relevant than the exact timings of these steps.

Finally, we recognize that there are often multiple approaches to implement a certain process improvement, often depending on the situational factors of the company. The original use case description template provides for this in the form of *Sub-variations*. We have removed this element from the method increment case definition template as the description's purpose is to capture a single process improvement effort. When bundled, method increment case descriptions can provide insight into alternative paths for similar process improvements.

The final attribute of the original use case description template is *Extensions*. This attribute's purpose is to describe each step that is altered, the condition under which this happens and the actions or sub-use cases that are performed extra. However, we have removed extensions in the method increment case description, due to the reason that extensions can also be seen as individual improvement steps that precede or succeed the step they are supposed to extend. As the goal is to find individual improvement steps to a level of granularity as low as possible, these extensions are considered as individual steps themselves. Thus, these extensions or extended paths fall under the attribute *Increment Path*.

3.2 Linking Method Increment Case Descriptions to PDD's

A field related strongly to SPI is that of method engineering, which is defined "the engineering discipline to design, construct and adapt methods, techniques and tools, for the development of information systems" [8]. These methods are based on a specific way of thinking and consist of "directions and rules, structured in a systematic way in development activities with corresponding development products". Method engineering that takes the situational factors of the concerned company into account is referred to as situational method engineering [15]. During the last several years, several modularization constructs have been proposed for situational method engineering[10], [1] and [12].

The core concept within our research is the method fragment. Method fragments [15] are defined as "... a description of an IS engineering method, or any coherent part thereof". Method fragments consist of a process part and a product part, with a link between these two parts. The general approach to visualize a method is by means of a Process-Deliverable Diagram (PDD), which is a meta-modeling technique that is based on a combination of a UML activity diagram and a UML class diagram [28].

In an earlier study, van de Weerd et al. [30] applied the PDD modelling technique to describe process improvements, resulting in the definition of PDD increments. An example of such a PDD increment is shown in Figure 1.

We have included a reference to a PDD increment in the method increment case description template, in order to facilitate the extension of the case description with a visual diagram. We believe that this will increase the clarity of the process improvement description, and that it will provide more insight into the impact of the described changes.

4 Evaluation

The proposed template has been evaluated with the results from a case study at a small Dutch software company with 25 employees, providing a software application and accompanying services to fulfill the need of achieving a higher efficiency from the utilization of human resources. The product has around 400 customers, mostly local authorities, insurance companies and banks, with a total of 300.000 users utilizing the product. In the period between 2009 and 2011, the case company implemented several process improvements in the areas of requirements management, release planning, and development processes.

At the start of the process improvement efforts, the organization was developing in an unstructured, waterfall-like fashion. Roles were not clear and customers were not involved in the process. Furthermore, the documentation of the requirements was problematic, due to which the requirements could not be traced back to the original feature request and corresponding customer who placed it. The fact that requirements were often not (well) documented ultimately resulted in requirements getting lost.

In order to measure the maturity of the organization's SPM process at that moment (June 2009), we employed the Situational Assessment Method [4], described in section 2.2. The situation is visualized in the form of an SPM Maturity Matrix in Table 2 with the light gray boxes indicating the original maturity.

Table 2. Partial SPM Maturity Matrix for the Case Company

Focus Area		**Maturity Levels**										
Title	**Code**	**0**	**.9**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Requirements Management												
Requirements Gathering	RG		A		B	C		D	E	F		
Requirements Identification	RI			A			B		C			D
Requirements Organizing	RO				A		B		C			
Release Planning												
Requirements Prioritization	RP			A		B	C	D			E	
Release Definition	RD			A	B	C				D		E
Release Definition Validation	RDV					A			B		C	
Scope Change Management	SCM				A		B		C		D	
Build Validation	BV				A			B		C		
Launch Preparation	LP		A		B		C	D		E		F

After two years, the maturity of the requirements management process has increased significantly, due to an improved requirements management workflow, improved tooling and better improved customer involvement. Currently, all requirements are centrally stored and rewritten to product requirements, while a connection with the original market requirement (and information such as related customer) remains. The dark gray boxes in Table 2 indicate the added capabilities after the process improvements.

The observed process improvements at the case company can be grouped in terms of requirements management, release planning, customer involvement, tooling, and development improvements. In order to evaluate the proposed template, the results of the case study have been described in five method increment

Table 3. Method Increment Case for Requirements Management Tool Support

Name	**Improving the Requirements Management Tool Support**
Goal in context	# Put requirements at disposal of all relevant stakeholders # Improve identification and organization of requirements
Scope	Requirements Management and Tooling.
Stakeholders	# Management Team # Consultants # Support Specialist
Trigger	Requirements gathering and communication is inadequate
Pre-Conditions	# A simple tool exists to centrally store and organize requirements and keep track of basic information. # Inadequate implementation of RG:B.
Post-Conditions	Better Req. Mngmt. maturity in the SPM maturity matrix. - Improved Req. Gathering, Identification and Organization. - Improved Scope Change Management.
Increment Path	1. Disbanding of the requirements organization tool *Debby* - *Driver*: Tool has become outdated, unsupported with an increased risk of failure, and unused by employees. - *Stakeholders*: Managing Director, Requirements Gatherers - *Affected Capabilities*: Remove RG:B; RO:A; RO:B. 2. Introduction of (hardcopy) leaflets to store the requirements (as user stories) - *Driver*: Requirements were not documented anymore and got lost - *Stakeholders*: Product Manager, Requirements Gatherers. - *Affected Capabilities*: Add RI:A. *Further details omitted due to space restrictions*
Unordered Increments	-
Failed Paths	-
Reference to PDD	Figure 1

cases; preparing the introduction of Scrum, the introduction of Scrum, adjusting the Scrum implementation, improving requirements management tool support, and improving customer involvement. These cases were all described using a method increment case description. However, due to space limitations, we only present a part of these descriptions here.

Table 4. Method Increment Case Description for Adjusting the Scrum Process

Name	**Adjusting the Scrum Development Process**
Goal in context	# Put requirements at disposal of all relevant stakeholders # Improve identification and organization of requirements
Scope	Development Process (Scrum)
	...Details omitted due to space limitations...
Unordered Increments	# Include "attitude towards Scrum" in yearly employee assessments. - *Driver*: Increase employee motivation towards Scrum. - *Stakeholders*: Management Team; Developers. # Initiate a continuous improvement project. - *Driver*: Employees needed time apart from the sprints to improve processes, so as to increase employee motivation towards Scrum. - *Stakeholders*: Management Team; Developers.
Failed Paths	# Removing the daily stand-up meetings. - *Driver*: Decrease resistance from development (who felt restricted and controlled by the daily stand-up meetings). - *Stakeholders*: Scrum Master. - *Failure Reason*: Daily stand-up meetings were considered essential in the Scrum process by the management board.

Table 3 shows a method increment case description for the tooling improvements. The goal during this improvement effort was to improve the requirements management process by putting the requirements at the disposal of all relevant stakeholders instead of only the product manager, and to improve the amount of information available for each requirement. In this specific method increment case description, we did not see any failed paths. Some tools were introduced and later replaced, but they were an essential step in the increment path. Furthermore, the increments were all executed in a logical flow which is the reason that no unordered increments have been described.

To demonstrate that these elements are in fact relevant, we provide an example of another method increment case description in Table 4, which is related to the adjustments made to Scrum after its initial introduction. Under *Unordered elements* we see some continuous activities, while under *Failed activities* we see an example of a change that was deemed unsuccessful.

In addition, we applied the PDD modeling technique described in section 3.2 to the described process improvements. This has resulted in the PDD increment described in Figure 1. The PDD corresponds with the changes in Table 3.

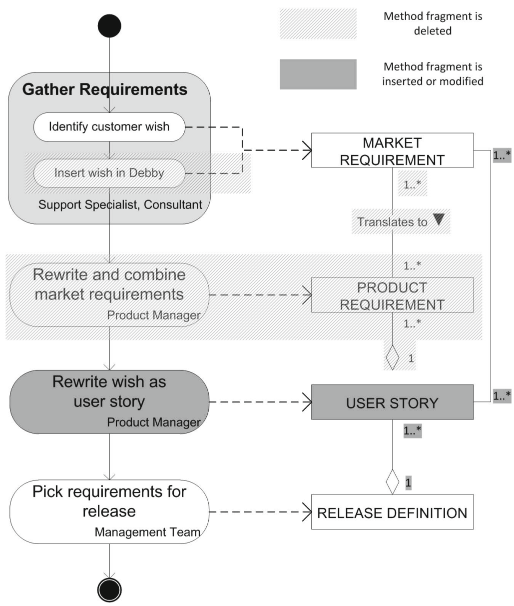


Fig. 1. PDD of the Method Increment Between the Initial and Final Situation

Describing process improvements using the method increment case description template leads to a better insight into the drivers behind a specific process improvement effort. It provides insight into the possible issues related to certain process improvements and it provides a direct link between specific goals, detailed process attributes (such as the SPM capabilities), pre-conditions and post-conditions. Ultimately, it facilitates knowledge sharing by standardizing the gathering of relevant process improvement information.

The combination of process improvement modeling using PDDs with schematic improvement modeling according to the method increment case definition allows for a thorough and semantically rich approach to the description of process improvements. By coupling a PDD with the method increment case description, it is easier to gain quick insight into the impact of certain changes. It enhances communication and it facilitates correct documentation of the processes.

5 Conclusions and Further Research

The research question of this article was *"How can multiple increments that are part of a larger evolutionary process improvement be documented conveniently?"*. The question is answered by proposing a template to describe multiple increments part of a larger process improvement. The template is based on use case descriptions and is referred to as a *method increment case description*. The advantage of the resemblance with use case descriptions is that the technique is easy to learn and very recognizable, as use cases are widely applied in the information sciences. The template provides a high-level overview of the process improvement, detailing the individual steps with more low-level descriptions.

Evaluation of this new documentation method has been performed using the data gathered during a case study within a small Dutch software company. Two cases of large process improvements were examined resulting in five method increment cases. Descriptions of two cases of large process improvement were examined resulting in five method increment cases. One of the cases is described in this article. The case study results indicate that the method increment case description template is adequate to document these improvements in a clear and concise way. The final template, after evaluation, is shown in Table 1.

The method increment case description presented in this paper is part of a larger project that aims at developing an online knowledge system for incremental process improvement [27]. Method increments play an essential role within this infrastructure, but we have inadequate means to describe them at this point. An important challenge lies in formalizing the link between the modeling approach employed in Section 3.2 with the schematic description of method increments using method increment case descriptions. Furthermore, we foresee two types of use for method increment case descriptions. The first use is in retrospect, where it can be a tool to record experiences from practitioners. This experience forms a key component in situational process improvement. The challenge of this approach will lie in motivating practitioners to spend time on documenting past experiences. They will need a strong incentive to do so, and we think that the creation of a public process improvement knowledge management system can provide this incentive. On the other hand, method increment case descriptions can form the basis for the description of future process improvements. They can be used to structure relevant knowledge required to implement process changes step by step. However, this approach has not yet been investigated.

In addition, the approach has only been used in the SPM domain. This has resulted in the usage of the SPM Maturity Matrix for coupling the consequences of the individual improvement steps to the influenced SPM capabilities. When using the template for other domains, other maturity measurements will need to be used or developed. Full validation of this template has not been achieved by performing the case study. Especially external validity cannot be ascertained. Full validation should therefore be part of future research.

References

1. Ågerfalk, P.J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S., Ralyté, J.: Modularization Constructs in Method Engineering: Towards Common Ground? Situational Method Engineering: Fundamentals and Experiences 244, 359–368 (2007)
2. Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O.: A Conceptual MAP of Software Process Improvement. Scandinavian Journal of Information Systems 13, 81–101 (2001)
3. Ashrafi, N.: The Impact of Software Process Improvement on Quality: in Theory and practice. Information and Management 40, 677–690 (2003)
4. Bekkers, W., Spruit, M., van de Weerd, I., van Vliet, R., Mahieu, A.: A situational assessment method for software product management. In: Alexander, T., Turpin, M., van Deventer, J. (eds.) Proceedings of the European Conference on Information Systems, Pretoria, South-Africa, pp. 22–34 (2010)
5. 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 International Workshop on Software Product Management, pp. 41–48. IEEE Computer Society, Washington, DC (2008)
6. 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)
7. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (1999)
8. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
9. Cockburn, A.: Structuring Use Cases with Goals. Technology 84121(801), 1–13 (1997)
10. Cossentino, M., Gaglio, S., Henderson-Sellers, B., Seidita, V.: A metamodelling-based approach for method fragment comparison. In: Proceedings of the 11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2006) (2006)
11. Davenport, T.H.: Process innovation: reengineering work through information technology. Ernst & Young, New York (1993)
12. Deneckère, R., Iacovelli, A., Kornyshova, E., Souveyet, C.: From Method Fragments to Method Services. In: Proceedings of EMMSAD 2008 (2008)
13. Ebert, C.: The Impacts of Software Product Management. Journal of Systems and Software 6(80), 850–861 (2007)

14. El Emam, K.: SPICE: The theory and practice of software process improvement and capability determination (1997)
15. Harmsen, F., Brinkkemper, S., Oei, J.L.H.: Situational method engineering for informational system project approaches. In: Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle, pp. 169–194. Elsevier Science Inc., New York (1994)
16. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
17. Humphrey, W.: Managing the Software Process, 1st edn. Addison-Wesley Publishing Company, Massachusetts (1989)
18. Iversen, J.H., Mathiassen, L., Nielsen, P.A.: Managing Risk in Software Process Improvement: An Action Research Approach. Mis Quarterly 25(3), 395–433 (2004)
19. Komi-Sirvio, S.: Development and Evaluation of Software Process Improvement Methods. Ph.D. thesis, University of Oulo (2004)
20. Kuvaja, P., Simila, J., Krzanik, L., Bicego, A., Saukkonen, S., Koch, G.: Software Process Assessment and Improvement - The Bootstrap Approach (1994)
21. March, S.T., Smith, G.F.: Design and natural science research on information technology (Invited Paper). Decision Support Systems (Special issue on WITS 1992) 15(4), 251–266 (1995)
22. McFeeley, B.: IDEAL: A User's Guide for Software Process Improvement. Tech. rep., Carnegie Mellon University (1996)
23. McGarry, F., Pajerski, R., Page, G., Waligora, S., Basili, V., Zelkowitz, M.: Software Process Improvement in the NASA Software Engineering Laboratory. Tech. rep., Software Engineering Institute, Pitssburgh (1994)
24. Paulk, M.C., Weber, C., Curtis, B., Chrissis, M.B.: The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Boston (1995)
25. Rolland, C., Prakash, N., Benjamen, A.: A multi-model view of process modelling. Requirements Engineering 4(4), 169–187 (1999)
26. Thiagarajan, R.K., Srivastava, A.K., Pujari, A.K., Bulusu, V.K.: BPML: A Process Modeling Language for Dynamic Business Models, p. 239 (June 2002)
27. Vlaanderen, K., van de Weerd, I., Brinkkemper, S.: The Online Method Engine: From Process Assessment to Method Execution. In: Ralyté, J., Mirbel, I., Deneckère, R. (eds.) ME 2011. IFIP AICT, vol. 351, pp. 108–122. Springer, Heidelberg (2011)
28. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, ch. III, p. 35. Information Science Publishing (2008)
29. van de Weerd, I., Brinkkemper, S., Souer, J., Versendaal, J.: A situational implementation method for web-based content management system-applications: method engineering and validation in practice. Software Process: Improvement and Practice 11(5), 521–538 (2006)
30. van de Weerd, I., Brinkkemper, S., Versendaal, J.: Concepts for Incremental Method Evolution: Empirical Exploration and Validation in Requirements Management. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 469–484. Springer, Heidelberg (2007)
31. van de Weerd, I., Brinkkemper, S., Versendaal, J.: Incremental method evolution in global software product management: A retrospective case study. Information and Software Technology 52(7), 720–732 (2010)
32. Weinberg, G.M.: Quality Software Management: Anticipating change. Quality Software Management. Dorset House Pub. (1997)

Towards an Integration of Multiple Process Improvement Reference Models Based on Automated Concept Extraction

Simona Jeners, Horst Lichter, and Ana Dragomir

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

Abstract. A variety of process improvement reference models (IRM) such as CMMI, COBIT or ITIL support IT organizations. These reference models cover different domains (e.g. IT development, IT Services or IT Governance) but also share some similarities. There are organizations that address multiple domains and want to use different IRMs. As IRMs are described in different structures and are using different terminologies, we propose a tool based approach to extract IRMs' concepts and to normalize the terminologies. Our solution enables to semi-automatically build an integrated database of IRMs' concepts based on common meta-models and on a common terminology.

Keywords: process improvement, improvement reference models, natural language processing, CMMI.

1 Introduction

Nowadays, clients are requesting better and cheaper software products. However, the Standish Group regularly reports a high failure rate of IT-projects: 68% of IT-projects do not meet the deadlines nor achieve the requested quality or are cancelled [1]. One important factor to project success is the quality of the applied processes. Hence, more and more organizations want to establish and improve their processes systematically. Because the process improvement road is quite long and expensive it needs to be guided. To support process improvement different improvement reference models (IRM) such as CMMI (2006), ISO/IEC 15504 (2007), COBIT (2007) or Functional Safety IEC61508 (2010) can be considered and applied. 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 coordinately address different and common areas of IRMs. 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 the competition strength on the IT market.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 205–216, 2012.

One premise for organizations to be able to exploit the synergy effects of multiple IRMs and to efficiently assess them is the availability of an integrated view of IRMs allowing to compare practices from different IRMs and to identify dependencies between them. Thus, organizations can effectively and efficiently adopt and assess multiple IRMs; the efficiency increases through an automated comparison approach.

1.1 Challenges and Goals

ISO/IEC 24744 [2] defines guidelines to "allow the combination of processes from different reference models, ease the development of new models and facilitates comparison of models". The main IRM integration challenge according to this standard is that IRMs "vary in format, content and level of prescription". In the following we detail this challenge:

Each IRM Has Its Own Structure

- Often IRMs use different names for the same structure element. For example in CMMI "procedures" are called "specific" or "generic practices" while in Functional Safety they are called "requirements". While a group of processes addressing the same topic is called "domain" in COBIT, it is entitled "category" in CMMI. This hampers the understanding of the IRMs' content.
- IRMs are written on different levels of abstraction. We found reasonable similarities between COBIT control objectives, COBIT control practices, CMMI specific-goals, generic-goals, -practices, sub-practices, SPICE practices and Functional Safety objectives and requirements. Without this mapping information, not all possible similar procedures of the IRMs can be identified.
- IRMs do not always contain all information needed to compare them on a detailed level. To identify the similarity degree of procedures, the different elements of a procedure such as inputs, outputs or roles have to be known. Only some IRMs define outputs of procedures (e.g. in CMMI under typical work products). Other elements such as inputs, roles or activities are not defined. Without suitable identification guidelines we observed that experts identify such elements differently (they map procedure content to different procedure element types).

Therefore, one goal of our approach is to define a common structure for multiple IRMs to model the IRMs' content consistently.

Each IRM Uses a Specific Terminology

- IRMs use different terms to express the same semantic concept. For example, in Functional Safety the term "hazard" and in CMMI the term "risk" is used for the same concept. SPICE uses phrases such as "risk associated with project life cycle" and CMMI uses "project risks". This hampers to understand and to compare IRMs.
- Each IRM uses a specific writing style. While in some of the IRMs verbs in their active form are used, some other IRMs make extensive use of passive, gerunds or nominalizations. This hampers understanding IRMs and identifying different terms that are semantically similar.

Therefore, the second goal is to define guidelines to normalize the terminology (terms and writing style) of IRMs.

As a manual modeling of IRMs cannot always be done consistently the third goal is to develop a tool box supporting the expert to model the IRM's content based on a common structure and terminology.

The remaining of this paper is organized as follows. In the second chapter we will present the MoSaIC approach. Then we introduce our concept to semi-automatically extract fine grained procedure elements according to the MoSaIC approach. Evaluation results, conclusions and a summary conclude this paper in the last two chapters.

2 The MoSaIC Integration Approach

In the following we describe the MoSaIC way to integrate IRMs. Based on our previous work [3] we improved MoSaIC with additional elements based on an in-depth language analysis of IRMs' procedures. First, we give a short overview of the integration approach and describe the MoSaIC meta-models to create a common structure and terminology. Additional guidelines to achieve a common terminology are presented in the last section of this chapter.

2.1 Meta-models

To support organizations in adopting multiple IRMs we have developed MoSaIC, a new model-based approach to integrate different IRMs and to select appropriate improvement concepts. 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) being part of the overall MoSaIC IRM Integration Model. 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.

There are many contributions in the literature to the integration of IRMs and their comparison. In contrast to Ferchichi and Bigand [4] and Liao, Qu and Leung [5] we model each IRM on a more fine granular level. So, we do not only model information such as categories, processes or procedures (such as e.g. in [6], [7], [8] and [9]) but also procedure elements, such as activities, roles, artifacts (outputs or inputs). Guidelines for a uniform description of IRMs [2] also call for the definition of activities and artifacts.

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. Examples are "project plan" or "work breakdown structure".

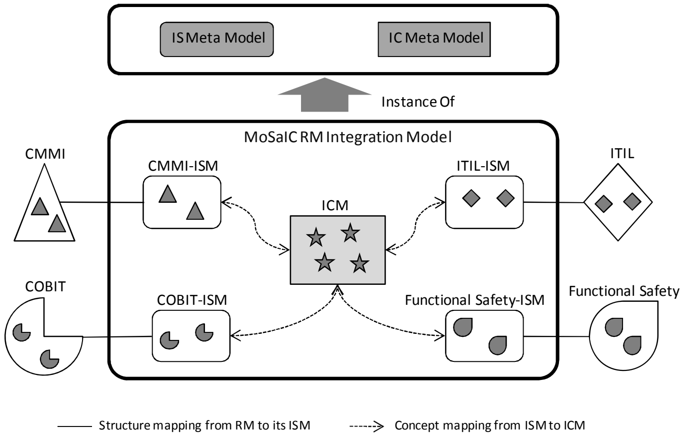


Fig. 1. Model-based integration approach of multiple IRMs

The fine model granularity allows comparing concepts of IRMs in detail. This reveals which concepts of different IRMs are similar and thus, what should an organization address more to adopt these IRMs. Malzahn [6], Soto and Münch [9] also model fine grained concepts but do not give guidelines to identify them. Based on an analysis of the writing styles applied to formulate procedures we added meta-model elements defining the different concept types on the language level (see package Language). This leads to a consistent modeling of concepts.

ICM (the only instance of its IC Meta-Model) is part of MoSaIC's IRM Integration Model as well. It defines all concepts and semantically links all IRM-ISMs by connecting related concepts across the borders of single IRMs. Compared to Malzahn [6] we base the concept comparison not only on "equivalence" but on different similarity relations (e.g. generalization) to get a more accurate degree of similarity. The uniqueness of the ICM concepts, their consistent identification and their traceability back to the original concepts of the IRMs allows to automatically identify the similarities of different IRMs.

Figure 2 shows the most important elements of the IS and IC Meta-Model. The elements of the IS Meta-Model are grouped in three packages.

- Package **Core** contains elements mostly defined by meta-models of existing IRMs. *ReferenceModel* represents an IRM and is structured by means of *Categories*. A category defines a certain topic that is addressed in one or more *ProcessAreas*. A process area addresses a topic to be improved by defining *Procedures*. By means of the *dependsOn* relation dependencies between procedures are modeled.
- Package **Concepts** contains elements to model concept information of IRMs on a fine grained level. We differentiate between *Activities, Artifacts* (*Inputs* and *Outputs*), *Contexts* for activities and *Roles*. These *ProcedureConcepts* are used to model IRM procedures. Each procedure concept from an IS Meta-Model *relatesTo* a concept in the IC Meta-Model.

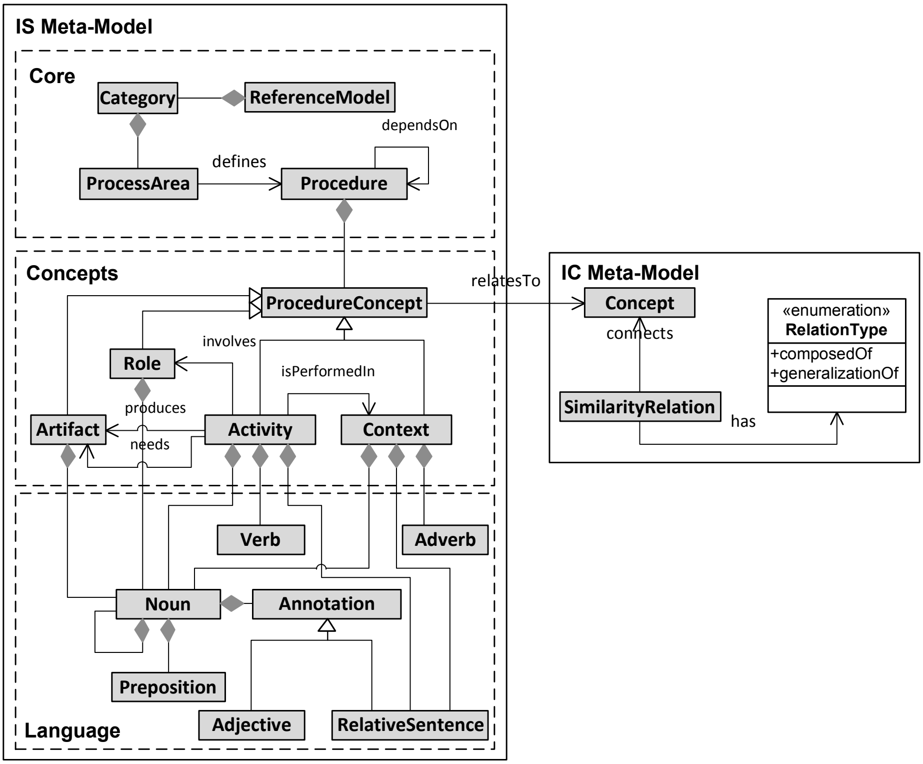


Fig. 2. Integrated Structure and Concept Meta-Model

- Package **Language** contains the syntactical elements used to describe concepts. Artifacts and roles are composed of one or more *Nouns*. Nouns can contain *Prepositions* and build composed Nouns ("records of quality assurance activities"). Nouns can be *annotated* by the use of an *Adjective* ("formal practice") or a *RelativeSentence* ("organisational structure that reflects business needs"). Activities can be composed of one verb and noun ("Create a supplier agreement") or by a verb and a relative sentence ("verify that personnel have the competencies"). *Contexts* can be composed of an *Adverb* ("Formally confirm the agreement") or a relative sentence ("Use effective methods to package the assembled product") or a noun introduced by a preposition ("Deliver with confirmation").

Additional rules developed to transform parts of a sentence to a concept according to the IS Meta-Model and examples support experts in consistently modeling concepts [10]. The IC Meta-Model contains only concepts (activities, roles, inputs, outputs or contexts) and their semantic relations (*generalizationOf*, *composedOf*). For example, "stakeholder" is a generalization of the concept "project manager" and "software requirement" is composed of "functional requirement". The concepts should be unique regarding their semantic interpretation thus trying to reach a normalization of the terminologies used in different IRMs. Therefore, the ICM contains the closure of all semantically different concepts appearing in the IRMs.

2.2 Normalization of Writing Style

According to their personality, educational and cultural background the authors of IRMs tend to express the same ideas differently. For example, while in COBIT and Functional Safety procedures are abundant in passive sentences in CMMI procedures are written almost completely using the active form. As we aim to normalize the language of IRMs and to propose a predefined syntactical structure for the modeled elements, we analyzed some recommendations from Requirements Engineering to write clear, consistent, complete and unambiguous requirements [11], [12].

We postulate that the active form verbs should always be used instead of passive voices, modal and present continuous tenses (as recommended in literature for scientific writing[1]) and instead of nominalizations and gerunds the corresponding verb in active form should always be used.

A normalized writing style allows to identify similar concepts and to build an ICM enabling to automatically compare procedures.

3 Automated Extraction of IRM Concepts

In this chapter we present our solution to support the extraction of concepts. At first we define rules to transform a procedure to its basic concepts. Afterwards we introduce Natural Language Processing (NLP) [13] tasks to apply those rules. Finally, we present a tool chain to create an integrated model of multiple IRMs according to the MoSaIC approach.

3.1 Transformation Rules

Our approach to automatically extract concepts takes a procedure as input and has two steps: In the first *normalization step* the procedure is transformed to a normalized writing style. In the second *extraction step* the concepts are extracted (see Fig. 3).

For both steps we define a set of rules transforming a procedure into concepts. The normalization rules specify how to transform the original IRMs' procedures into the normalized writing style. The extraction rules specify how to identify the normalized concepts according to the IS Meta-Model.

[1] Literature for Scientific Writing
http://abacus.bates.edu/~ganderso/biology/resources/writing/HTWgeneral.html,
http://www.columbia.edu/cu/biology/ug/research/paper.html,
http://maic.jmu.edu/journal/index/General_Rules_for_Scientific_Writing.pdf,
http://www.ugr.es/~agcasco/tierra/Docs/kowalski_scientific_writing.pdf,
http://faculty.uca.edu/march/bio1/sciwriting/writtips.htm.

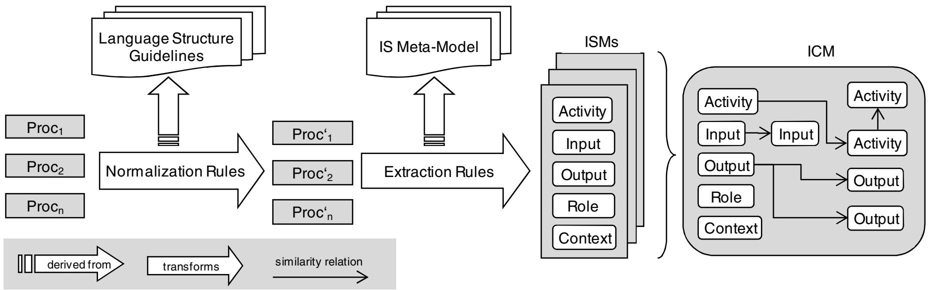


Fig. 3. Steps of normalization and concept extraction

Normalization and extraction rules transform and extract the concepts of procedures. In the definition of the rules the sign "→"means that the left side of the rule is transformed to the right side. The left and right side are specified using a variation of EBNF. But, we use the sign "+" instead of "," because in linguistics "+" is often used to concatenate grammatical structures. A representative example is the Activity Rule:

{Adverb} + Verb + Noun1 + {RelativeSentence} + {Preposition + Noun2} → Verb + Noun1

It defines that if a sentence contains a verb followed by a noun and other verb related grammatical elements (such as adverbs, relative sentences, nouns introduced by prepositions) then an activity composed of the verb and its noun is extracted. The complete set of rules with explaining examples can be found in [10].

To demonstrate the application of the rules we consider as input the CMMI procedure "Training for individuals to perform their roles effectively is provided". First, the normalization rule (VerbPassive | VerbModal | VerbPresent-Continuous → VerbActiv) transforms the passive form of the verb in its corresponding active form "Provide training for individuals to perform their roles effectively". Secondly, concepts are identified using the following extraction rules:

- Activity rule ({Adverb} + Verb + Noun1 + {RelativeSentence} + {Preposition + Noun2} → Verb + Noun1) identifies the activity "provide training for individuals".
- Output rule (Verb + Noun → Noun) identifies the output "training for individuals".
- Role rule (Verb + Preposition + Noun → Noun) identifies the role "individuals".
- Context rule: (Verb + Noun + Relative Sentence → Relative Sentence) identifies the context "to perform their roles effectively".

3.2 Natural Language Processing Support

There are some well-known NLP tasks that can be used to normalize and extract concepts. *POS Tagging* and *Lemmatization* support the normalizations rules. POS tagging chunks a sentence in part of speech (POS) tokens that are "a category to which a word is assigned in accordance with its syntactic functions"[2]. In our case, the POS tagger

[2] "Part Of Speech", Oxford Dictionaries http://oxforddictionaries.com/definition/part+of+speech?region=us

identifies verbs (e.g. "determine"), gerunds (e.g. "managing"). Lemmatization reduces a word to its canonical form allowing identifying the corresponding active form.

For the normalization of the writing style we use different NLP tasks to identify and extract concepts. As these transform a sentence in certain chunks, we call them *Phrase Chunking* tasks. *POS Tagging* identifies the grammatical structures of a sentence. Then *Noun Phrase Chunking* is applied to indentify noun phrases that are a combination of more nouns having a standalone semantic meaning (e.g. "work breakdown structure"). Furthermore, *Named Entity Recognition* determines all entities belonging to certain predefined categories. Such categories could be "persons", "organizations", "locations" and "roles". This supports the identification of role concepts.

3.3 Tool Support

In this section we describe a tool chain to manually and semi-automatically model the IRMs' content according to the MoSaIC approach. The tool chain takes as input a procedure and visualizes the normalized concepts in a tree (Fig. 4). Activities of procedures are presented at the root level. Attached inputs, outputs, contexts and roles are represented as children of their corresponding activities. The tree elements are editable; their type (input, output, etc) can be changed and they can be dragged and dropped to different positions in the tree. After validation and eventual modification of the proposed concepts, the user can store the elements in the IC and IS Model.

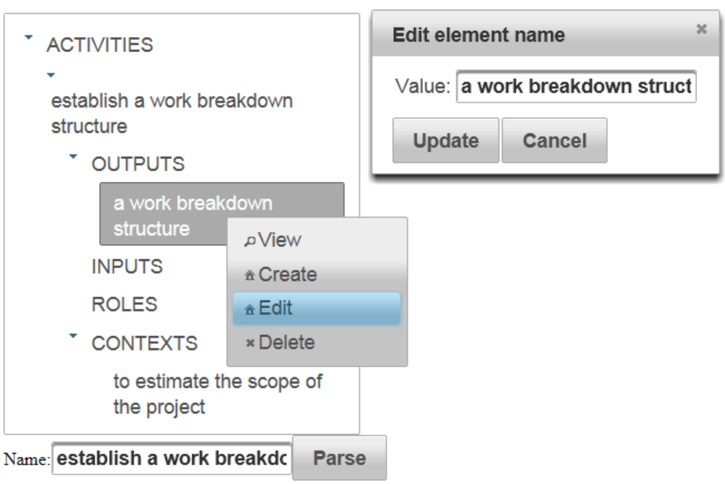


Fig. 4. Screen shot from the MoSaIC concept extraction tool

Our tool chain, implementing a pipe-and-filter architecture, re-uses existing tools to perform the required NLP tasks: GATE (General Architecture for Software Engineering) [14] offers a package of tools to implement NLP tasks. Furthermore, it allows developing so called JAPE-transducers to implement the rules mentioned in section 3.1. The Dragon Toolkit [15] and RiTa.Wordnet [16] are further tools to implement the NLP task of lemmatization. Our tool chain consists of three main components (see Fig. 5).

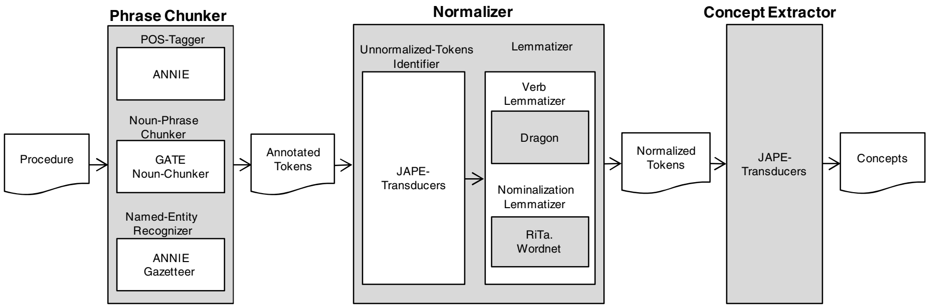


Fig. 5. Architecture of MoSaIC's NLP tool chain

The *Phrase Chunker* offers tools to split a given procedure in *Annotated Tokens*. The *POS-Tagger* (using the GATE plug-in ANNIE – based on a default lexicon and rule set resulting from the training on a large corpus taken from the Wall Street Journal) identifies tokens (such as verbs, nouns, adjectives, adverbs) and annotates them correspondingly. The *Noun-Phrase Chunker* (using GATE Noun-Chunker – an implementation of the Ramshaw and Marcus base noun phrase chunker [17]) annotates tokens as composed nouns. The *Named-Entity Recognizer* (using ANNIE Gazetter – based on a plain text file containing a list of categories and its elements) annotates tokens as category nouns.

The *Normalizer* transforms the *Annotated Tokens* that do not conform to our proposed writing style in *Normalized Tokens*. It contains two components: the *Unnormalized-Tokens Identifier* utilizes JAPE-Transducers to identify the tokens not conforming to our writing guidelines (see section 2.2). For example, in the procedure "a top-level work breakdown structure should be established" a transducer uses as input the verb tokens ("should", "be", "established") and marks these as passive activities. Another example is the recognition of nominalizations based on word suffixes, such as "ent" (e.g. deployment) or "ion" (e.g. categorization).The *Lemmatizer* (Dragon Toolkit and RiTA.Wordnet) uses as input the marks of the transducers and produces *Normalized Tokens*. Dragon Toolkit uses an English lemmatizer to extract the infinitive form of the verb (perfect continuous-verbs, gerunds and passive). Dragon also offers a Porter Stemmer. As stemmers usually produce less reliable results than lemmatizers, we preferred to use the second variant. RiTA.Wordnet uses relations (so called Cross-POS-relations) between concepts in Wordnet to identify the corresponding verb given a certain noun (nominalization).

The *Concept Extractor* uses the JAPE-Transducers to transform the normalized tokens in *IRM Concepts*. In our example, the transducer takes the noun token ("work breakdown structure"), the adjective ("top-level") and the active form of the passive activity ("establish") and mark these together as a concept activity.

4 Evaluation

In the following we present first evaluation results of applying the proposed concept extraction approach to procedures defined by CMMI, COBIT and SPICE. The evaluation was organized as follows.

- First, we manually determined procedures written in different writing styles (verbs in passive, perfect continuous or modal form; gerunds; nominalizations) and also containing different concept types (activities, inputs, outputs, roles, contexts).
- Secondly, we applied our tool chain to process such procedures (14 CMMI, 18 COBIT and 15 FS procedures) and extract the IRM concepts.
- Finally, we validated the obtained results with experts.

Figure 6 summarizes the results.

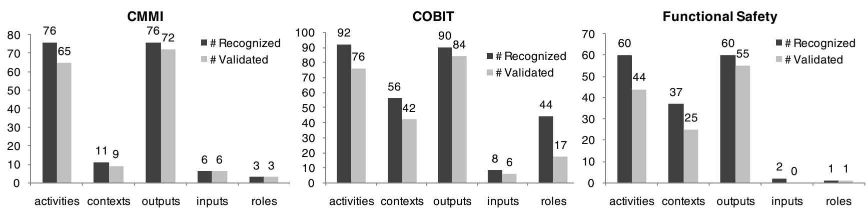


Fig. 6. Comparison of extracted and validated concepts

Obviously there were some failures in the transformation of IRM procedures to IRM concepts. After a thorough analysis we identified the following failure categories.

- **Normalization of writing styles.** Gerunds and nominalizations could not be identified always properly. Not all gerunds should be transformed to their corresponding active form (e.g. "The process is institutionalized as an optimizing process"; "optimizing" is an adjective and can be left unnormalized). Furthermore, not all nouns with the mentioned suffixes (see section 3.2) are nominalizations (e.g. "information"). Therefore, we decided only to provide concept candidates that can be corrected and modified by the user.
- **Complex syntactical structure of IRM's procedures.** Combinations of verbs and nouns connected by junctions ("and", "or") lead to an incomplete identification of activities and outputs (e.g. "Coordinate the activities and evaluation of multiple projects"). The first activity is not completely identified ("coordinate the activities" instead of "coordinate the activities of multiple projects"). This failure can be corrected by considering also junctions in the JAPE-Transducers.
- **Semantic based extraction**. The identification of inputs depends sometimes on the semantics of the verb (in "communicate policy", the concept "policy" is input and output; in "eliminate risks", the concept "risks" is only input). This failure category could be corrected by creating a dedicated verb database and by treating them separately. Furthermore, some prepositions introduce a context, an artifact or part of an artifact. It depends on the semantics of the sentence to identify the correct concept type. In "consider a mechanism for inclusion in the contract agreement" the preposition "for" introduces a context "for inclusion in the contract agreement" while in "consider a mechanism for monitoring the capability of supplier", the preposition "for" introduces a part of the output "mechanism for monitoring the capability of supplier". These cases cannot be corrected automatically and need manual interaction.

- **Not recognized grammatical tokens**. ANNIE does not recognize all verbs when performing the POS tagging. An improvement of ANNIE or its replacement by another POS tagger might alleviate this failure.
- **Small database of prepositions.** The identification of outputs, roles and inputs often depends on the prepositions that introduce the nouns. Inputs were identified as being outputs or parts of outputs/inputs were identified as roles. Our database with prepositions has to be enlarged.

Generally we observed that the quality of the proposed concept candidates is acceptable. The evaluation also showed that the syntactical structures in COBIT and Functional Safety are more complex than in CMMI.

5 Future Work and Conclusions

In this paper we presented a tool based approach to integrate multiple IRMs based on IRM concepts that can be extracted semi-automatically from IRM procedures. Based on NLP tasks, the presented tool chain allows a semi-automated modeling of IRMs concepts. Our approach supports the connection of similar IRM concepts by predefined relations and thus allows the identification of similar procedures that are composed of related concepts.

The language used in IRMs is sometimes too complex to be interpreted by machines and our tool chain does not always extract all the concepts. Some improvements can be done easily by improving the transducers that implement the transformation rules. Another improvement would be to enlarge our database by learning from the modeling decisions of the experts that are using our tool. However, our approach does not offer a fully automated way to extract concepts and to create an integrated view on different IRMs. But, it substantially supports the expert in this complex modeling task.

In the future, we will cover more areas of the most popular IRMs and we will consider also other IRMs such as SPICE or V-Model.

To summarize, the proposed tool based approach supports the expert to semi-automatically model IRMs according to a certain language structure, thus achieving a normalization of multiple IRMs. This realizes a basis for the integration and comparison of multiple IRMs, thus supporting organizations in an effective and efficient adoption and assessment of multiple IRMs.

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. Pricope, S., Lichter, H.: A Model Based Integration Approach for Reference Models. In: Caivano, D., et al. (eds.) 2nd Proceedings of 12th Int. Conference on Product Focused Software Development and Process Improvement, Torre Canne, Italy, June 20-22, pp. 6–9. ACM, New York (2011)
4. Ferchichi, A., Bigand, M.: An Ontology for Quality Standards Integration in Software Collaborative Projects. In: Proceedings of MDISIS 2008, Model Driven Interoperability for Sustainable Information Systems, Montpellier, France. CEUR Workshop Proceedings, vol. 340, pp. 17–30 (2008)
5. Liao, L., Qu, Y., Leung, H.: A Software Process Ontology and Its Application. In: Proceedings of IWFST 2005, Int. Workshop on Future Software Technology, Shanghai, pp 1–10 (2005)
6. Malzahn, D.: Assessing - Learning - Improving, an Integrated Approach for Self Assessment and Process Improvement Systems. In: Proceedings of ICONS 2009, Fourth International Conference on Systems, Gosier, Guadeloupe, France, pp. 126–130 (2009), doi:10.1109/ICONS.2009.31
7. Siviy, J., Kirwan, P., Marino, L., Morley, J.: Process Architecture in a Multimodel Environment. White Paper, Software Engineering Institute, CarnegieMellon (2008), http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp4_processArch_052008_v1.cfm
8. Wang, Y., King, G., Dorling, A., Wickberg, H.: A unified framework for the software engineering process system standards and models. In: Proceeding ISESS 1999 Proceedings of the 4th IEEE International Symposium and Forum on Software Engineering Standards, p. 132. IEEE Computer Society, Washington, DC (1999)
9. Soto, M., Münch, J.: Using Model Comparison to Maintain Model-to-Standard Compliance. In: Proceedings of the ICSE Workshop Comparison and Versioning of Software Models (CVSM 2008), Leipzig, Germany, May 17, pp. 35–40 (2008)
10. Jeners, S.: MoSaIC Reference Model Modeling Rules, Software Construction Group (2012), http://www.swc.rwth-aachen.de/DOCUMENTS/MOSAIC/EuroSPI/ModelingRules.pdf
11. Grinder, J., Bandler, R.: The Structure of Magic II: A Book about Communication and Change. Science and Behavior Books, Palo Alto (1976)
12. Pohl, K., Rupp, C.: Requirements Engineering Fundamentals – A study Guide for the Certified Professional Requirements Engineering Exam – Foundation Level – IREB compliant. Rocky Nook, Inc., Santa Barbara (2011)
13. Kao, A., Poteet, S.R.: Natural Language Processing and Text Mining. Springer, London (2007)
14. Cunningham, H., Maynard, D., Bontcheva, K., et al.: Text Processing with GATE (Version 6), University of Sheffield Department of Computer Science, April 15 (2011), http://gate.ac.uk/releases/gate-6.1-build3913-ALL/tao.pdf
15. Zhou, X., Zhang, X., Hu, X.: The Dragon Toolkit Developer Guide, http://dragon.ischool.drexel.edu/tutorial.pdf
16. Howe, D.: RiTA.Wordnet, http://www.rednoise.org/rita/wordnet/documentation
17. Ramshaw, L., Marcus, M.: Text Chunking Using Transformation-Based Learning. In: Proceedings of the Third ACL Workshop on Very Large Corpora, pp. 82–94 (1995)

Innovating Innovation: A Conceptual Framework

Thomas Peisl[1] and Juergen Schmied[2]

[1] Munich University of Applied Sciences, Department of Business Administration, Am Stadtpark 20, 81243 Munich, Germany
tpeisl@hm.edu
[2] Method Park Management GmbH, Wetterkreuz 19a, 91058 Erlangen, Germany
juergen.schmied@methodpark.com

Abstract. The authors propose a conceptual framework to extend their previous research on innovation capability determination to innovating innovation. Based on an expert analysis of process assessments the authors conclude that innovation, in particular the holistic understanding of innovation dimensions, has not yet been the focus in organizations. Most of our applied research indicates a clear focus on incremental product and/or process innovation. The authors argue that the interdependency of the proposed 14 innovation dimensions is essential for a future driven approach to innovation assessments based on uncertainty. The findings also include an innovation view on the SPI Manifesto developed at EuroSPI 2010.

Keywords: Innovation Management, Innovation Dimensions, Disruptive Innovation, Incremental Innovation. Innovation Processes, CMMI, ICE, ISO/IEC 15504.

1 Introduction

Leading organizations have implemented process improvement based on CMMI and/or SPICE principles [37]. The focus of improvements, and hence assessments, has been on processes. The people and technology dimensions are not yet central elements in assessing organizational capabilities [21][37]. The inherent structural conflict between evaluating current processes and disruptive change results in fundamental managerial challenges. Process analysis today is centered on judging existing processes based on historical data and usually on selected processes [21][37]. Peisl, Reger & Schmied [30] argue that "a potential solution is using strategic themes to identify a portfolio of strategic innovation initiatives and, based on a dynamic quantitative and qualitative process analysis, creating a separate new class of innovation centered initiatives."

The primary objective and, hence, the research question of this paper, is to investigate to what extent organizations are prepared to innovate innovation. The authors propose to extend the view on continuous innovation based on actual processes and products to disruptive innovation in a multi-dimensional context. Therefore, a framework is applied and tested in order to assess the future readiness of

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 217–228, 2012.

organizations. It compares actual strategic choices of selected organizations with the innovation dimensions proposed, and rates the organizations capability accordingly. We extend our existing research model on innovation capability determination (ICE) including 14 innovation dimensions to disruptive innovation [7][8] in order to create a framework for innovating innovation. Focus is on the interdependency of innovation and disruptive change reflecting the uncertainties organizations are facing today, shifting from process improvement to systems thinking, and ultimately to successful innovation management.

The importance of innovation to the success of a company has been highlighted ever since Schumpeter's formulation of his theory [35] of Creative Destruction where there is a constant search to create something new which simultaneously destroys the old rules and establishes new ones. Sawhney, Wolcott & Arroniz [32; p.76] propose a holistic definition of business innovation as "the creation of substantial new value for customers and the firm by creatively changing one or more dimensions of the business system". The understanding of innovation in the context of this research paper is defined as: "Value innovation requires companies to orient the whole system toward achieving a leap in value for both buyers and themselves." [23; p. 17]. Chesbrough (2003) conceives the importance of innovation in "companies that don't innovate die."

When evaluating current literature, it has been noticed that organizations predominantly pursue a road map approach instead of applying an integrated analysis approach when making strategic decisions [3][25]. While the road map approach examines potential strategic options based on current resource and market attractiveness looking into the future, the future analysis approach determines the future based on megatrends as a given fact and deducts from the future back in order to determine an innovation strategy [20]. Byrne [4; p. 2] defines megatrends as "global, sustained and macroeconomic forces of development that impact business, economy, cultures, careers and personal lives, thereby defining our future world and its increasing pace of change." Horx, [20] argues for a trend in literature that future analysis becomes more and more important for strategic decisions and, hence, is critical for the survival of organizations. The authors propose a comprehensive approach including both views. In particular the innovation dimensions and the interdependencies can only be assessed when including futures and current trends.

This paper includes a reflection on existing research, disruptive innovation and uncertainty, and a case based approach to improvement and process assessments. Based on our continuous research an innovation update to the SPI Manifesto is drafted, and further research is proposed.

2 Existing Research: Methodology and Cases

As underlying research strategy, a case study approach is applied [17][41]. This approach focuses on theory building and allows a combination of multiple levels of analyses and data collection methods. Pursuing a combination of a deductive-inductive approach, qualitative and quantitative data collection techniques are used.

Thus, whilst the development of the assessment model derives from secondary data analysis, primary investigation is applied by conducting semi-structured expert interviews. The selection criteria of the sample are based on a non-probability, purposive sampling strategy. The credibility of our initial findings: reliability, validity, and generalizability were considered. In order to increase reliability and validity of the findings, the interview questions are piloted. Furthermore, an ethical framework has been considered throughout the entire research process [33] (Adams et al., 2007). The goal is to generate a comprehensive theory by analyzing multiple studies on megatrends, business strategies as well as innovation management.

The nature of the research approach applied is exploratory, since the aim is to find out 'what is happening', to seek new insights and thus, to develop a coherent theory [15][33]. Combining induction and deduction as basis theories of the exploratory study, the process of theory building starts with the collection of data without necessarily the formation of an initial framework. Hence, a theory can be developed leading to certain predictions. Our research is predominantly based on data gathered from interviews and experience, and hence may involve subjective meanings entailing that research is value bound (Bryman, Bell, 2007). The adopted research philosophy influences the results of the study and therefore needs to be considered very carefully [33].

2.1 Innovation Capability dEtermination (ICE)

The result of our research on applying the concept of ISO/IEC 15504 to innovation processes was first published in 2007. The authors defined the concept of capability as the degree to which a process is performed, managed, established, predicted, and continuously optimized [21]. The generic innovation process from idea generation to innovation diffusion (IIP), i.e. the successful – and profitable – positioning of new products and services in the market, includes five steps: Idea generation, concept evaluation, concept implementation, innovation piloting, and innovation diffusion [11][13]. Other authors propose a simple map involving searching, selecting, implementing, and capturing value [39; p. 55].

The analysis of organizational objectives provides the necessary precondition for the innovation process. The framework is completed with process improvement measures across all steps as well as a systems controlling. Tidd, Bessant [39; p. 21] developed a 4Ps model in order to provide organizations with a map identifying where organizations can innovate. The four categories of the innovation space include "product innovation (changes in the products/services that an organization offers), process innovation (changes in the ways in which they are created and delivered), position innovation (changes in the context in which products/services are introduced, and paradigm innovation (changes in the underlying mental models which frame what the organization does)." Sawhney, Wolcott & Arroniz [32] propose 12 dimensions for companies to innovate.

The 14 connected innovation dimensions (CIDs) proposed by the authors [28] are structured according to the perspectives of the balanced scorecard [24]. The customer perspective includes product/service, solution, customer and market, brand and marketing, value capture, and customer experience innovation. The business process perspective includes process, value chain, distribution, and business design innovation. Learning and growth involves platform, networking, and human resource innovation, and the financial perspective adds balance sheet innovation to the conceptual framework discussed in this paper. Following the concept of open innovation [6] the authors included spin-off innovation and new venture management as additional innovation dimensions to the innovation capabilities set in 2008. Amit, Shoemaker [2; p. 35] define capabilities as: They "[...] refer to a firm's capacity to deploy resources usually in combination, using organizational processes, to effect a desired end." When, at any stage of the innovation process, the organizational objectives or internal competencies fail to meet the resource requirements of the innovation, either a spin-off or a new venture with partners needs to be considered. The other alternative is to cancel the project, thereby actually destroying value for the organization.

The substantial contribution confirmed by industry partners in our applied research is the holistic integration of the innovation process and process maturity as the core element, and the innovation dimensions as the funnel identifying the potential to innovate. The distance from the core to the individual dimension reflects the process maturity and/or the disruptive potential in a given area [39].

2.2 Disruptive Innovation and Uncertainty

The probability and frequency of disruptions as well as the uncertainties in the way we do business is continuously increasing. Abernathy [1; p. 59] differentiated incremental from radical innovation: "Incremental innovations give impetus to and further shape the direction of existing design approaches, functioning as steps in an underlying long-term trend. Radical innovations introduce entirely new approaches."

Tushman & Anderson [40; p. 441] provide a distinction between continuous and discontinuous innovation: "Technological change is a bit-by-bit, cumulative process until it is punctuated by a major advance. Such discontinuities offer sharp price-performance improvements over existing technologies. Major technological innovations represent technical advance so significant that no increase in scale, efficiency, or design can make older technologies competitive with the new technology." In addition they distinguish between competence-destroying and competence-enhancing technological discontinuities: "Competence-destroying discontinuities are so fundamentally different from previously dominant technologies that the skills and knowledge base required to operate the core technology shift. Such major changes in skills, distinctive competence, and production processes are associated with major changes in the distribution of power and control within firms and industries." [40; ; p. 441] According to their research this is the main reason why incumbents fail when new technologies emerge. In contrast: "Competence-enhancing discontinuities are order-of-magnitude improvements in price/performance that build on existing know-how within a product class. Such innovations substitute for older technologies, yet do not render obsolete skills required to master the old technologies." [40; p 442]

Tidd & Bessant [39] support this view and extend it by introducing several factors that can lead to a discontinuous innovation. “Or it can come through the emergence of a completely new market with new characteristics and expectations” and “discontinuity can also come about by reframing the way we think about and industry- changing the dominant business model and hence the ‘rules of the game” [39; p. 30]. They distinguish between incremental (“doing what we do but better)” and radical innovation (“doing something very different”). The more towards outside the circle the more radical the innovation will be, the more to the center, the more incremental.

“Most new technologies foster improved product performance. I call these sustaining technologies. Some sustaining technologies can be discontinuous or radical in character, while others are of an incremental character. What all sustaining technologies have in common is that they improve the performance of established products along the dimensions of performance that mainstream customers in major markets have historically valued” [7; p. XVIII]. Disruptive innovations are: “…innovations that result in worse product performance, at least in the near term… disruptive innovations bring to a market a very different value proposition than had been available previously” [7; p. XVIII]. Products made of disruptive technologies usually underperform existing products in mainstream markets, but due to the different value proposition. Entering a different market or a lower market segment, disruptive technologies can, when subsequently improved through sustaining innovations, enter the mainstream market eventually and meet the performance demands. Products of existing players then are sometimes “too high tech, massively over engineered and, most important, much too expensive.” [7; p. XX] The second reason for problems of established firms when confronted with disruptive change is their value network, cost structure, which is focused on high margins in the high-end market and does not allow them to focus on small markets with lower margins. ”Hence, most companies with a practiced discipline of listening to their best customers and identifying new products that promise greater profitability and growth are rarely able to build a case for investing in disruptive technologies until it is too late.” [7; p. XX][3]

The critical question is, hence, how organizations faced with increasing uncertainty and disruptions in the way of conducting business cope with innovation: Moving from a product and process view to an integrated understanding of innovation dimensions and open business models, including new measurements [5][6][8][30].

3 Case Based Approach to Improvement and Process Assessment

Process assessments based upon maturity models respectively process assessment models like ISO 15504, CMMI (or former versions like SW-CMM and BOOTSTRAP) were conducted for decades. Nevertheless a profound analysis on a positive correlation between process maturity and innovation capability, and in particular disruptive innovation, has not been proposed yet. Hence **H1**: There is no positive correlation between maturity and innovation capability (disruptive view). Certainly we need to investigate the impact of maturity on incremental innovation as well. A sub-hypothesis to H1 is H1-1: Organizations at maturity level 5 are not necessarily recognized as innovation leaders in their respective industries. On the contrary H1-2 is formulated. H1-2: Industry changers are not necessarily assessed by the SEI or related institutions. Ultimately H1-3: High ML-assessed organizations (maturity level 4-5) have the same/or higher exposure to disruptive technological changes than other organizations. **H2**: There is a positive correlation between maturity and process improvements but limited to an incremental view. If we relate process improvement to ISO 15504, success stories were only published by various single organizations, but not by research institutes in a more generalized approach. There are more success stories in process improvement based on CMMI (and the former model SW-CMM) because these models were published and used already ten years earlier (this is also the reason why we use CMMI related success stories even we proposed our ICE model based upon ISO 15504-2). Nevertheless these data support H2 but are mainly limited to the SW or systems developing industry. In addition to the process dimension only limited analysis has been done on the interdependency of innovation dimensions. Therefore the authors propose **H3**: Assessments today focus only on processes and, hence, affect future organizational capabilities with respect to the innovation capability as discussed above. One of the most detailed papers on success stories in process improvement was published in 2006: “Performance Results of CMMI®-Based Process Improvement” (Technical Report CMU/SEI-2006-TR-004) supporting H2 and H3 initially. More general results are provided in the CMMI Appraisal Results, published twice a year by the SEI. These reports include community trends of CMMI-based appraisals and appraisal results. An analysis of the case studies on organizations at maturity level 5 shows, that they typically only report major improvements in process innovation, partly supporting H1-2, H2, and H3. This can be attributed to the built-in concept of CMMI to focus on processes, other dimensions of innovation as proposed by the authors are not considered. The same is valid for ISO 15504-5 (or other SPICE variants) related process improvement projects.

In principle CMMI is including incremental improvements and innovations in the process area Organizational Performance Management [36]. Even if we extend the term innovation as it is used to include disruptive innovation, the performance results published within the technical report in 2006 show that these companies typically do not focus on disruptive innovations. Hence, H1 and H3 are supported. Low Maturity companies typically struggle with establishing and deploying standard processes, whereas High Maturity companies are typically focusing on stabilizing and optimizing processes. This is also supported by our own assessment and process improvement experience. Innovation in R&D organizations is far too often driven by levels and incremental in order to achieve compliance but not to innovate innovation. **H4**: Assessments are caused by customer and market compliance requirements and not necessarily by future capabilities. In this context we try to identify the cause-effect of assessments and not to question the validity of organizational objectives as defined in CMMI or SPICE for process improvements.

In a recent study using the case research approach discussed above on innovation capability based on the proposed ISO 15504 construct at SMEs in Germany we tested our propositions with ten organizations, predominately in automation, IT services and medical systems. The starting point, in addition to the usual filtering questions on the organization, has been on the number of innovation initiatives and the resources allocated to the innovation process to get a subjective response on the importance of innovating within the company. Only two organizations felt that the number of innovation initiatives is unsatisfactory, but seven interviewees indicated that the resources are not sufficient to match the requirements. Research shows that organizations do not lack ideas to implement new products or services but structured ways to allocate resources on the right innovation initiative [16][39].

The main part of our research is related to the innovation dimensions. In step 1 we defined the 14 dimensions. Then we classified the interview responses according to the level of awareness and importance to the organization using a competence model [26]. For this, the rating scale (N/P/L/F) defined in ISO/IEC 15504-2 was used, but we have not explicitly used the defined assessment process (either SPICE or CMMI related). Innovating Innovation includes a systems thinking in multiple innovation dimensions that are used as a filter to identify and/or prioritize management decisions. The selection of the relevant dimensions for an organization depends on the innovation objectives derived from the strategic objectives and is different for each single organization. The dimensions proposed by the authors are described in terms of the process ID, name, and purpose statement. An illustrative process outcome will be described for the pilot assessments. By using the dimensions as a filter previous to the throughout the innovation process the idea generation is focused on the selected innovation dimensions that are based on the organizational objectives and strategy. Within a brainstorming session the strategic business team including managers, employees, and others, e.g. present and potential customers, can generate ideas in the selected innovation dimensions. If ideas do not fit into the current innovation objectives the organization should consider a new venture or follow a spin-off strategy.

The customer perspective includes product/service, customer and market, brand and marketing, customer experience, value capture, and solution innovation.

Process ID	CID.1
Process name	Product or Service Innovation Process
Process ID	CID.2
Process name	Customer and Market Innovation Process
Process ID	CID.3
Process name	Brand and Marketing Innovation Process
Process ID	CID.4
Process name	Customer Experience Innovation Process
Process ID	CID.5
Process name	Value Capture Innovation Process
Process ID	CID.6
Process name	Solution Innovation Process

The financial perspective adds balance sheet innovation:

Process ID	CID.7
Process name	Balance Sheet Innovation Process

The business process perspective includes value chain, process, distribution, and business design innovation.

Process ID	CID.8
Process name	Value Chain Innovation Process
Process ID	CID.9
Process name	Process Innovation Process
Process ID	CID.10
Process name	Distribution Innovation Process
Process ID	CID.11
Process name	Business Design Innovation Process

Learning and growth involves platform, networking, and human resource innovation.

Process ID	CID.12
Process name	Platform Innovation Process
Process ID	CID.13
Process name	Networking Innovation Process
Process ID	CID.14
Process name	Human Resource Innovation Process

In order to test our research approach we have followed a five-step construct. 1. To validate the fourteen innovation dimensions, 2. To verify the interdependency of the innovation dimensions depending on organizational objectives, 3. To add a maturity score to identify process capability, 4. To extend our model to disruptive innovation, and 5. To add tools, methodologies, and practices to design an integrated assessment model for innovating innovation.

In our continuous research we will test stage 4. Our new propositions include P1: Maturity level 1-3 reflect an organizations capability to manage incremental innovation, and P2: Maturity levels 4 and 5 need to be extended to disruptive innovation in order to build in agility, adaptability, and uncertainty. It may be essential to add that any organization needs the capability to continuously improve existing processes, and we argue for an extended view on future capabilities added to level 4 and 5, i.e. to add chaos to the strategic thinking [35]. Again the distance from the innovation process as the core to the individual dimension reflects the process maturity and the disruptive potential in a given area [39]. Current research from Kim, Mauborgne [23] can be added as level 1-3 might be considered Red Ocean, whereas level 4-5 are Blue Ocean dimensions.

4 Innovation Update to the SPI Manifesto

The current version of the SPI Manifesto includes innovation in principles 3 and 8. We propose to adapt the principles considering the following thoughts on innovating innovation. Principle 3: “Base improvement on experience and measurements” should be updated. Certainly we agree with this concept but it is again mainly related to incremental innovation: Incremental gathering of experience and incremental collection of measures and improving these measures stepwise. We argue that disruptive innovation can also imply to create a new product with even worse product performance. Typically we do not have experience with a product targeting a new market with a new set of performance characteristics, i.e. a disruptive innovation. The reason why this product is an innovation is, that by adding other values instead of performance, like functionalities not provided by any other product, it addresses a new market. As a consequence improvement is not only based upon measures and experience. The challenge is to validate the innovation because we do not have experience / measures from the market. Nevertheless the later proof of this must be done by measurements but this does not automatically mean that for proofing the added value the same measures are used than before introducing the innovation. Typically the measurement itself is affected by the disruptive innovation. Actually the authors propose that for any disruptive innovation an organization needs to develop a new metric system, otherwise change will not sustain and competitors will gain momentum. We propose a new title: “Achieve verifiable improvement through generating added value”.

As a consequence we propose to re-formulate the explanation of principle 3. As processes are what people do, any innovation effort must optimize their ‘doing’ and bring added value to their day-to-day business. The conditions for innovations can be set, but only the individual can change actions. Thus you need individual competences, readiness, and willingness to learn and innovate. ‘Readiness’ means now in addition the readiness for radical change. A radical change could even mean to destroy existing structures. But this should not be misunderstood in the sense of generating uncontrolled risks. Instead, a radical change shall be implemented based upon a strategic procedure. The aspect ‘competence' must be updated: Here we have to add the competence of “systems thinking”. It is not enough to think only in one dimension, regardless which dimension is chosen. The individual who wants to innovate has to have the ability to consider all of the 14 mentioned dimensions. Of course not all dimensions are relevant for the specific business. But it is necessary to understand these dimensions and to filter these dimensions in the right way based upon the own business goals. Within the aspect ‘willingness’ we do not see the need for change. But we propose an additional example to highlight the idea of a disruptive innovation. Example 4: Up to now a software developing company (SDC) was getting paid only per software license sold. Now the new business concept is based upon application service providing. As a consequence SDC is offering new services around application hosting and is now getting paid time-based in addition to license based. Value to SDC's customers was added in the sense of additional services and alternative payment methods. Value to SDC was added in the sense of additional business and revenue stream.

With respect to principle 8 we propose to add the spin-off and new venture management dimension as organizational innovation challenges. Hence, change is not limited to adapt the organization as it is to disruption and uncertainty, but requires an inside-out view as well as an outside-in perspective. Objectives and measures need to be altered accordingly. Our last example based upon application service providing can serve as a good example for principle 8 too. After SDC started and established the new business concept around application service providing SDC's management decided to separate different business in different companies to communicate more clearly to their customers who is responsible for what type of business. The result is that SDC splits product development together with sales/ distribution of license-based business from the service oriented business model. This leads to founding a new company plus to a reorganization of the existing company SDC.

5 Summary and Outlook

In this research paper the authors propose an integrated view on innovation based on 14 connected innovation dimensions, process capability and maturity, incremental and disruptive innovation, and the interdependency to manage future uncertainty. The case based approach provides the context for our research on how to initially test and assess innovation initiatives in an open innovation context. The need to innovation and continuously create value to customers and stakeholders is based on the understanding that processes and process measurements shape culture and innovative behavior in all organizational dimensions. We integrate practices and methodologies, and design a framework to innovate innovation, but we still need further research and verification across industries. This paper is a work in process documentation and we would like to invite interested organizations to join in our applied research project.The next steps include a Danish case study with two innovation leaders, an extension to the EU Innovation Manager program, and a proposal to include a innovation process assessment as an extension to the ISO portfolio.

References

1. Abernathy, W.J.: The Productivity Dilemma: Roadblock to Innovation in the Automobile Industry. The John Hopkins Press, London (1978)
2. Amit, R., Shoemaker, P.: Strategic Assets and Organizational Rent. Strategic Management Journal 14, S33–S46 (1993)

3. Hamilton, B.A.: The Customer Connection: The Global Innovation 1,000 (2007), http://www.boozallen.com/news/2007Innovation1000
4. Byrne, M.: Frost & Sullivan identifies top global mega trends (2011), http://www.smeadvisor.com/2011/04/frost-sullivan-identifies-topglobal-mega-trends/ (accessed: March 16, 2012)
5. Chesbrough, H.: Open Business Models: How to Thrive in the New Innovation Landscape. Harvard Business School Press, Boston (2006)
6. Chesbrough, H., Vanhaverbeke, W., West, J.: Open Innovation – Researching a New Paradigm. Oxford University Press, New York (2006)
7. Christensen, C.M.: The Innovator's Dilemma. Harvard Business School Press, Boston (1997)
8. Christensen, C.M., Raynor, M.M.: The Innovator's Solution – Creating and Sustaining Successful Growth. Harvard Business School Press, Boston (2003)
9. Cooper, R.G., Edgett, S.J.: Lean, Rapid and Profitable: New Product Development. Product Development Institute, Ancaster (2005)
10. Cooper, R.G., Edgett, S.J.: Maximizing Productivity in Product Innovation. Research Technology Management 51(2), 47–58 (2008)
11. Cooper, R.G., Edgett, S.J.: Overcoming the Crunch in Resources for New Product Development. Research Technology Management 46(3), 48–58 (2003)
12. Cooper, R.G., Edgett, S.J., Kleinschmidt, E.J.: New Problems, New Solutions: Making Portfolio Management More Effective. Research Technology Management 43(2), 18–33 (2000)
13. Cooper, R.G., Edgett, S.J.: Developing a Product Innovation and Technology Strategy for Your Business. Research Technology Management 53(3), 33–40 (2010)
14. Cooper, R.G., Edgett, S.J.: Maximizing Productivity in Product Innovation. Research Technology Management 51(2), 47–58 (2008)
15. Cooper, D.R., Schindler, P.S.: Business Research Methods, 9th edn. McGraw-Hill, London (2006)
16. Dooley, Subra, Anderson: Maturity and its impact on new product development project performance. Research in Engineering Design 13(1) (August 2001)
17. Eisenhardt, K.M.: Building Theories from Case Study Research. Academy of Management Review 14(4), 532–550 (1989)
18. Freeman, C., Clark, J., Soete, L.: Unemployment and Technical Innovation: A Study of Long Waves and Economic Development. Frances Printer, London (1982)
19. Hopfenbeck, W., Peisl, T., Müller, M.: Wissensbasiertes Management, Managementkonzepte in der Internet Ökonomie. MI Verlag (2001)
20. Horx, M.: Theorie der Trend- und Zukunftsforschung (2010), http://www.horx.com/Zukunftsforschung/2-01.aspx#Hdr (accessed: August 15, 2011)
21. ISO/IEC 15504-2:2003: Information technology – Process assessment – Part 2: Performing an Assessment
22. ISO/IEC TR 15504-7:2008: Information technology – Process assessment – Part 7: Assessment of Organizational Maturity
23. Kim, Mauborgne: Blue Ocean Strategy. How to create uncontested market space and make the competition irrelevant. Harvard Business School Press, Boston (2005)
24. Kaplan, Norton: The Balanced Scorecard. Translating Strategy into Action. Harvard Business School Press, Boston (1996)
25. Novitski, B.J.: Understanding megatrends helps firms plan for the future. Architectural Record 197(8) (2009)

26. O'Connor, J., Seymour, J.: Introducing NLP. HarperCollins, London (2002)
27. Peisl, T.: Barrieren in Veränderungsprozessen. Ein Erklärungsmodell für das Scheitern von Veränderungsprozessen in Mittel- und Großunternehmen. Dissertationsschrift (1995)
28. Peisl, T., Schmied, J.: Connected Innovation: Innovation Capability dEtermination (ICE), International SPICE Days. Frankfurt/Main (2007)
29. Peisl, T., Schmied, J.: Improvement through Innovation. In: International SPICE Days, Prag (2008)
30. Peisl, T., Reger, V., Schmied, J.: Innovation Process Design: A Change Management and Innovation Dimension Perspective. In: O'Connor, R.V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. CCIS, vol. 42, pp. 117–127. Springer, Heidelberg (2009)
31. Porter, M.E.: Competitive Advantage: creating and sustaining superior performance. Free Press, New York (1985)
32. Sawhney, Wolcott, Arroniz: The 12 Different Ways for Companies to Innovate. MIT Sloan Management Review 47(3), 75–81 (2006)
33. Saunders, M., Lewis, P., Thornhill, A.: Research Methods of Business Students, 5th edn. Pearson Education Limited, Harlow (2009)
34. Schmied, Wentzel, Gerdom, Hehn: Mit CMMI Prozesse verbessern. dpunkt (2008)
35. Schumpeter, J.: Capitalism, socialism and democracy, 4th edn. Allen & Unwin Ltd., London, UK Edit. 1942 (1935)
36. Software Engineering Institute (SEI): CMMI for Development: Guidelines for Process Integration and Product Improvement (SEI Series in Software Engineering), 3rd edn. Addison-Wesley Longman, Amsterdam (2011)
37. Software Engineering Institute (SEI): CMMI® for SCAMPI(SM) Class A Appraisal Results 2011 End-Year Update; Carnegie Mellon University, Pittsburgh, `http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2012MarCMMI.pdf`
38. Software Engineering Institute (SEI): Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document; CMU/SEI-2011-HB-001, `http://www.sei.cmu.edu/reports/11hb001.pdf`
39. Tidd, J., Bessant, J.: Managing Innovation: Integrating Technological, Market and Organizational Change, 4th edn. John Wiley & Sons, West Sussex (2009)
40. Tushman, M.L., Anderson, P.: Technological Discontinuities and Organizational Environments. Administrative Science Quarterly 31(3), 439–465 (1986)
41. Yin, R.K.: Case Study Research: Design and Method, 3rd edn. Sage, London (2003)

Towards an Ideation Process Applied to the Automotive Supplier Industry

Martin Neumann[1], Andreas Riel[2], Serhan Ili[3], and Daniel Brissaud[4]

[1] KSPG AG, 41460 Neuss, Germany
martin.neumann@de.kspg.com
[2] EMIRAcle AISBL c/o Grenoble INP, G-SCOP UMR5272, 38031 Grenoble, France
andreas.riel@grenoble-inp.fr
[3] ILI CONSULTING, Karlsruhe, Germany
ili@ili-consulting.de
[4] Grenoble INP, G-SCOP UMR5272, 38031 Grenoble, France
daniel.brissaud@grenoble-inp.fr

Abstract. Under the continuously rising innovation pressure, companies from many industry sectors find themselves confronted with the need to introduce or reinforce systematic innovation management methods and processes. It is widely recognised that ideas coming from both within and outside the organisation can be considered as the very source of innovations. Nevertheless, ideation - the generation and management of ideas - happens in the so-called "fuzzy front-end" of the otherwise well-structured process landscape of numerous modern organisations. This makes it difficult to assess and control idea generation in a way that the organisation can capitalise on the creativity of internal and external stakeholders to maximum. This article investigates the need of a structured approach towards idea generation and management within a company, and tries to give essential elements of an answer on the basis of the analysis of key success factors and a practical case study carried out at a German automotive supplier.

Keywords: Fuzzy Front-End of Innovations, Idea Generation, Idea Management, Process Development, Innovation Management, Automotive Supplier Industry.

1 Introduction

Innovation management in research and practice has largely focussed on finding the ideal innovation process [1][2]. In literature the innovation process is divided in an early phase considered as the front-end of innovation and a later downstream phase [3]. At the cutting point of these two phases, the first official discussions about the realisation of a particular innovation idea take place. This is where the top management decides upon the funding, staffing and the launch or kill of the project [4][5]. This decision point is also called the "money gate" [6].

Although there is a widespread agreement that the fuzzy front-end of innovation plays a critical role in the success of an innovation project, many companies lack a

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 229–240, 2012.

systematic approach to handle these first impulses and/or opportunities for a new product or service during the early part of the innovation process [4][6][7]. The first crucial stage of the fuzzy front-end is the phase of ideation, where ideas are generated and collected internally and/or externally with respect to the organisation [7].

Smith and Reinertsen introduced the term "fuzzy front-end" in 1991 [8] to explain the earliest stages of new product development [4]. This early stage of the innovation process includes all the time spent on the idea as well as the activities enforcing it or not; so the fuzzy front-end covers the steps from idea generation to either its approval for development or its termination [9]. The fuzzy front-end is challenged to combine sufficient room for creativity and freedom of ideation on the one hand, and systematic activities to enhance efficiency on the other [6].

Despite its fuzzy nature, an increasing number of studies highlight the importance of the front-end of research and development (R&D) projects for the overall success of innovations [1][5][10][11]. The reason for this is that decisions made in the very early phase largely determine not only the resulting innovation, but also the whole innovation process with its related costs, time frame and the resources needed [12]. The fuzzy front-end with its sub-phases of idea generation, evaluation and selection affects eminently the quality and maturity of the generated ideas. The effectiveness of the evaluation and selection methods applied during the whole innovation process has a significant impact on the downstream process phases, especially on the development and commercialisation [13].

Due to their highly creative and dynamic character, it is practically impossible to describe the fuzzy front-end activities in the form of one generic front-end process. Senhar points out that the "one size fits all" paradigm assumed in project management literature does not take effect [14]. Consequently, differences in the structural and environmental factors of R&D projects and the increasing importance of this diversity have to be taken into account by R&D management research as well as R&D practice [14][15].

In the last couple of years, the automotive industry has witnessed numerous strategic business activities undertaken by key players worldwide, driven largely by a tough operating environment that has been facing stagnating or even declining demands, rapid consolidation, rising raw material costs and severe price pressures. Original equipment manufacturers (OEMs) have identified product and service innovation as a key long term measure to enhance their market shares. To this end, advanced technology and product development initiatives are becoming critical issues on which all automakers are focusing [16].

As a complementary development, the increasing focus on innovation has also led to the shifting of product development responsibilities from the OEM level to the component supplier level to achieve cost efficiency [17][18]. Since OEMs are under immense pressure to differentiate their products through innovation, some of the top component suppliers have been forced to take over engineering, designing, R&D and assembling responsibilities that were previously assumed by the OEMs. Suppliers are therefore also under pressure to strengthen their R&D investments in order to develop breakthrough products and technologies, which would complement the investments being made by OEMs in the R&D field [19].

Consequently, suppliers have been creating strategic partnerships with other companies and research institutions in order to adapt to this trend. They are also aiming to create integrated technological platforms through which various business units of a company can share knowledge and collaborate. This helps companies speed up product development, reduce costs and deliver better value to customers [20].

Besides creating networked operational processes for the product development, the actors in the automotive market are also shifting towards new forms of innovation management [21]. The systematic involvement of as many as possible stakeholders of the product lifecycle in the innovation management system has huge potential [22]. Ili et al. illustrate in their study that Open Innovation is already considered appropriate for the automotive industry, and that it will be a crucial factor in the next 10 years [23]. However, one of the major concerns with respect to innovation management is an adequate organisation of the fuzzy front-end of innovations, especially the development of an idea generation process [24].

This paper presents strategic steps toward the development of a new idea generation process to enhance the existing innovation management at the automotive supplier KSPG AG according to the authors' earlier publications [25][26]. Also this work spotlights idea generation for high-grade new product innovations, business models innovations or rather service innovations with commercialisation potential on the market. That excludes ideation for pure internal process innovations or cost efficient organisational new changes within companies. In this context, an idea is understood as a start impulse for an innovation activity which is beyond an incremental improvement. This characterisation allows a founded differentiation to contiguous field, like the corporate suggestion system or the continuous improvement process. Both are institutionalised concepts that are primary focused on constant incremental innovations like e.g. the efficiency at the working place or work safety [27] or aligned to improvements of internal processes in administration and manufacturing [28].

The paper is structured as follows: section 2 presents the general framework, such as the selected methodology of coming up with an idea generation process, whose essential process development steps are presented in section 3. Finally, section 4 concludes the paper and gives an outlook on the authors' future research activities.

2 Drivers for the Development of an Idea Generation Process

2.1 Motivation and Methodology

The current innovation management process at KSPG AG is part of the KSPG Advanced Development Process (ADP). The ADP and the division-specific Product Development Processes (PDP) are clearly defined and well established stage-gate processes [2]. The tools used by the innovation management and the ADP are also very well defined and practically proved and applied. The innovation management has been streamed up to the advanced development process and is responsible for the collection, selection, and ranking of product ideas to feed the advanced development department with new promising ideas [25][26].

This established innovation process at KSPG AG follows the innovation value chain paradigm defined in [29]. The innovation value chain is based on applied research and one of today's most important frameworks to explain the innovation process of the most successful companies. Hansen and Birkinshaw recommend considering innovation as a value chain comprising three phases: idea generation, conversion and diffusion [29].

The internal and external spread of product ideas is already very well handled through the ADP and the PDP. Also the development of new products is very well organised in the ADP. The selection method of ideas, up to now the main task of innovation management at KSPG, is also satisfying, however an improvement potential based on the Innovation Database is expected [30]. The focus of improvement lies on idea generation. Figure 1 shows this analysis against the background of the innovation value chain.

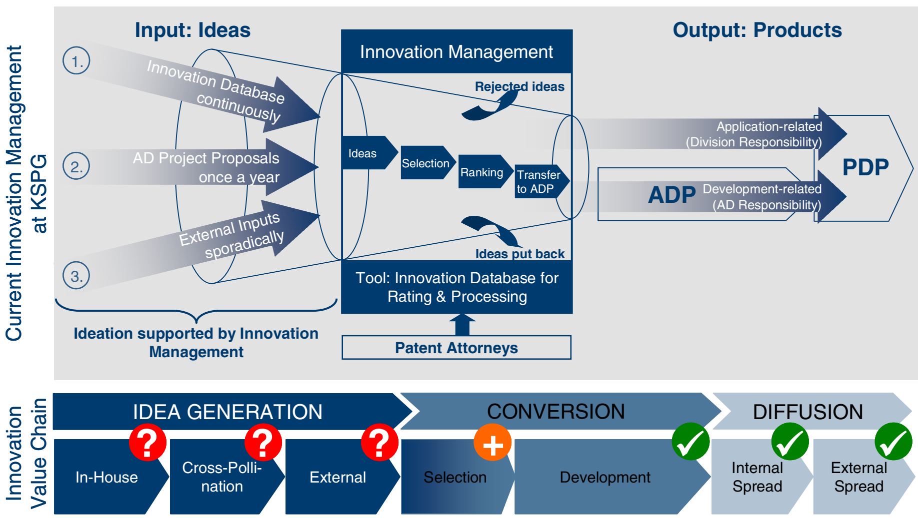


Fig. 1. Current Innovation Management at KSPG AG according to the Innovation Value Chain

Focusing on the fuzzy front-end of the innovation management process, numerous questions concerning the systematisation of the idea generation have come up. The main questions are:

- Where do new ideas come from?
- Which internal and external sources of ideas exist?
- What is the best process?
- What kind of organisational culture supports the generation of ideas?
- Is it possible to measure the success of ideas, and if yes, how?
- Does an overall innovation strategy exist in the company?

Based on these practical questions and a detailed as-is analysis of the current idea sources at KSPG AG [26], the challenges of the innovation management at KSPG AG have been defined as follows:

- Implementation and optimisation of a company-wide strategic innovation management.
- Innovation management has to run in a structured way to guarantee sustainability through the whole product life cycle.
- Systematic innovation management is an important means of decision support.
- The fuzzy front-end of innovation management should be clearer.
- Innovation management has an important role in the active generation of ideas.
- An evaluation scheme allowing to monitor ideas and to rate their commercial success is required.
- The establishment of an organisational culture that leverages innovation by motivating employees to be creative and to actively generate and communicate ideas.

The target is to propose and implement an idea generation process based on the integration of external and internal stakeholders to feed the innovation funnel systematically and continuously with promising new ideas. To this aim, a pilot project has been started in February 2011, which focuses on the establishment of a process for the fuzzy front-end, and its application to a specific strategic target innovation field of the company. Its main deliverable will be an idea generation process that will help to estimate the potential of systematic stakeholder integration in innovation management. Furthermore, several positive effects are expected to occur thanks to this project:

- The project will come up with a clear documentation of the methodology that has been applied, and the experiences gained from it. It will also deliver a critical assessment of each step, as well as of the global result.
- It will create a positive attitude of stakeholder groups with respect to their own involvement in innovation management.
- It will open the mindset of affected stakeholders for changes that will significantly contribute to the improvement of the organisation's innovation power.
- The project will deliver an increased number of new ideas contributed by several experts from different fields.

Based on the results and experiences, this methodology will offer methods and tools ready for formal integration into the existing innovation management system of KSPG AG.

2.2 Objectives

The analysis of the existing innovation management system at KSPG revealed that:

1. currently at KSPG AG ideation consists of the collection of ideas rather than their generation, and
2. idea generation is limited to a core group of employees who act as idea contributors [26].

This situation represents a threat of idea stagnation. This is why the company's innovation management has declared the improvement of idea generation as one of its major strategic objectives. Based on this, the main goals of the pilot project introduced in section 2.1 are the following:

- to develop a KSPG specific idea generation process and
- to implement this new process successfully into the existing innovation management process.
- Furthermore, the idea generation process has to be exemplarily applied to the field of electro-mobility (e-mobility).

The application of the idea generation process has the following aims:

- to examine the new developed idea generation process with regard to effectiveness and efficiency,
- to embed internal and external stakeholder with different expertise in the idea generation process by applying selective methods,
- to generate ideas for future product innovations and business model innovations with special regards to e-mobility,
- to pursue quantitatively and qualitatively the idea generation process and support pro-actively the hereby connected activities, and
- to assess the applicability for future innovation fields by a detailed analysis of the approach.

3 Development of an Ideation Process at KSPG AG

The following questions serve as a guideline to achieve the objectives mentioned in section 2.2, in the particular context of the KSPG AG:

- How do other enterprises structure their idea generation process within and outside of the automotive industry?
- Which Best Practice examples can be derived?
- Which Lessons Learned have to be considered during the development and implementation of an idea generation process?
- How can the idea basis of the KSPG AG in consideration of all external and internal aspects be effectively extended?
- Which internal and external sources are especially suitable for the generation of ideas in general and for the KSPG AG in particular?
- Which methods are particularly efficient for the idea generation, and which of them can be optimally used within the KSPG AG?
- Which core competencies and innovation priorities does the KSPG AG possess?
- How high is the receptiveness of internal and external ideas?
- Which further processes, methods and systems are connected with the idea generation process (decision making process, declined ideas, etc.)?

In the following, some of the main workflow steps will be described in detail.

3.1 Analysis of the Initial Situation at KSPG AG

The current innovation management at KSPG AG had been investigated earlier with the participation of internal and external experts. For this first as-is analysis the operative project team counted three members – one insider employed at KSPG AG and two outsiders with research and consulting expertise. The insider perspective offered detailed know-how about typical practices in the daily business at the reviewed company. The outsider perspective allowed a critical distance to this processes and activities and an in-depth reflection based on experiences from the concerned business sector. To sum up, through this collaborative observation of new knowledge about the existing situation and associated restrictions could be produced.

In addition to that, a quantitative survey of 437 ideas from KSPG's two divisions Pierburg and Pierburg Pump Technology (PPT), which have led to patents and product innovations, reveals that most of the ideas came up through the idea contributor's own consideration. Figure 2 summarises the results of the analysis.

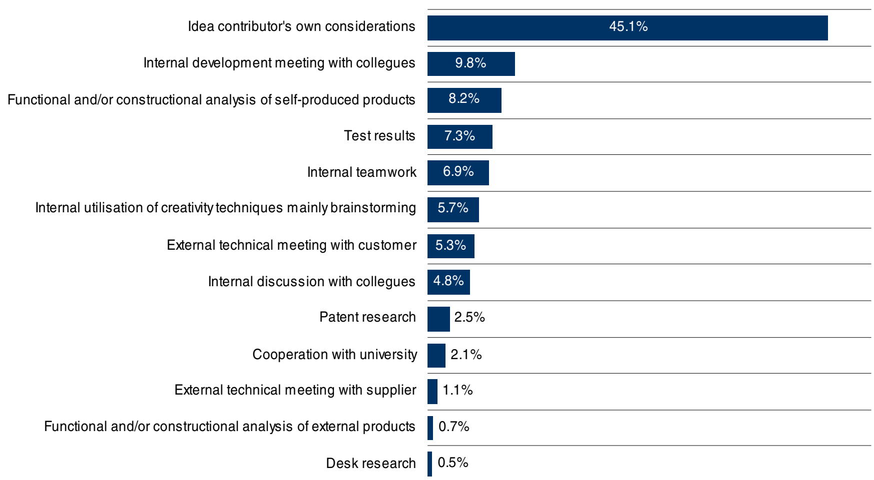


Fig. 2. Origins of Ideas (n=437 patent applications at Pierburg and PPT)

Another main origin of ideas is the internal exchange with colleagues, either during internal development meetings, teamwork, internal brainstorming sessions or discussions. Tests of existing self-produced or external products and the analysis of current market requirements and future trends help to generate ideas in 19.2% of the cases. Very important for the implementation of a stakeholder integrated idea generation process in the existing system is the fact that only 8.5% of the ideas occur through the involvement of external stakeholders. The most influential external stakeholders are the customers, universities and suppliers.

The as-is analysis shows that several organisational measures will have to be adopted in order to capitalise on a more open innovation system and to exploit its potential.

3.2 Analysis and Evaluation of Best Practice Examples

The authors' examination of best practice examples from the automotive industry (OEMs and suppliers) and other business sectors, like telecommunications, internet business, chemistry etc., results in the identification of following key success factors for an effective idea generation and management process:

- Make innovation a subject of every employee.
- Focus on clearly defined and communicated strategic ideation topics and targets.
- Give employees the mission and the freedom to be creative, and to elaborate their ideas to a high degree of maturity.
- Assure quick response times to idea submissions.
- Assure systematic and transparent pursuit of ideas.
- Leverage interdisciplinary teams with individual leadership.
- Systematically activate and involve potential idea sources within and outside the organisation.

In particular, the authors identified that an open innovation culture within the organisation is a major prerequisite in order to implement an effective idea generation process. All the above mentioned success factors obviously contribute to create and reinforce this kind of culture. Also, the authors observed that there is a variety of different methods and tools that help implementing these success factors. Looking at Best Practices is very useful, but one has to be aware of the fact that what is a best practice in one organisation is not necessarily a practice in another. Their relevance and effectiveness within a given environment depend on the specific organisational culture. The major challenge of this project will be to find those methods and tools that can be applied to KSPG in a way that they fit into the current organisational culture, yet at the same time lead to the desired cultural transformation. Naturally, the ultimate target of this work is to derive from these experiences elements of practices and indicators that can be generalised to a certain extent, and can therefore serve as valuable guideline for other organisations.

3.3 Approach towards an Ideation Process at KSPG AG

The reflections so far lead to the assumption that especially the following ideation activities seem to be practicable in the environment of KSPG AG:

- Top management defines strategic innovation topics, for which ideation is wanted (prioritisation).
- Experts from different fields are nominated, and get the mission to work together half a day per week on a new idea topic in so-called ideation teams (freedom and multidisciplinary teams).
- Once per months, results (new ideas, advances on existing ideas) are consolidated and reported (systematic and transparent follow-up).
- Each ideation team needs a leader who can act as the moderator and reporter (leadership).
- The core of ideation teams should be stable over time; however, teams can be extended and re-composed at certain points of time.
- Deliverables of these teams are product/service ideas well formulated in the Innovation Database and with a high maturity level.
- Regular involvement of top management is crucial for the success (immediate feedback)

An essential element underlying these actions is the personal communication and the continuous flow of information and knowledge among different experts. Such exchange should be leveraged by in dedicated ideation meetings. The proposed time intervals for meetings and reporting to the top management are indicative, and can be adjusted to the major objectives of the meetings:

- The wider the ideation topic, and the earlier the status of the ideation process, the longer the meeting intervals can be.
- The more concrete the discussed ideas are, the more often the ideation teams should meet, and the more intensively their work should be targeted at making the idea(s) more mature (idea maturation process).

The already existing Innovation Database [30] can support these meetings effectively as a reporting tool. Forms should be available for all the criteria needed by the strategic decision committee. Also, the role of the team moderator (leader) is vital for leading the team discussions into the right direction from the very beginning. New ideas coming up during such meetings (even if they are not directly related to the focus idea under discussion) have to be kept track of, and communicated after the meeting.

3.4 Essential Elements of the Ideation Process

Implementing these measures requires the commitment from top level management, as they affect all business units and departments [31]. A prerequisite for this is to be able to show that these are not isolated actions, but rather part of a consistent process that aims at feeding the ADP with more and more mature ideas. Elaborating this process in a way that it can be implemented within the organisation will be an essential part of the project. At the time of writing this article, there were not sufficiently many validated and non-confidential results available to publish. However, the essential elements that serve as a guideline for the development of this process will be listed in this section.

Essentially, the authors want to distinguish between three phases:

1. Prerequisite,
2. Execution,
3. Selection.

The prerequisite phase covers all the activities expected from top management level to set the right frame for the ideation activities:

- External and internal analysis of the needs and situations of the business units mainly in terms of marketing and core competencies;
- definition to the business unit's innovation strategy;
- agreement on the ideation targets and priorities;
- agreement on indicators and assessment criteria;
- commitment to available resources;
- commitment of top level management visible for all employees.

Most of these prerequisites needed to pass on to execution should be available to a large extend from strategic activities that are already carried out as part of existing processes, and are therefore not to be seen as additional charge imposed by the new ideation process.

Essential activities to be carried out during execution are the following:

- Stakeholder management;
- expert network management;
- partner management;
- creation of a spirit of challenge and competition;
- selection and deployment of ideation tools and methods.

Throughout the ideation execution phase there has to be good balance between freedom for creativity and relevance to the clearly communicated innovation targets, as well as the defined indicators and assessment criteria. Only this will assure that ideas will mature and propagate to the final selection phase, in which the following steps are considered most important:

- Idea assessment;
- idea communication;
- idea transfer to the ADP.

It is very important that this framework is implemented in a way that it allows for the flexibility required to leverage the intrinsic dynamics of idea generation, i.e., ideas coming up at any stage of this process must be handled efficiently. Also, ideas that are not selected have to be maintained in a way that they can participate in future selection phases: ideas not relevant today may become relevant tomorrow.

These guidelines will serve to define a new ideation process at KSPG AG, and introduce it in the context of the strategic e-mobility subject.

4 Conclusion and Outlook

Structuring the fuzzy front-end of the innovation process has turned out to be an effective measure in many of today's innovation leading companies. The way of structuring ideation, as well as the choice of methods and tools depends on the current and targeted innovation culture of a specific enterprise. This article attempts to establish an experience-based framework for the development of an ideation process at the German automotive supplier KSPG AG. One of the identified key issues is the need for the achievement of top level management support for the undertaking of targeted ideation activities by employees. Many of the proposed steps are thus oriented towards the achievement of this commitment, as well as its clear communication within the complete organisation. The ideation process has to clearly address all employees in a way that each of them understands that her/his contributions to ideation for the company's strategic innovation topics are really wanted and leveraged.

The authors' current and near-future activities aim at defining the process based on the identified criteria and objectives, and apply it to the subject of e-mobility. This will not only give indispensible feedback from experience, but also trigger the initial transformation of the company's innovation culture, which will be a prerequisite for the successful deployment of the ideation process.

References

1. Bröring, S.: The Front End of Innovation in Converging Industries – The Case of Nutraceuticals and Functional Foods. Deutscher Universitäts-Verlag/GWV Fachverlage GmbH, Wiesbaden (2005)
2. Cooper, R.G.: Winning at New Products: Creating Value Through Innovation, 4th edn. Basic Books, New York (2011)
3. Globocnik, D.: Front End Decision Making: Das Entstehen hochgradig neuer Innovationsvorhaben in Unternehmen. Gabler Verlag, Wiesbaden (2011)
4. Khurana, A., Rosenthal, S.R.: Integrating the Fuzzy Front End of New Product Development. Sloan Management Review 38(2), 103–120 (1997)
5. Koen, P.A., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amoe, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejkov, A., Wagner, K.: New Concept Development Model: Providing Clarity and a Common Language to the 'Fuzzy Front End' of Innovation. Research Technology Management 44(2), 46–55 (1997)
6. Herstatt, C., Verworn, B. (eds.): Management der frühen Innovationsphasen: Grundlagen – Methoden – Neue Ansätze, 2nd edn. Gabler Verlag, Wiesbaden (2007)
7. Bullinger, A.C.: Innovation and Ontologies – Structuring the Early Stages of Innovation Management. Gabler, Wiesbaden (2008)
8. Smith, P.G., Reinertsen, D.G.: Developing Products in Half the Time. Van Nostrand Reinhold, New York (1991)
9. Zhang, Q., Doll, W.J.: The fuzzy front end and success of new product development: A casual model. European Journal of Innovation Management 4(2), 95–112 (2001)
10. Kim, J., Wilemon, D.: Focusing the Fuzzy Front-End in New Product Development. R&D Management 32(4), 269–279 (2002)
11. Stevens, G.A., Burley, J.: Piloting The Rocket of Radical Innovations – Selecting the right people for the right roles dramatically improves the effectiveness of new business development. Rechearch-Technology Management 46(2), 16–25 (2003)
12. Bröring, S., Leker, J.: Decision Making in the Front End of Innovation on new Industy Segments – the case of the new nutraceuticals and funcitonal foods sector (Finalist Best Paper Award). In: Bart, C., Bontis, N., Head, M. (eds.) Converence Proceedings of the 25th McMaster World Congress, Hamilton, Canada (2004)
13. Murphy, S.A., Kumar, V.: The front end of new product development: a Canadian survey. R&D Management 27(1), 5–16 (1997)
14. Shenhar, A.: One size does not fit all projects: Exploring classical contingenca domains. Management Science 47(3), 394–414 (2001)
15. Butler, J.: New challenges for R&D Management. In: Durand, T., et al. (eds.) Bringing Technology and Innovation into the Boardroom, Houndmills, pp. 260–291 (2004)
16. Ili, S.: Open Innovation im Kontext der Integrierten Produktentwicklung – Strategien zur Steigerung der FuE-Produktivität (Open Innovation in Context of Integrated Product Development – Strategies for Increasing RnD-Productivity). IPEK Forschungsbericht, Band 33, Institut für Produktentwicklung Universität Karlsruhe (TH), Karlsruhe (2009)
17. Dannenberg, J.: Große Chance für Zulieferer. Interviewed by Tina Rumpelt. In: Automobil Produktion, pp. 18–19. Innovationen in der Automobilindustrie, Sonderausgabe (2008)
18. Dannenberg, J., Burgard, J.: Car Innovation 2015 – A comprehensive study on innovation in the automotive industry. Published by Oliver Wyman Automotive, Munich (2007), http://www.car-innovation.com/pdf/studie_car_innovation_2015.pdf
19. Kurek, R.: Erfolgsstrategien für Automobilzulieferer – Wirksames Management in einem dynamischen Umfeld. Springer, Heidelberg (2004)

20. Stockmar, J.: Erfolgsfaktoren für Automobilzulieferer – Strategien für 2010. In: Ebel, B., Hofer, M., Al-Sibai, J. (eds.) Automotive Management – Strategie und Marketing in der Automobilwirtschaft, pp. 13–38. Springer, Heidelberg (2004)
21. Ili, S. (ed.) Open Innovation umsetzen – Prozesse, Methoden, Systeme, Kultur. Symposion Publishing GmbH, Düsseldorf (2010)
22. Neumann, M., Riel, A., Brissaud, D.: Sustainable innovation management in the automotive indrusty. Int. J. Technology Intelligence and Planning 7(4), 327–343 (2011)
23. Ili, S., Albers, A., Miller, S.: Open innovation in the automotive industry. R&D Management 40(3), 246–255 (2010)
24. Neumann, M., Riel, A., Brissaud, D.: Improvement of Innovation Management through the Enlargement of Idea Sources. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 121–132. Springer, Heidelberg (2011)
25. Neumann, M., Riel, A., Brissaud, D.: Innovation Management System for the Automotive Supplier Industry to Drive Idea Generation and Product Innovation. In: Proceedings of the 17th EuroSPI Conference, vol. 17, pp. 7.13–7.24 (2010)
26. Neumann, M., Riel, A., Brissaud, D.: Improvement of Idea Generation by Capitalisation on Internal and External Stakeholders. In: Industrial Proceedings of the 18th EuroSPI Conference, vol. 18, pp. 3.25–3.33 (2011)
27. Thom, N.: Betriebliches Vorschlagswesen – Ein Instrument der Betriebsführung und des Verbesserungsmanagements, 6th edn. Peter Lang, Bern/Berlin/Bruxelles/Frankfurt a.M./NY/Oxford/Wien (2003)
28. Kostka, C., Kostka, S.: Der Kontinuierliche Verbesserungsprozess – Methoden des KVP, 5th edn. Carl Hanser Verlag, München (2011)
29. Hansen, M.T., Birkinshaw, J.: The Innovation Value Chain. Harvard Business Review 85(6), 121–130 (2007)
30. Neumann, M., Riel, A., Brissaud, D.: IT-supported innovation management in the automotive supplier industry to drive idea generation and leverage innovation. Journal of Software Maintenance and Evolution: Research and Practice (2011), doi:10.1002/smr.578
31. Hauschildt, J., Salomo, S.: Innovationsmanagement, 5th edn. Verlag Franz Vahlen, München (2011)

Launching Innovation in the Market Requires Competences in Dissemination and Exploitation

Kerstin Siakas[1], Richard Messnarz[2], Elli Georgiadou[3], and Marja Naaranoja[4]

[1] Alexander Technological Educational Institution of Thessaloniki,
Department of Informatics, P.O. Box, 141, GR-57400 Thessaloniki, Greece
siaka@it.teithe.gr
[2] International Software Consulting Network Ltd.
Florence House, 1 Florence Villas,
Bray, Co. Wicklow 1E02, Ireland
rmess@iscn.com
[3] Middlesex University, School of Engineering and Information Sciences,
The Borroughs, London NW4 4BT, UK
e.georgiadou@mdx.uk
[4] Unívers̀ity of Vaasa, Department of Industrial Management
P.O. Box 700, FIN-65101 VAASA , Finland
marja.naaranoja@uwasa.fi

Abstract. It is widely recognised that innovation is required for economic growth on a number of levels, such as in Small and Medium sized Enterprises (SMEs), large organisations, regions and nations. Innovation is an important driver for organisational survival, sustainability, improvement, long-term productivity and economic growth. However, innovation in isolation is not only wasteful but also not useful. All projects, and in particular innovation projects, need to disseminate and exploit their results for maximising achievements and increasing sustainability after their completion. This includes launching of the innovation to market, transfer of results and best practices to different and broader contexts; potential tailoring to the needs of others; continuation after the funding period has finished; influences on policy and practice; as well as serving the public good. The emphasis should be on optimising the value of the project and on boosting its impact.

Many European and other projects seem to be missing a good valorisation strategy. Projects seem to be carried out in isolation, and finish without essential impact. In addition there is a lack of skills for carrying out valorisation actions. Taking this into consideration a new certified Valorisation Expert profession was created and sponsored by the European Commission aiming to fill the gap and to train various stakeholders in launching innovative and other project results in the market. In this paper we describe the aims and the objectives of the project and the competences base the project is aiming to create through its content and teaching approach.

Keywords: Innovation Networking, Stakeholder Networking, Dissemination Strategies.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 241–252, 2012.

1 Introduction

Recent trends in the world economy including globalisation of markets and technology supply, networking as business models and technologies enabling distance mode of working have increased the complexity and competition level of organisations. There is a push for organisations to produce innovative products and services for survival, sustainability and growth. Simultaneously processes need to be innovative to keep costs down and to improve productivity. Innovation is about finding new ways of doing things and of obtaining strategic advantage.

The degree of novelty in products or services (unique offering), and/or in process (faster, lower cost, customisation) is decisive for the strategic advantage of innovation. 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 [16]. It is not always the innovation or the technology in itself that matters, but innovation-in-use (e.g. growth of email use). In [3] they argue that innovation is an invention implemented and taken to market.

Beyond innovation there lies disruptive innovation that changes social practices by changing consumer behaviour and/or causing disruption in the way business is done. In [4] they show, by using lessons learned of successes and failures of leading companies, how disruption innovations can initially be rejected by mainstream customers because they are not ready to use the new product or service. As a result firms with strong customer focus may allow important innovations to languish, because they did not concentrate on new customers for the products or the services of the future. The more innovation develops over time, the more players are brought into the game [17]. A complex exchange network emerges of individuals and interest groups (customers, partnerships, joint ventures, company acquisitions, sponsors etc.) engaging in various transactions necessary for moving the innovation forward and to launch it on the market.

Many innovative projects, however, seem to be missing an innovative business model and a good valorisation strategy. The French term valorisation is often used encompassing all activities that maximise the achievements of a project, including dissemination and exploitation of results and outputs.

The word dissemination derives from Latin 'disseminare' ('*dis'* means to spread or scatter widely and *'semen'* means seed[1]) and indicate spreading of information to ensure that others benefit from experiences gained in the project. Exploitation of the results of project activities means that more people can share in the successes, experiences and lessons learned.

The European Commission defines dissemination as "a planned process of providing information on the quality, relevance and effectiveness of the results of programmes and initiatives to key actors. It occurs as and when the results of programmes and initiatives become available and exploitation as *"mainstreaming: the planned process of*

[1] http://www.thefreedictionary.com

transferring the successful results of programmes and initiatives to appropriate decision-makers in regulated local, regional, national or European systems, and multiplication: the planned process of convincing individual end-users to adopt and/or apply the results of programmes and initiatives“[2].

2 Innovating in Europe

In EuroSPI (European Systems and Software Process Improvement and Innovation) European industry and task forces for innovation published various case studies about success criteria in European industry.

A most recent set of publications [11], [12], [13] illustrates that the European industry has transformed into a networked cluster of partnerships and stakeholders who jointly integrate larger products in e.g. Automotive, Aerospace, and medical industry. Strategies were developed by leading industry to create idea and stakeholder networks and infrastructures to build a networked ground for idea creation, innovation evaluation and exploitation of ideas (see Figure 1).

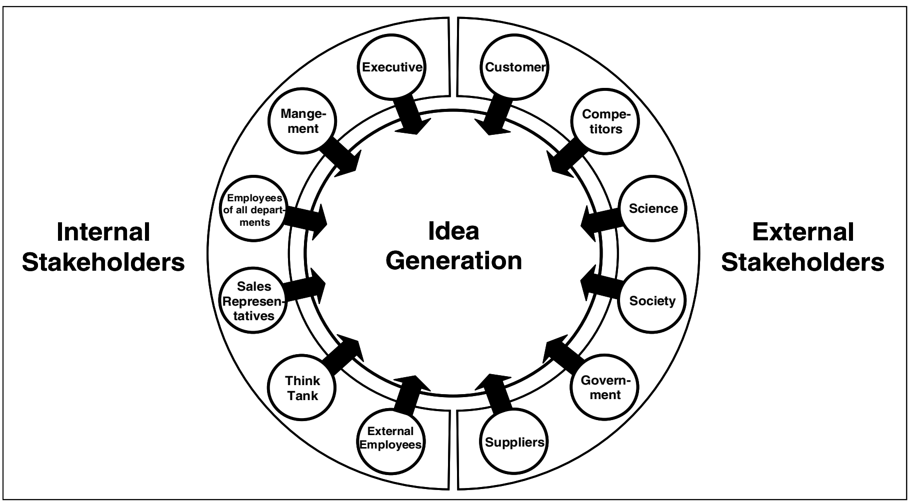


Fig. 1. Making Idea Generation a Professional Process Linked with Lead Industry [20]

In addition to that KSPG (a leading Automotive company owned by Germany’s largest steel industry) published that companies nowadays need a professional infrastructure to support this level of networking but also to integrate ideas in a database and to evaluate with the help of supervision teams and transporting them into real products.

---

[2] http://ec.europa.eu/education/programmes/llp/guide/valor/what_en.html#2

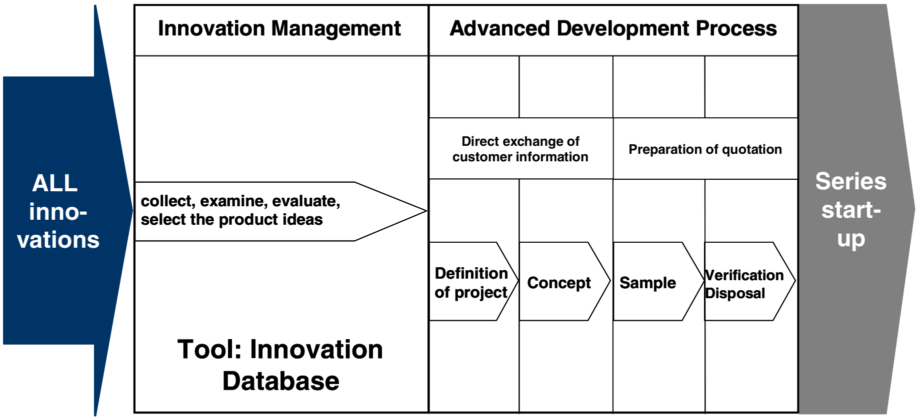


Fig. 2. Supporting Idea Generation and Innovation with Professional Infrastructure [12]

In [10] the transformation of the innovation networking in a European dimension is described and it is highlighted that currently virtual expert topic clusters are built which are connected to such stakeholder networks to deliver solutions for leading industry in Europe. This type of dynamic networking is then seen as a driver for European innovation success in the future.

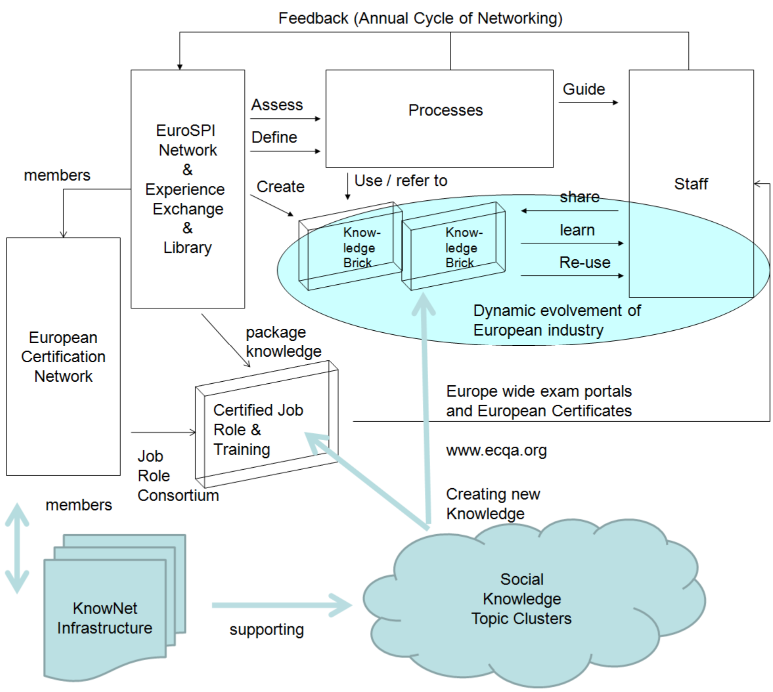


Fig. 3. EuroSPI Strategy 2020 - Expert Clusters as Knowledge Clusters to be Networked to Industry and Qualification Strategies [10]

The EuroSPI 2020 objectives published in [10] foresee that the networking and knowledge exchange has to be enriched by service based networking functions (shown in Figure 3), including:

1. Packaging knowledge into service packages and training with certificates. For this a European Certification and Qualification Association (ecqa.org) was set up supported by a series of EU projects from 2005 – 2012.
2. Creating groups of experts who form a knowledge community which can solve problems and give advice to industry. This resulted in e.g. workshop communities around EuroSPI and in e.g. Germany the Bavarian industry kick off financed in 2003 a set of cross – company task forces in areas required to keep leadership on the market.
3. Building on social media based knowledge communities and software which can create a knowledge based for innovation networking across regions and companies.

If innovations are successful this usually leads to the hype cycle which was published by Gartner group and is continuously discussed in various articles [1],[10].

The “1. Technology Trigger” phase in the hype cycle includes building a critical mass of interest which leads to a growing investment into the innovation as a successful start up.

Also the SPI Manifesto [14] which describes 3 key values and 10 success principles for industry implementing improvement and innovation contains organisational success principles which directly support the dissemination and distribution of improvement ideas, human aspects and networked learning and growth.

In addition, the EU project ORGANIC (Certified Innovation Manager, 2003 – 2005, [8]) developed competencies for dynamic learning and networking to build on continuous innovation and growth.

Conclusion Concerning European Industry

This means that innovations in Europe must get linked into these stakeholder networks, expert clusters, and organisational strategies of leading industry to get their ideas accepted, exploited and used in a broad sense. A good valorisation strategy must find ways to disseminate and network into the industry. And researchers can expect that leading industry has created such networks. It is necessary to find a way to get connected.

3 Innovation Success Factors in General

Innovative organisations usually do extremely well in their core mission, but they often fail to capitalise on their own learning [5]. They use passive approaches relying on the assumption that evidence-based practice is enough to spread innovations and best practice into broad use across organisations and among organisations and the market in terms of consumers and competitors. Dissemination, however, is not just to inform potential stakeholders about innovations and best practice but to embed these in the organisational culture, including structures, processes, underlying behaviours etc.. The critical initial criterion for dissemination comprises an innovation or practice that is worthy of replication and spread. This is equivalent to a CMMI maturity level of 2 (managed). It can be said that at this level the organisation focuses on projects and is reactive rather than pro-active. Organisational conditions need to be supportive of inventive activity, originality of thought, creativity and breakthrough insights. It is generally recognised that creativity generates the basis of innovation [2]. The likelihood for innovative work attuned and responsive to the market in terms of consumers and competitors is only made possible by social conditions inside organisations that can affect psychological conditions likely to lead to creative work. The creation of an innovative culture involves a learning process that builds on evaluation, reflection and development of the organisation toward response maturity for emerging challenges. The relationship between social attribution and technological possibilities are cornerstones for the learning process. An emergent challenge is tapping collective explicit and tacit knowledge and intelligence of users (customers and consumers) by social media networks and thus reaching beyond the conventional boundaries of the organisation. Users' tacit knowledge can for example be tapped through reflection in practice by launching prototypes for user tests before the product is launched on the market. Another key factor in tapping collective knowledge is the leverage if disparate assets of people from different cultures, different disciplines and different organisations. Today the Information and Communication Technologies (ICTs) provide opportunities, such as social media, to connect people together in a totally different fashion than before and also to shape artificial intelligence prototypes that can evoke tacit opinions by customers.

In [16] they propose that key factors for competitive success include organisation specific knowledge and capacity to exploit this knowledge. Other success-factors are related to the accumulation of the firm specific knowledge, level of uncertainty regarding present and future technology developments, competitive threats and market demands. In addition they argue that internal structures and processes must continuously balance conflicting requirements to identify, develop and exploit specialised knowledge across technological fields, business functions and product divisions. Ideas flowing out of the organisation for evaluation and flowing into the organisation as new offerings and new business models is called Open Innovation [3]. Finding the right balance and mechanism for this situation seems to be a core issue.

In [6] they discuss the concept of C-space (Culture Space) by analysing the flow of knowledge within and among organisations. The framework consists of two dimensions, the codification, that expresses the extent to which information can be expressed and the diffusion, the extent to which information is shared by a given population. The interaction between codification and diffusion result in the social learning curve in culture space depicted in Figure 4.

The social learning curve consists of four stages: problem solving, diffusion, absorption and scanning. The problem-solving process results in higher levels of codified information (ease of expression), which in turn increases the diffusion level and the absorption of the information by external stakeholders. The scanner stage is the feedback loop that contributes to learning and thus to the social learning curve.

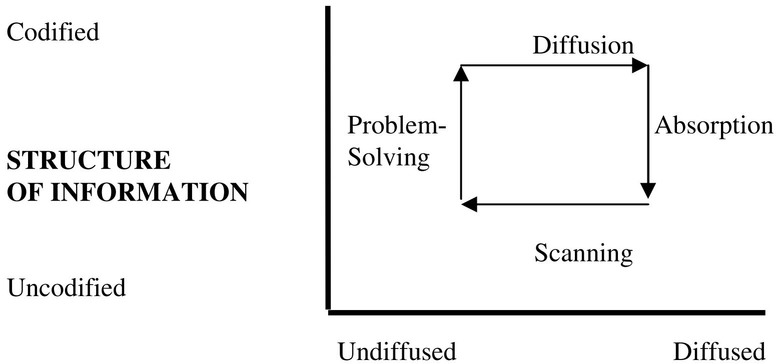


Fig. 4. The social learning curve in culture space

In [8] the authors discuss dissemination of innovation in community psychology by looking at the Development of Innovation (DOI) paradigm related to the improvement of human and community functioning. They conclude that the DOI typically involves the dissemination of particular social programs that consists of three components, namely adoption, implementation and institutionalisation. They also argue that DOI, as a paradigm, offers a valuable framework to community psychologists for social change, including developmental perspectives on innovation, properties of innovations and innovators, implementation and post-adoption activity, measuring implementation, fidelity vs. adaptation and reinvention, as well as dissemination of interventions and innovations into routine practice.

4 Diffusion of Innovation

Using a staged model of behaviour change, untying is seen as a precursor for dissemination activities, which in turn exposes one to consider change in their practices e.g. use the research results. Implementation activities capitalise on this by "enabling" and subsequently "reinforcing" the desired behaviour change. Different skills are needed for each activity.

Rogers [15] defines the innovation-decision process as the "process through which an individual (or other decision making unit such as a group, society, economy, or country) passes through the innovation-decision process". According to Rogers there are five stages in the innovation-decision process:

(1) from first knowledge of innovation,
(2) to forming an attitude toward the innovation,
(3) to a decision to adopt or reject,
(4) to implementation of the new idea,
(5) to confirmation of this decision.

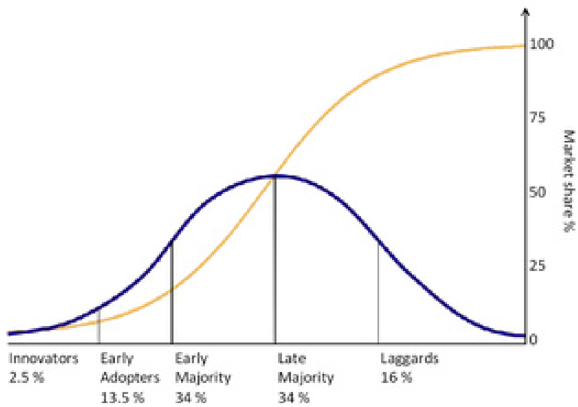


Fig. 5. With successive groups of consumers adopting the new technology (shown in blue), its market share (yellow) will eventually reach the saturation level. In mathematics the S curve is known as the logistic function (Rogers 1962, Fisher et al 1997)

The previous practice, felt needs/problems, innovativeness, and norms of the social systems affect the innovation decision process. The first stage of the innovation-decision process entails seeking one or more of three types of knowledge about the innovation. Rogers [15] describes these as:

1. Awareness knowledge is information that an innovation exists.
2. How-to-knowledge consists of the information necessary to use an innovation properly, and
3. Principles knowledge consists of information dealing with the functioning principles underlying how the innovation works.

Rogers states that awareness and knowledge of an innovation can be made most efficiently through mass media. It will be interesting in twenty years or so, to ascertain if mass media will still be considered the most efficient means to create product awareness and knowledge.

According to Rogers [15] the diffusion of innovations depends on the type of adopter. Rogers defines an adopter category as a classification of individuals within a social system on the basis of innovativeness. Rogers suggests five categories of adopters in order to standardize the usage of adopter categories in diffusion research. The adoption of an innovation follows an S curve (Fig.3) when plotted over a length of time [7]. The categories of adopters are: innovators, early adopters, early majority, late majority, and laggards [15]. Each of these categories need a different approach when disseminating the innovation.

5 Building on the Valorisation Competences

In order to support organisations in building competences in dissemination and exploitation of results of innovations, research projects and lifelong learning projects a new valorisation competence is currently developed within the frame of the

European Certification and Qualification Association[3]. The *ECQA Valorisation Expert Training and Certification*" (VALO) two year project started in October 2011 with funding from the EU Lifelong Learning Programme. The project aims at developing a new skill set and a job role qualification study program, where competencies in valorisation are customised for the European industry into an online study program complemented with an on-line examination and certification training and certification schema for Valorisation experts. A pilot training will take place in the participating organisations/member states (Austria, Finland, Greece, Ireland and the UK) and the study programme will be refined and improved based on systematic feedback. Table 1 provides the skills developed during the training. The training material is modular, consisting of Units (U) and Elements (E). Every element consists of 3-6 performance criteria in line with the European Qualification Framework (EQF)[4].

6 Pedagogic Framework and Quality Assurance

The production of the training material follows the European Learning Outcomes framework embodied in the Bologna Process whereby expectations at the end of the study are expressed as knowledge and understanding but also kills i.e. what the learner will be able to do The training material available in the learning portal is supported by exercises. The accredited training can either be provided in traditional or distance mode. The same structure for trainings, self test, exam and certificate ensure a common Europe wide integrated Infrastructure and training base. The ECQA Exams are computer-based and random. Exams are generated from a large central question pool translated to several European languages (depending on countries participating at each profession schema).

One of the objectives of the Bologna Process is the establishment of quality assurance systems enabling the evaluation of programmes and institutions in the form of internal assessment together with external reviews [18]. It is aimed that all stakeholders (member states, institutions, staff, students, industry) participate in the process and that results are transparent so that good practice and lessons learned can be shared [19]. For this reason the development of the materials will undergo systematic reviews and updates by specialists drawn from members of the consortium. Because the learner will engage in self directed study and self assessment it is important to anticipate as many learner questions as possible. The exercises, scenarios and model answers will provide knowledge, guidance, feedback and encouragement to the learner.

As the number of people who are taking the Valorisation Manager Certification grows (wider acceptance) the database of exercise and exam questions will grow. In order to make the system sustainable comprehensive and systematic updating, maintenance and quality assurance of the database are incorporated enabling wider dissemination, and providing a self-funding system. The ultimate measure of success

[3] www.ecqa.org

[4] http://ec.europa.eu/eqf/home_en.htm

Table 1. The Skill Card of the ECQA Valorisation Competences

U1	**Understanding the importance concerning valorisation of innovation and EU project results**
U1E1	Introduction to valorisation (terms such as: broad and deep dissemination – exploitation: sustainability, value creation and mainstreaming)
U1E2	Innovation
U2	**Dissemination**
U2E1	Broad Dissemination
U2E2	Deep Dissemination
U2E3	Dissemination Strategy
U3	**Exploitation**
U3E1	Creating Stakeholder Value
U3E2	Sustainability
U3E3	Mainstreaming
U3E4	Exploitation strategy
U4	**Valorisation Methods**
U4E1	Diffusion
U4E2	Communication to potential stakeholders
U4E3	Valorisation channels - formal and informal valorisation
U4E4	Valorisation tools (cluster building, open innovation etc.)
U4E5	Intellectual Property Rights (IPR)

will be the widespread dissemination and the sustainability of the project outcomes. The benefit will satisfy the requirements of all stakeholders starting from the individuals (teachers and learners), the participating organisations and the funding bodies. Valorisation of the results will feed back to the reviewing process of new research and development proposals to avoid duplication of effort and re-inventing the wheel.

7 Conclusions

All projects (innovations, research projects and LLPs) need to valorise (disseminate and exploit) their results for maximising achievements and increasing sustainability after their lifetime. This includes transfer of results and best practices to different and broader contexts. Many European and other projects seem to be missing a good valorisation strategy. Projects seem to be carried out in isolation, and finish without essential impact. In addition, there is a lack of skills for carrying out valorisation actions. The aim and objective of this work is to fill the gap and create a new certified Valorisation Expert profession adding to the 18 existing certifications in the European Certification and Qualification Associations. The skill card of the new profession was described. Before the development of the actual training material feedback is sought from the industry and other stakeholders in order to develop competences valuable to the market.

European industries have already started to develop and implement open innovation strategies by building up cross company task forces, networks of stakeholders, idea databases, innovation supervision teams, etc. [9],[10],[11],[12][13]. The dissemination and exploitation of new ideas in future requires to get connected to these networks and get dynamically involved. This will require considerable dissemination, valorisation and especially networking competences.

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. This publication reflects only the views of the authors, and the Commission cannot be held responsible for any use, which may be made of the information contained therein.

References

1. Biró, M.: The Software Process Improvement Hype Cycle. In: Dalcher, D., Fernández-Sanz, L. (guest eds.) Invited contribution to the Monograph: Experiences and Advances in Software Quality. CEPIS UPGRADE, vol. X (5), pp. 14–20 (2009)
2. Bullinger, H.J.: Turbulent times require creative thinking: new European concepts in production management. International Journal Production Economics (60-81), 9–27 (1999)
3. Chesbrough, H.: Open Innovation, The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, Boston (2003)
4. Christensen, C.M.: The innovator's dilemma: when technologies cause great firms to fail. Harvard Business School Press, Boston (1997)
5. Dearing, J.W.: Dissemination of Innovation: the Will to Change an Organisation. The Permanente Journal 12(3), 75–77 (2008)
6. Griffiths, D., Boisot, M.: Are there any competences out there? Identifying and using technical competences. In: Tidd, J. (ed.) From Knowledge Management to Strategic Competence, pp. 199–229. Imperial College Press, London (2000)
7. Fisher, J.C., Pry, R.H.: A Simple Substitution Model of Technological Change. Technological Forecasting & Social Change 3(1) (1971)
8. Mayer, J.P., Davidson II, W.S.: Dissemination of innovation as social change of community psychology. In: Rappaport, J., Seidman, E. (eds.) Handbook of Community Psychology, xxi, 1011, pp. 421–443. Kluwer Academic Publishers, Dordrecht (2000)
9. Messnarz, R., et al.: ORGANIC - Continuous Organisational Learning in Innovation and Companies. In: Smith, B.S., Chiozza, E. (eds.) Proceedings of the E2005 Conference, E-Work and E-commerce, Novel Solutions for a Global Networked Economy. IOS Press, Amsterdam (2004)
10. Messnarz, R., Biró, M., Koinig, S., Reiner, M., Vajde-Horvat, R., Ekert, D.: The Future of SPI Knowledge and Networking in Europe – A Vision. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 268–277. Springer, Heidelberg (2011)

11. Neumann, M., Riel, A., Brissaud, D.: Improvement of Innovation Management through the Enlargement of Idea Sources. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 121–132. Springer, Heidelberg (2011)
12. Neumann, M., Riel, A., Brissaud, D.: Innovation Management System for the Automotive Supplier Industry to Drive Idea Generation and Product Innovation. In: Proceedings of the 17th EuroSPI Conference, vol. 17, pp. 7.13–7.24 (2010)
13. Neumann, M., Riel, A., Brissaud, D.: Improvement of Idea Generation by Capitalisation on Internal and External Stakeholders. In: Industrial Proceedings of the 18th EuroSPI Conference, vol. 18, pp. 3.25–3.33 (2011)
14. Pries-Heje, J., Johansen J.: SPI Manifesto. Version A.1.2.2010 (2010), http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf
15. Rogers, E.M.: Diffusion of Innovations. The Free Press, New York (1962)
16. Tidd, J., Bessant, J., Pavitt, K.: Managing Innovation, Integrating technological, market and Organisational Change, 2nd edn. John Wiley & Sons Ltd., West Sussex (2001)
17. Van de Ven, A.H., Polley, D.E., Garud, R., Venkataraman, S.: The innovation journey. Oxford University Press, New York (2008)
18. Bologna, http://ec.europa.eu/education/policies/educ/bologna/bologna_en.html (retrieved April 22, 2012)
19. Siakas, K., Georgiadou, E.: A Flexible Adaptable Internal Quality Assurance Framework for Higher Education: Designing Key Performance Indicators. In: Proceedings of International Conference (Tempus JEP-27178-2006) on Quality Enhancement: Experience, Challenges and Perspectives for Armenian Higher Education, Yerevan, October 5 and 6, pp. 190–203 (2009)
20. Neumann, M.: Presentation at the EuroSPI 2011 conference, Workshop about "Creating Environments Supporting Innovation and Improvement" (2011), http://2012.eurospi.net/images/Documents/eurpspi%20ws1%20neumann.pdf

EU Project SafEUr – Competence Requirements for Functional Safety Managers

Andreas Riel[1], Volker Ovi Bachmann[2], Klaudia Dussa-Zieger[3], Christian Kreiner[4], Richard Messnarz[5], Risto Nevalainen[6], Bernhard Sechser[3], and Serge Tichkiewitch[1]

[1] EMIRAcle c/o Grenoble Institute of Technology GSCOP UMR5272, Grenoble, France
{andreas.riel,serge.tichkiewitch}@grenoble-inp.fr
[2] SIBAC, Mittelbiberach, Germany
info@sibac.de
[3] Method Park Software AG, Erlangen, Germany
{klaudia.dussa-zieger,bernhard.sechser}@methodpark.de
[4] Graz University of Technology, Institute for Technical Informatics, Graz, Austria
christian.kreiner@tugraz.at
[5] ISCN GmbH, Graz, Austria
rmess@iscn.com
[6] Spinet Oy and FISMA, Finland
risto.nevalainen@falconleader.fi

Abstract. Functional Safety has become a vital property of many products and systems. There is a clear trend to move safety functions from pure mechanics into electronic control units. Therefore over the recent years, various standards have evolved describing the properties and certification criteria of safety-critical electronically controlled products and systems. In order for a particular product to be adapted to the requirements of different target markets, the integration of functional safety considerations according to the relevant standards into the complete product creation processes and organisations is essential. This integration requires special competencies that define the new job role of a Functional Safety Manager. This article gives an overview of these competency requirements in the context of the documentation of the first results, the targets, and the scope of the EU project SafEUr, which aims at putting in place a Europe-wide on-line training and certification program for functional safety managers according to the requirements of several industry sectors, as well as the experiences from research and practical safety engineering projects.

Keywords: Functional Safety Management, Integrated Safety Design, Training, Certification.

1 Introduction

The management of functional safety is directly linked with integrated design and engineering competencies. In case of the terrible bus accident in Switzerland in March 2012 where a bus crashed frontally into the side wall of a tunnel while the road and

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 253–265, 2012.

traffic conditions were excellent, only few functional safety experts might know that this could also happen if

- there is an unwanted actuation of the electronic steering system,
- there is an unwanted actuation of an ESP system,
- there is an unwanted actuation of a torque vectoring system,
- and more.

Nowadays more than 80% of the car functionality is controlled by electronic and software and for ASIL-A to ASIL-D classified hazards, which means that decisions made by electronics might have a fatal impact. A modern car has 60 to 120 Electronic Control Units (ECUs) inside which are connected by a real-time bus system. A car function is assigned to one or more ECUs, and manufacturers define a communication matrix between the ECUs.

Safety Engineering is a discipline which extends existing design engineering principles and adds concepts such as hazard analysis, signal flow based FMEA, FMEDA, failure tree analysis, functional safety concepts, technical safety concepts and technical safety architectures, independent control levels, and more.

Thus safety is more than just managing safety in a project, it is a complex integration of norm understanding, managing and planning a safety life cycle, integrated design including design parameters related to safety on all levels (product, system, software), and legal aspects to reduce potential liabilities.

Functional Product and System Safety is also inherently linked with the good knowledge of the respective standards and norms. Training and certification programs have been developed on national levels which teach the content of these standards. In particular, influential national certification institutions such as the German TÜV [1] offer such trainings, with big commercial success. The feedback on such training programs of a large international group of industrialist in the automotive, aerospace, and nuclear power domains from Austria, Germany, France, and Finland gave the impetus for the proposal of the EU-project SafEUr – ECQA Certified Functional Safety Manager – which has been accepted in November 2011. The core issues identified with this target group were the following:

1) Trainings on the standards and norms have a focus on teaching the contents of these, whereas there is a lack of enabling the trainees to apply these standards and norms to real practical engineering problems.
2) The implementation of functional safety on a system level requires measures on the process level, as well as the taking into account of the complete life cycle of the project. The associated management competences are not trained sufficiently in the existing trainings.
3) Modern systems are consisting of several different disciplines, typically integrating complex software systems into hardware systems, which are themselves composed of electric/electronic, mechanical/mechatronic), hydraulic, thermodynamic, etc. subsystems. Existing trainings fail to take into account the particular challenges that are rooted in the integration.

SafEUr aims at implementing exactly these three lacking subjects by complementing existing trainings on functional safety standards rather than by supplementing them. Therefore, the SafEUr training program focuses on the competencies required to manage the implementation and assurance of functional safety on the holistic system level taking into account the whole product/system life cycle.

This paper gives an overview of the SafEUr project, with particular focus on the initial version of the SafEUr Skill Card, i.e. the specification of competencies required for functional safety managers to get certified. Section 2 highlights some particular modern challenges in functional safety, without wanting to be exhaustive. Section 3 presents the context and the methodology, while section 4 gives an overview of the SafEUr Skill Card. Section 5 briefly describes the content and targets of each of the proposed training elements, thereby pointing out the essential competency requirements to Functional Safety Managers from the project consortium's point of view. Section 6 concludes giving an outlook on how the program will be implemented and deployed at European level.

2 Modern Challenges in Functional Safety

Over recent years, the assurance of the functional safety of products and systems has evolved from a purely prescriptive and normative approach towards a systematic way of the integration of developers' and operators' well reasoned arguments that prove that their systems achieve targeted safety levels. Such arguments and the associated evidences are called Safety Cases. While the prescriptive approach aims to achieve system safety through the satisfaction of regulatory requirements by e.g. construction codes, the safety case approach requires a deep knowledge of the product/system itself, as well as its use and behaviour during its entire life cycle. Despite the wide requirements for safety cases across many industries, it has been far from clear what constitutes a 'good' safety case, or how to construct a safety case. This problem is treated exhaustively in [2], where an attempt is made to clearly define what Safety Cases are, as well as to introduce a systematic approach to the management of such cases.

Another central notion of functional safety is the so-called Safety Integrity Level (SIL), which assess how critical safety is for a particular system. In the European Functional Safety standards based on the IEC 61508 standard [3], four SILs are defined, with SIL 4 being the most critical, and SIL 1 being the least critical level. In other words, the SIL is defined as the relative level of risk-reduction provided by a safety function. According to IEC 61508, the SIL concept must be related to the dangerous failure rate of a system, not just its failure rate or the failure rate of a component part, such as the software. A system-oriented approach to functional safety engineering is thus indispensible to assure a required SIL level.

In general, modern design processes have to tackle the challenge of taking into account both the safety-related and availability-related requirements to safety-critical systems in the complete design process. Ideally, they allow doing this simultaneously, as both requirements types usually are contrary to one another but at same time inherently relevant for success and failure domain analyses. Recent works such as [4] have investigated how this target can be achieved by capitalizing on existing well-established performance and safety design methods. The main difficulty here lies in assuring the implementation of functional product/system safety by the design process, which by itself has to comply with relevant standards on performance and quality. Companies are therefore also demanding for an integrated approach towards assessing process performance and process compliance to functional safety requirements [5].

Another issue intrinsically linked to the management of safety over the whole process is supplier management and subcontracting. Competence allocation and pricing are major drivers for subcontracting safety-relevant design, development, and maintenance tasks to third parties. The IEC 61508 group of standards require that all companies involved with functional safety have and are able to demonstrate Functional Safety Management. When subcontracting is applied on several levels of the supply chain, companies end up with complex supplier networks, where responsibilities quickly get unclear, and the propagation of accurate and original information becomes a real problem. Managing the associated issues goes far beyond pure safety engineering competencies. Some major associated challenges in five different industrial domains are investigated in [6].

In electronically controlled systems there is a trend towards implementing safety functionality that was formerly covered by non-electronic and non-software (i.e., mechanic, hydraulic, etc.) subsystems into electronics and software. At the same time, the possibility to monitor these subsystems with the support of software, and carry out targeted failure and failure prevention reactions, can increase the overall system availability significantly. These reasons lead to a permanently increasing complexity and volume of control and monitoring software, which demands the adaptation of existing traditional design processes in order for them to support the implementation of specific methods on a process and system level. Such methods and process requirements are contained e.g. in the new ISO 26262 standard for the functional safety of road vehicles [7]. Fulfilling such requirements in the context of design process standards applicable to the respective industry sector poses a lot of new challenges to designers and users of functional safety methods. A practical example for the automotive sector, where design process definitions according to ISO 15504 (SPICE) [8] have been widely adopted, is presented in [9]. A risk analysis methodology that can be applied in the early concept design phases of the mechatronic system design process is explained in [10]. Furthermore, software-intensive systems impose special requirements to the specification and use of safety cases, as pointed out in [11].

3 Methodology

SafEUr – ECQA Certified Functional Safety Manager – has been launched in November 2011 as a Multilateral Project for the Development of Innovation in the Leonardo da Vinci Programme of the European Commission by a consortium of six partners in Europe with vast practical experience in industrial functional safety management projects, and solid partnerships with OEMs and tier one suppliers in automotive, aerospace, and nuclear power plants.

The six consortium partners are the following (in alphabetical order):

1) EMIRAcle – European Manufacturing and Innovation Research Association, a cluster leading excellence, an international non-profit association of 25 leading research laboratories, companies and government institutions in 16 different countries, with its legal office in Belgium and its executive office in France;
2) Institute of Technical Informatics at Graz University of Technology in Austria, who is the project coordinator;
3) ISCN GmbH, International Software Consulting Network, based in Austria and in Ireland, who manages the project operatively;
4) Method Park Software AG in Germany, a company that actively supports the development and management of functional safety related products in the automotive and medical device industry, and delivers training and consulting for different safety standards like ISO 26262;
5) SIBAC GmbH in Germany, a company that has experience in the automotive industry in the development of safety related projects (automatic transmissions, electrically supported steering systems, active suspension systems) with all large suppliers and OEMs;
6) Spinet Oy in Finland, a micro-size company specialized in process assessment and improvement services and a partner in the European ARTEMIS project RECOMP [12], whose aim is to develop new methods and certification schemes for safety-critical systems.

In order to integrate the training and certification program in a Europe-wide accepted and promoted scheme, this consortium has partnered up with the European Certification and Qualification Association, ECQA [13]. The ECQA currently promotes more than 20 modern professions, which are all enabled by an integrated web-based environment for self assessment, e-Learning, and examination for certification.

The first part of the project is the specification of the competence set, from which e-Learning based training material in the five consortium languages and test questions in English for the European certification will be developed. In this paper, the authors focus on the presentation of the competence set as it has been specified by the consortium according to the standard that is proposed by the ECQA. This standard is compliant with the European Qualification Framework (EQF) [14], and is based on the concept that the skills which characterise 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 so-called 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.

The methodological approach in the project is to define the set of competencies in the form a Skill Card including content descriptions in a first step. This has been done by the partners on the basis of their profound experiences in industry co-operations on both project and process level. OEMs and tier one suppliers coming mainly from automotive industry and nuclear power plants have been closely involved in this process, in order to assure the high degree of relevance of the result.

The so-defined skill card provides the basis for the elaboration of the training material, which will be largely based on principles demonstrated by practical case studies in the form of e-Learning enabled presentation material. Based on the experiences of the consortium partners, and their contacts to industry, this material will be focussed on the norms IEC 61508, the machine standard, and ISO 26262. This set of training material will be validated in both on-site and on-line pilot trainings in different EU countries, which will be free of charge during the project duration. Feedback will be collected and used to improve and enrich the training material, as well as the skill card if necessary. In a second step, trainers will be trained in the participating consortium countries, in order to assure the sustainability after the project. At the same time, further pilot trainings will be organized, and the focus will be put on targeted dissemination actions to prepare the training exploitation phase after the project end in November 2013.

4 Functional Safety Manager Skill Card

The challenge for this program is to cover the competencies that are necessary for a Functional Safety Manager to have a wide view on which and how functional safety aspects have to be taken into account from the early design phases over the complete life cycle, yet a sufficiently profound expert understanding of functional safety engineering methods and tools.

The initial version of the Skill Set elaborated by the consortium is shown in Figure 1. For the sake of clarity, only the Skill Units (main branches) and Skill Elements (secondary branches) are presented. E-Learning suitable training material will be developed for each of these elements, with an emphasis on the clear separation of principles and practical case studies.

In order to achieve this, a set of success factors has been defined by leading European companies to be considered when applying Functional Safety in the context of the development an integrated ISO 15504 (SPICE) and safety assessment approach [5]. These companies also participated in the ECQA Certified Integrated Design Engineer (iDesigner) program [15], which essentially deals with the complex issues of the integration of multidisciplinary competencies in the design process of products, services, and systems. They are also part of the functional safety working group of the German SOQRATES initiative, where more than 20 suppliers and leading engineering companies and research organisations from Germany and Austria collaborate and exchange knowledge about practical implementation of ISO 15504, Automotive SPICE, IEC 61508 and ISO 26262. In addition, partners from the ISO 15504 Part 10 working party have been invited to contribute.

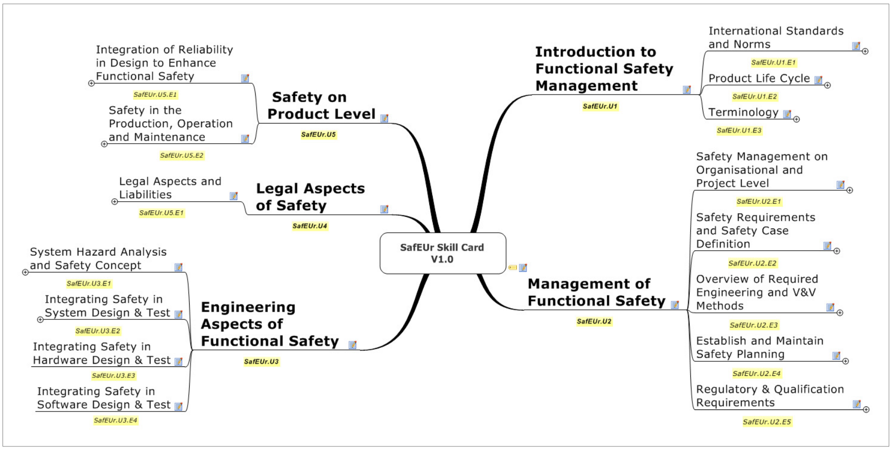


Fig. 1. SafEUr Skill Card showing Skill Units and Elements

The following section shortly describes each of the Skill Elements of the SafEUr Skill Card.

5 Functional Safety Manager Competency Requirements

5.1 Introduction to Functional Safety Management

The first training unit introduces the subject of Functional Safety, with a particular focus on issues related to the different existing safety standards. Basic knowledge about the complete product/system life cycle is also treated, with the aim to sensitise trainees for the importance of safety aspects in all life cycle phases.

International Standards and Norms

This element deals with international standards and norms related to functional safety. The market-specific certification of products and systems requires the choice of the appropriate applicable standard. Market-specific standard compliance, however, limits the accessibility of other markets, where different standards or and standard declinations may be applicable. For example, the IEC 61508 [3] group of safety standards is not specified in a way that makes it compatible with other safety standards, such as the DO-178b standard [16], which is specific to the aeronautic domain. The challenge is thus to take into account multiple standards in the product development and in the certification process. Functional Safety Managers are supposed to pick relevant standards, and to coordinate their integration in the complete product design process. To this aim, this element shall give trainees the ability to

a) be able to relate contemporary functional safety standards to the relevant industry sectors;
b) know the structures and content modules of the standards IEC 61508 and ISO 26262;

c) know the relationship between functional safety standards, norms, and process compliance;
d) know the relationship between functional safety standards, norms, and safety assessment and certification.

Product Life Cycle
This element introduces aspects of the product/system life cycle and their importance for functional safety engineering with respect to Supplier Qualification, Processes, Organisational Aspects, and Risk Management. Implementing safety in modern products and systems requires the integration of considerations about the specific safety requirements and constraints throughout all phases of the complete life cycle in the design. A detailed treatment of Integrated Engineering Design, also with a special look on functional safety issues, is promoted by the ECQA in the ECQA Certified Integrated Design Engineer Training and Certification program [17].

This element focuses on making trainees able to

a) know the key phases and stakeholders of a typical product/system life cycle;
b) know key safety-specific issues of the life cycle, such as safety case specification and supplier qualification;
c) think in terms of processes and their associated activities, roles, documentation, etc.
d) take into account organisational aspects, such as organisation structure and culture;
e) relate functional safety aspects to risk management.

Terminology
This element deals with aspects of Functional Safety Terminology, including terminology differences among standards. As the safety standards applying to different domains and markets have been created by domain experts, their nomenclatures are different. Safety Managers have to be able to carry out a semantic alignment of different standards relevant to a given system.

To this aim, this element aims at enabling trainees to

a) know key terms related to functional safety as they are used and defined in the standards;
b) be able to deal with differences in terminology among the different standards.

5.2 Management of Functional Safety

This unit investigates major management aspects of functional safety engineering on organisational and project level. The definition and management of so-called Safety Cases assumes a central role in the functional safety management activities, as safety cases are at the root of modern functional safety engineering methods.

Safety Management on Organisational and Project Level
This element addresses competencies needed to achieve independent organisational control, and to perform project specific work and responsibilities, as defined in processes covering the whole safety life cycle. It aims at enabling trainees to

a) recognise the importance of and requirements in safety culture to achieve systematic, high level safety awareness and responsibility in the whole organisation;
b) identify elements in organisational safety system management and assume the role of a safety manager;
c) define requirements for project / product / system level safety management;
d) establish the necessary safety and quality assurance for project / product / system level safety engineering work;
e) create and develop necessary documentation for organisational and project / product / system level safety management.

Safety Requirements and Safety Case Definition
This element addresses software and system safety classification schemes, as well as their requirements to collect evidences and to make necessary analyses to construct a full safety case. To this aim, it teaches how to

a) identity main elements of a safety case, based on standards and related concepts;
b) establish requirements for evidence collection to construct a full safety case;
c) create necessary arguments and modular safety cases;
d) explain a full safety case for organisational management and other stakeholders (customers, regulators, etc.);
e) review safety cases developed by suppliers or third parties.

Overview of Required Engineering and V&V Methods
This element investigates methods for engineering, validation and verification (V&V) that are required to implement functional safety on a project / product / system level. It aims at enabling trainees to be able to

a) select the right engineering and test approaches based on the provided method tables, the identified safety integrity level, and the product architecture;
b) set up a V&V Plan which covers all necessary test phases, test levels, test methods, test metrics, and evidences of complete functional safety coverage and compliance;
c) practically understand and implement safety related testing, such as fault injection testing, diagnostic coverage testing, equivalence class testing, load testing, branch coverage in testing, etc.
d) draw up a compliance map demonstrating the use of qualified tools and qualified engineering methods as part of the safety plan.

Establishing and Maintaining a Safety Plan
An essential element of introducing standard compliant function safety management is that the implementation of safety plans is monitored, and that the progress is tracked and reviewed. A safety plan documents the process used to analyse system safety and enumerates the mitigation techniques being used to ensure safe system operation. In this spirit, this element aims at making trainees able to

a) establish safety plans correctly;
b) monitor and review the progress of the implementation of such plans;
c) use safety plans as a tool for managing the function safety aspects of a development project.

Regulatory and Qualification Requirements
This element includes aspects of skills, responsibilities and communication capabilities among different stakeholders to be able to certify, qualify and/or licence safety-critical software intensive system. It teaches trainees to be able to

a) act as a responsible person and facilitator in his/her organisation, to support required certification and/or qualification tasks with other parties;
b) recognise and explain at least one certification and/or qualification scheme and its benefits;
c) participate and coordinate necessary data collection and evidences for certification and/or qualification;
d) review draft certification / qualification reports from independent parties;
e) communicate with regulatory body / bodies to satisfy their information needs in qualification and licensing.

5.3 Engineering Aspects of Functional Safety

This unit is the essential complement of Unit 2, i.e., the unit covering the management aspects of functional safety. Its main objective is to bridge the gap between the theoretical standards, and the practical implementation of the latter's rules and requirements. This is considered the main particularity that distinguishes SafEUr from comparable trainings in the same field.

System Hazard Analysis and Safety Concept
This element addresses the building blocks at the very basis of every functional safety engineering project: the identification of hazards, and the establishment of a safety concept. To achieve this, it teaches trainees to

a) understand the key vocabulary words to carry out a hazard and risk analysis;
b) describe the working environment and the item definition
c) understand the difference of functional and non-functional behaviour of the system,
d) be able to moderate a hazard and risk analysis in a development department,
e) be able to come to a correct assessment of the SIL or ASIL.

Integrating Safety in System Design & Test
This element looks at the integration of safety aspects in system design and test. To this aim, it enables trainees to

a) understand the difference between system requirements and system design;
b) explain signal paths in systems and their influence on the system,
c) show the allocation of subsystems to his systems requirements and system design;
d) describe a state machine on system level and allocate time slots for the subsystems on the safety critical path for the identified system reaction time.

Integrating Safety in Hardware Design & Test
This element focuses on hardware issues in system design and test. It aims at making trainees able to

a) explain the terms Failure, Fault, Error, together with Fault classes, Failure modes, and attributes of dependability (availability, reliability, safety, confidentiality, integrity, maintainability);
b) select the right strategy from a set of basic dependability strategies;
c) explain the basic terms of modelling hardware fault tolerance (hazard functions, MTTF, MTTR, MTBF, availability, maintainability) and select the right modelling strategy for hardware fault tolerance;
d) calculate the reliability of series, parallel, and mixed systems, as well as apply this theory to N-redundant systems.

Integrating Safety in Software Design & Test
This element focuses on software issues in system design and test. It aims at making trainees able to

a) explain Design Diversity strategies and select the right ones;
b) explain Data Diversity strategies and select the right ones;
c) explain and select the right fault tolerant software patterns (architectural, detection, error recovery, error mitigation, fault treatment) for a system to be designed;
d) select the right adjudication concept;
e) select the right Information Redundancy (codes).

5.4 Legal Aspects of Functional Safety

Legal Aspects and Liabilities
This element addresses the legal aspects of functional safety and the liability of the involved persons and organizations. It examines both the personal responsibility and the product liability. It includes aspects of how to interpret the requirements regarding functional safety in terms of personal responsibility as well as product related safety. It teaches trainees to

a) know the legal aspects of product liability (national and international);
b) know the personal responsibility not to harm any human being by developing defective products;
c) be able to estimate the residual risk of a product to be released;
d) know which kind of information must be provided in order to satisfy legal aspects.

5.5 Safety on Product Level

This Unit addresses reliability and safety engineering aspects for integrated product design, and covers as well the required safety control mechanisms in production and maintenance.

Integration of Reliability in Design to Enhance Functional Safety
This element includes aspects of how reliability engineering can be integrated into the design process. It deals with methods of establishing the links between top safety events and the design parameters of a given problem, as well as the link between the

modelling of the physical behaviour of the system and the design parameters. It teaches trainees to

a) define the chain of the reliability in design and the needed actors to intervene in this chain;
b) define the links between the top events and the elementary design parameters for a given problem;
c) know the link between the modelling of the physical behaviour of the system and the modelling of the design parameters.

Safety in the Production, Operation and Maintenance
This element includes safety aspects in production systems, adding to the current operative system the needed extensions to take into account the treatment of defaults without any damage. It proposes a methodology to get an overview about safe start and stop modes and safety control specifications. It presents how to

a) know the different states proposed in the guide for the study of the start and stop modes of a production system;
b) know how to build the specification for the safety control of a production system;
c) know the limits of the guide for the study of start and stop modes.

6 Conclusion and Outlook

From the authors' experiences in numerous functional safety engineering projects in different domains, as well as in the corresponding training market, there is currently no other comparable international training and certification program on functional safety management available. SafEUr therefore aims at filling 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. The main research issue in this training development program is the identification and elaboration of those issues in functional safety management which are the keys to bridging this gap. A strong associated consortium of industry partners in Europe assures the relevance of the results with current and future practical needs.

The project builds upon a solid, Europe-wide well-established e-Learning and certification platform promoted and maintained by the ECQA. As such, it will increase the latter's pool of Europe-wide certified training programs, and thus contribute to the further implementation of a European on-line Training and Certification Campus for engineers and managers assuming modern job roles in industry.

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-2011-AT-LEONARDO-LMP.

References

1. TÜV Rheinland Group, http://www.tuv.com (last accessed on April 11, 2012)
2. Kelly, T.: A Systematic Approach to Safety Case Management. Society of Automotive Engineers (SAE), paper No. 04AE-149, 10 pages (2003)
3. International Electrotechnical Commission (IEC): IEC 61508. Functional safety of electrical/electronic/programmable electronic (E/E/PE) safety related systems, Parts 1–7 (1998)
4. Ahmed, R., Mo Koo, J., Jeong, Y.H., Heo, G.: Design of safety-critical systems using the complementarities of success and failure domains with a case study. Reliability Engineering and System Safety 96, 201–209 (2011)
5. Messnarz, R., Ross, H.-L., Habel, S., König, F., Koundoussi, A., Unterreitmayer, J., Ekert, D.: Integrated Automotive SPICE and safety assessments. Software Process: Improvement and Practice (SPIP) 14(5), 279–288 (2009)
6. Marucheck, A., Greis, N., Mena, C., Linning, C.: Product safety and security in the global supply chain: Issues, challenges and research opportunities. Journal of Operations Management 29, 707–720 (2011)
7. International Organization for Standardization (ISO): ISO 26262. Road vehicles – Functional safety – Parts 1–9 (2011)
8. International Organization for Standardization (ISO): ISO 15504. Information technology – Process assessment – Parts 1–10 (2004)
9. 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)
10. Sierla, S., Tumer, I., Papakonstantinou, N., Koskinen, K., Jensen, D.: Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework. Mechatronics 22, 137–151 (2012)
11. Braun, P., Philipps, J., Schaetz, B., Wagner, S.: Model-Based Safety-Cases for Software-Intensive Systems. Electronic Notes in Theoretical Computer Science 238, 71–77 (2009)
12. RECOMP – Reduced Certification Costs Using Trusted Multi-core Platforms, http://atc.ugr.es/recomp/ (last accessed on April 18, 2012)
13. ECQA – European Certification and Qualification Association, http://www.ecqa.org (last accessed on April 11, 2012)
14. The European Qualifications Framework (EQF), http://ec.europa.eu/education/lifelong-learning-policy/doc44_en.htm (last accessed on November 22, 2010)
15. iDesigner – ECQA Certified Integrated Design Engineer, http://www.integrated-designer.eu (last accessed on April 11, 2012)
16. N.W.S., W.D., R.I., C.A.N.W.S, W.D., RTCA, Inc.: DO-178b: Software considerations in airborne systems and equipment certification (1992)
17. 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)

Experiences with Trial Assessments Combining Automotive SPICE and Functional Safety Standards

Richard Messnarz[1], Frank König[2], and Volker Ovi Bachmann[3]

[1] ISCN GesmbH,
Schieszstattgasse 4, A-8010 Graz, Austria
rmess@iscn.com
[2] ZF Friedrichshafen AG, Friedrichshafen,
Germany
[3] SIBAC GmbH, Mittelbiberach,
Germany

Abstract. In 2009 and 2010 [6], [7] papers were published at EuroSPI explaining how a task force of leading suppliers extended Automotive SPICE with additional practices to cover aspects of IEC 61508 and ISO 26262 as well. In 2011 [8] the partnership published at EuroSPI an example of how Automotive SPICE compliant engineering processes have been extended to cover functional safety architectures as well. This integrated assessment model has been used in 2011 in trial assessments at Tier 1 (leading Automotive) suppliers and in this paper we describe the lessons learned and the next steps the working group is taking in 2012.

Keywords: Automotive SPICE, Functional Safety Standards, Safety Requirements, Technical Safety Concept and Safety Architecture.

1 The Integrated Assessment Model Used

In [6] and [7] we described how we mapped Automotive SPICE towards IEC 61508 and ISO 26262.

During an assessment the "Functional Safety View" can be activated with the following effects:

- Base practices will have additional criteria.
- Generic practices include additional criteria.
- New safety practices will appear.
- A safety methods table per process can be opened to consider the use of methods when assessing the practices.

In the functional assessment view we ask about extended safety practices which relate to the previously described strategy of extending Automotive SPICE to cover safety as well.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 266–275, 2012.

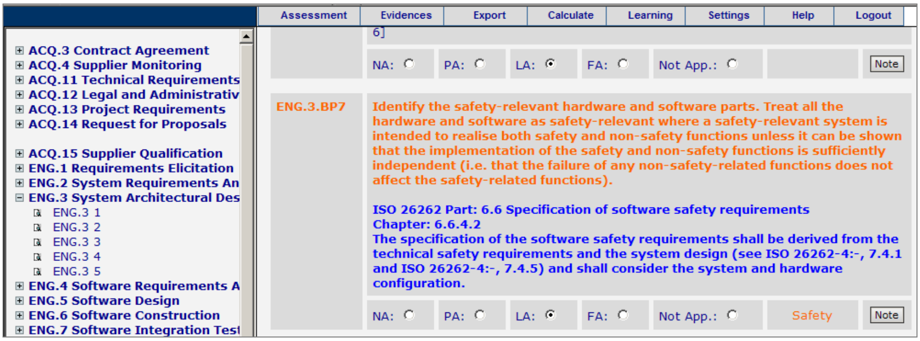


Fig. 1. Activated Functional Safety Views – ISO 26262 Extensions on Base Practice Level

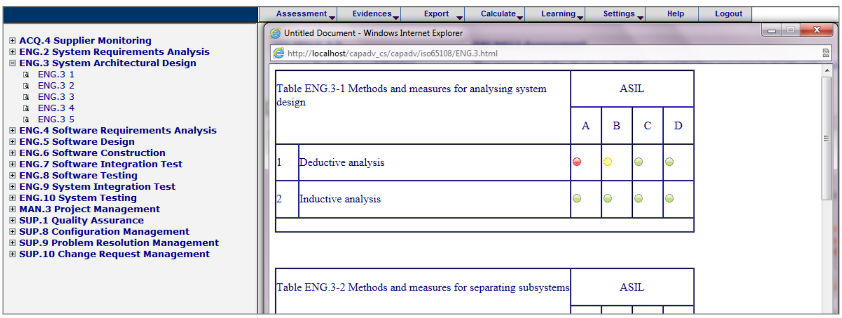


Fig. 2. Activated Functional Safety Views – Method Tables

For instance, in Figure 1 a new safety base practice ENG.3.BP7 has been added by additional requirements originating from IEC 61508 and ISO 26262.

The overall text for ENG.3.BP7 derives from general safety requirements from IEC 61508, while the specific element related text (reference ISO 26262 Part 6.6, chapter 6.6.4.2) describes specific safety requirements stemming from the Automotive specific functional safety standard ISO 26262.

The ISO 26262 part of the question asks about the aspect illustrated in Figure 6 where software safety requirements are derived from technical safety requirements. The IEC 61508 part of the question asks about the independence of the control function as described in the technical safety concept as part of the systems architecture in Figure 5.

All extended or new safety practices in the integrated safety and SPICE assessment are derived from such extensions needed to cover safety aspects in Automotive SPICE assessments as well. For further assessment approach details please read the articles [6], [7].

GPI 2.1.1 Identify the objectives for the performance of the process.

NOTE 1: Performance objectives may include: (1) quality of the artifacts produced, (2) process cycle time or frequency (3) resource usage and (4) boundaries of the process.
Performance objectives are identified based on process requirements.
The scope of the process performance is defined.
Assumptions and constraints are considered when identifying the performance objectives.
NOTE 2: At minimum, project performance objectives for resources, effort and schedule should be stated.

This includes objectives related with the coverage of system safety requirements.

ISO 26262 Part: 6.5 Initiation of product development at the software level
Chapter: 6.5.4.1
The planning of the software development includes the coordination with the product development at hardware level (see ISO 26262-5).

ISO 26262 Part: 6.7 Software architectural design
Chapter: 6.7.4.7
Safety-related software components that are reused without modifications or that are COTS products shall be qualified in accordance with ISO 26262-8:-, Clause 12.

ISO 26262 Part: 6.7 Software architectural design
Chapter: 6.7.4.8
Safety-related software components that are reused without modifications or that are COTS products shall be qualified in accordance with ISO 26262-8:-, Clause 12.

Fig. 3. Activated Functional Safety Views – ISO 26262 Extensions on Generic Practice Level

2 Reference Case for the Extension of Automotive SPICE with ISO 26262

The idea was to add further content and safety related design on the basis of the existing functional understanding and traceability mechanisms of Automotive SPICE [6], [7], [8],[9],[10].The Figures 4 to 6 illustrate the extensions based on an ASIL-D

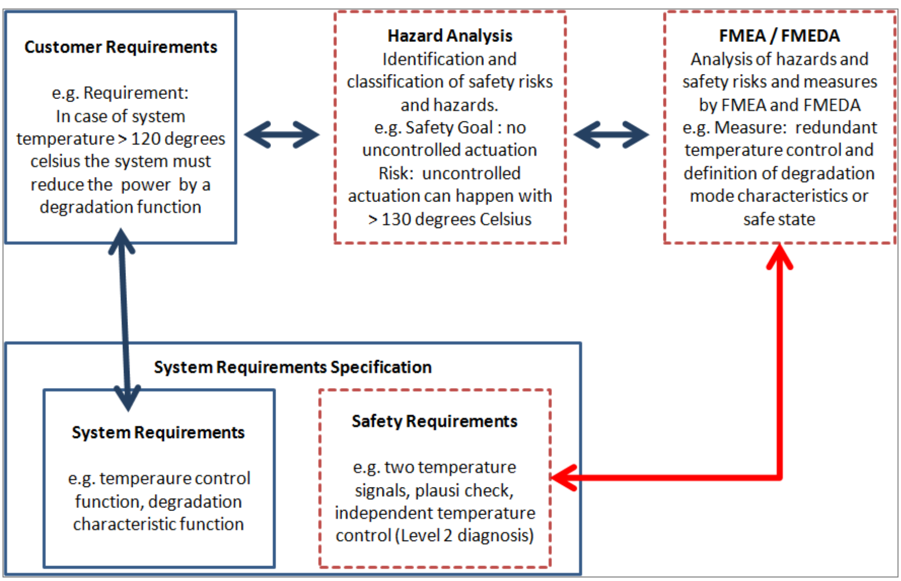


Fig. 4. Strategy Extensions in the process to Derive System Requirements from Customer Requirements

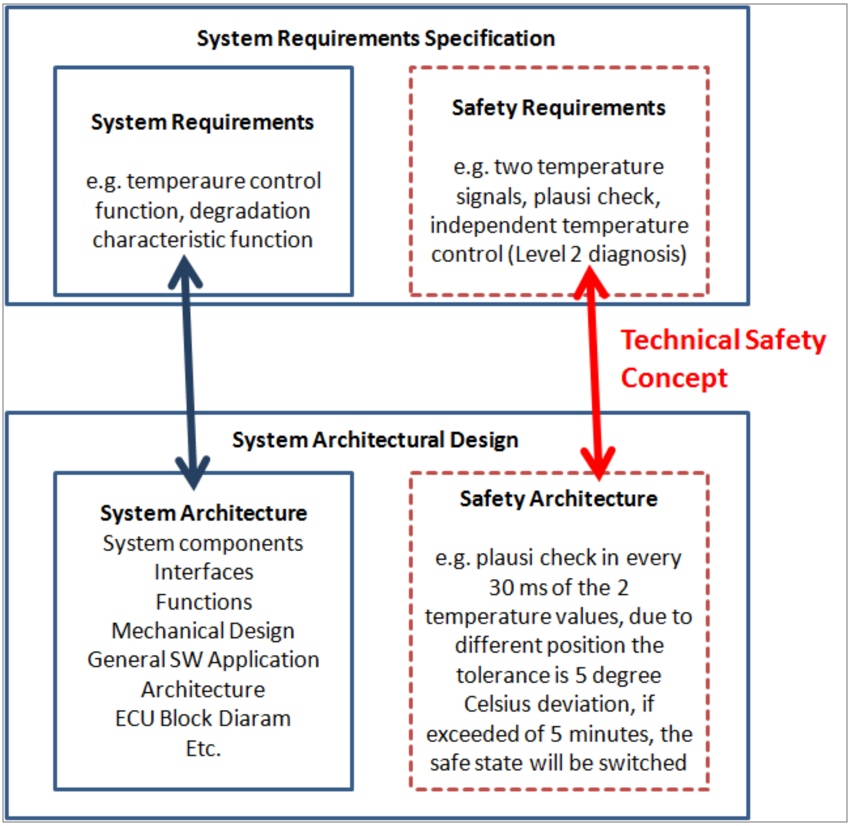


Fig. 5. Extensions in the process to Derive a System Architecture from System Requirements

classified example. The hazard is the unwanted actuation of a steering system, the FMEDA shows that this can happen at a too high temperature of the bridge on the ECU, and thus the control of the temperature is part of the safety critical signal flow.

The squares with dotted lines show the additional content requested by the safety standard which needs to be integrated into existing Automotive SPICE compliant engineering processes [8].

The system requirements specification would contain functional safety requirements (new in comparison to Automotive SPICE!), and functional safety requirements will be decomposed into technical safety requirements such as e.g. a plausibility check of the temperature running in an independent diagnosis control function. In the technical safety architecture (It is also part of the technical safety concept, and which is new in comparison to Automotive SPICE!) the details of the control cycle with plausibility check, tolerance values, will be designed, e.g. two analogue temperature sensors with one on the electrical circuit, one on the power amplifier component, a system evaluation that both positions allow a maximum difference of 5 °C, and a plausibility check which leads to a safe state, if the difference is above the tolerance threshold for more than a certain time.

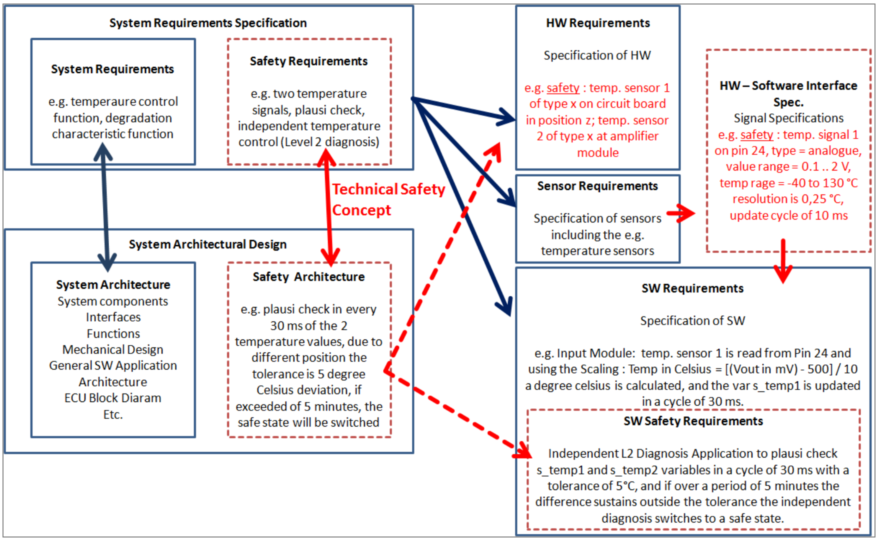


Fig. 6. Overall Picture of the Extensions in the Process to Derive Sub-System Requirements (see also Part 2)

The technical safety concept (Fig. 6) impacts the design of the subsystems. Especially the safety critical signal flows are analyzed and documented including new documentation such as a HSI (Hardware – Software – Interface – Spec.). Details were described in [8].

3 Trial Assessments

Case 1 - Automatic Gear System for High Speed Vehicles

The assessment was done for a leading customer and leading supplier in Automotive in an area of high speed car gear systems. The rpm was around 15000 which meant above 500 km/h speed.

The problem with nowadays automatic gear systems is that recently they were classified with an ASIL –D, the highest level of safety integrity in Automotive. The reason is that nowadays automatic gear systems can e.g. switch from the highest to the 1st gear while in the old hand based gear systems this was simply not possible.

The hazard is the uncontrolled deceleration of the car and the FMEA delivers a number of aspects about how to diagnose and control this situation [2].

This results in signal flow based architectural designs on system and software level to control the hazardous situation and switch to a safe state in case the hazardous situation is identified.

Figure 8 shows that a plausibility check needs to be done for the gear shift decision. Originally the gear shift decision is checked against the rpm of the output shaft. In the new model the speed is calculated from the wheel speed of the four wheels and also the rpm of the engine is received by CAN. Comparing the wheel speeds and the rpm engine a validated speed can be calculated. Then first the gear is calculated based on the old model (rpm output shaft), another gear decision calculation is done in parallel based on the speed and rpm engine, in a plausibility check the two gear decisions are compared and if they are not the same, the system will switch to a safe state.

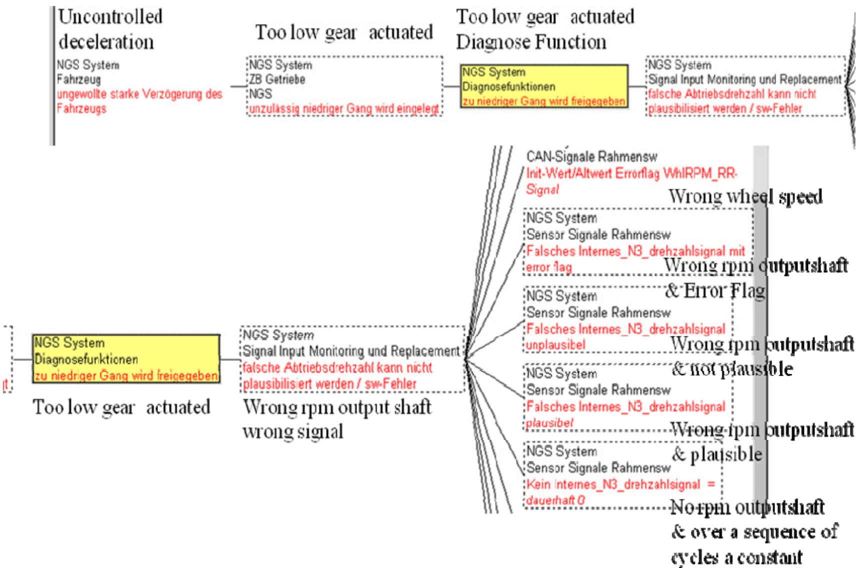


Fig. 7. Results of FMEDA / Signal Flow FMEA – Wrong Gear Actuated

This describes by far not all necessary actions to implement the safety flow, some more actions are e.g.:

- Assuring that the calculations by the processor are right (Watchdog)
- Assure that the normal functional code does not interfere with the safety code (memory layout and independent storage)
- Assure that the functional flow (call sequence of modules) is correct by a central state machine.
- Assure the fit rate / reliability of hardware along the safety critical signal paths
- And many more.

In the Automotive SPICE Assessment this additional safety understanding leads to a number of additional questions.

- The system requirements analysis will include questions about the functional and technical safety requirements.
- The system and software architectural design will include questions about the signal flow designs and control paths.
- In all engineering processes we check the method tables.
- And so forth.

In this assessment the assessors were Automotive SPICE assessors with a background in functional design and who collaborated in a functional safety working party for about 4 years already. In this case the Automotive SPICE assessors merge both worlds, SPICE and functional safety.

The assessment produced both, an Automotive SPICE based capability level profile as well as a deviation record for the parts 4 and 6 of ISO 26262. See example Figure 9.

An interesting observation was that the “traceability” required by Automotive SPICE can lead to findings in assessments which help to achieve an overall improvement for the product. E.g. the product was specified for around 15000 rpm, the system spec contained x < 15000 rpm, ad the base software was configured for y < x rpm. This meant that linking this function illustrated that software avoided a full actuation of the system which means less speed (in case of a sport motor).

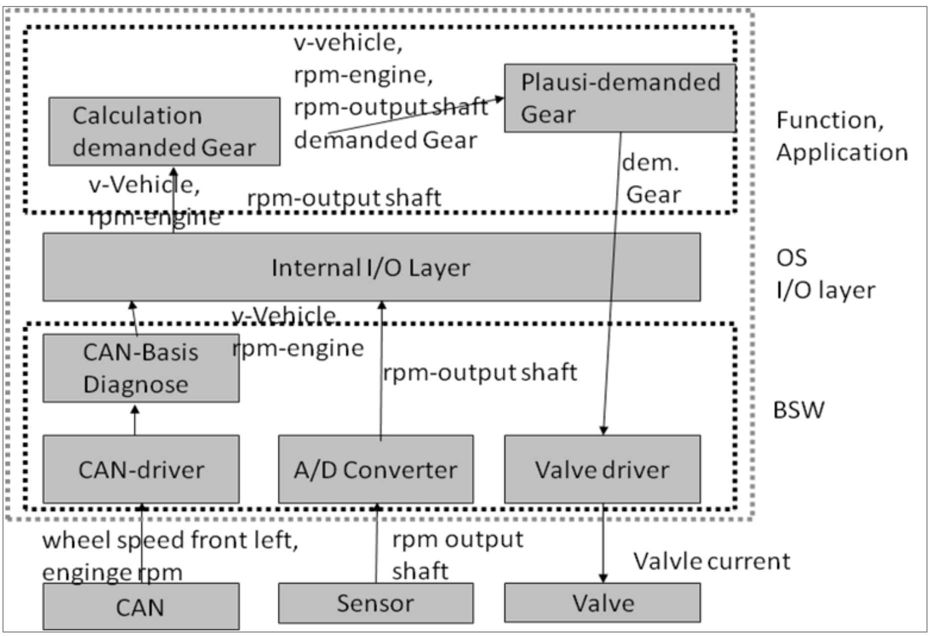


Fig. 8. Signal Flow Based Plausibility Functions

Case 2 - Automatic Gear System for Heavy Trucks

The assessment was done for a leading supplier in Automotive in cooperation with TÜV. The product is an automatic gear system with up to 16 gears and 40t vehicle weight. The project produced a base transmission for a number of trucks where the system can be adapted by application parameters. The software is re-usable across a number of projects. The project had a very experienced project and technical lead and had a systems architect with a long standing engineering experience. The systems architect had already incorporated the reference architecture [8] in his work.

The goal of the assessment was that

1. The certification body TÜV participates and can in parallel complete the own safety checklist.
2. With one SPICE assessment and additional safety questions there is only a one time the effort for the project.

The concepts about control flows (Figures 7,8) applied here as well.

The idea was that the certifier TÜV only asks what has noat been covered by the integrated Automotive SPICE and Safety Assessment already. These questions were to be asked after the assessment.

Assessment	Process	Element	Practice	Assessor	Safety Score	Comment	ISO 26262 Part	ISO 26262 Chapter	ISO 26262 Content
AssName	System Requ	ENG.2 1	ENG.2.BP1 I	Ovi Bachmar	LA	KLH Kompon	Part: 6.6 Specif	Chapter: 6.6.4.4	If other funct
AssName	System Requ	ENG.2 1	ENG.2.BP1 I	Richard Mes	FA	Das KLH (Kur	Part: 6.6 Specif	Chapter: 6.6.4.4	If other funct
AssName	System Requ	ENG.2 1	ENG.2.BP2 A	Ovi Bachmar	FA	Im KLH sind zunächst alle Funktionalen Anforderungen enthal			
AssName	System Requ	ENG.2 1	ENG.2.BP2 A	Richard Mes	FA	Die Kundenanforderungen im DOORS sind ResposnibleGroup			
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Ovi Bachmar	FA	Wer von ein	Part: 6.6 Specif	Chapter: 6.6.4.2	The specifica
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Ovi Bachmar	FA	Wer von ein	Part: 6.6 Specif	Chapter: 6.6.4.2	The specifica
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Ovi Bachmar	FA	Wer von ein	Part: 6.6 Specif	Chapter: 6.6.4.3	The hardware
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Richard Mes	LA	Es gibt eine S	Part: 6.6 Specif	Chapter: 6.6.4.2	The specifica
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Richard Mes	LA	Es gibt eine S	Part: 6.6 Specif	Chapter: 6.6.4.2	The specifica
AssName	System Requ	ENG.2 1	ENG.2.BP3 D	Richard Mes	LA	Es gibt eine S	Part: 6.6 Specif	Chapter: 6.6.4.3	The hardware
AssName	System Requ	ENG.2 1	ENG.2.BP6 E	Ovi Bachmar	LA	Wird dadurch sichergestellt, dass das KLH als Pflichtenheft in			
AssName	System Requ	ENG.2 1	ENG.2.BP6 E	Richard Mes	LA	Zum Kunden hin ist das klar, da im DOORS ein Modul als Komb			
AssName	System Requ	ENG.2 2	ENG.2 2.1.1	Ovi Bachmar	LA	Systemanfor	Part: 6.5 Initiat	Chapter: 6.5.4.1	The planning
AssName	System Requ	ENG.2 2	ENG.2 2.1.1	Ovi Bachmar	LA	Systemanfor	Part: 6.7 Softw	Chapter: 6.7.4.7	Safety-relate
AssName	System Requ	ENG.2 2	ENG.2 2.1.1	Ovi Bachmar	LA	Systemanfor	Part: 6.7 Softw	Chapter: 6.7.4.8	Safety-relate
AssName	System Requ	ENG.2 2	ENG.2 2.1.1	Ovi Bachmar	LA	Systemanfor	Part: 6.7 Softw	Chapter: 6.7.4.10	If the embed

Fig. 9. ISO 26262 Deviation Record – Extra Output from SPICE Assessment

4 Lessons Learned

Case 1 - Automatic Gear System for High Speed Vehicles

The assessment based on an extended HIS scope needed 5 instead of 4 days. The assessors needed to go very deep into the details because safety also requires to audit specific architectures, functions, and methods applied. The effort for reporting was the same as with Automotive SPICE, only the type of report was Excel and not word because we needed to be able to filter based on Automotive SPICE processes or ISO 26262 elements at the same time.

The biggest problem was that the additional content displayed in the functional safety view in the assessment tool was very hard to interpret without very detailed understanding of the functional safety standards. A normal SPICE assessor with no safety background would be overwhelmed.

Case 2 - Automatic Gear System for Heavy Trucks

The assessment based on an extended HIS scope needed 5 instead of 4 days. The assessors needed to go very deep into the details because safety also requires to audit specific architectures, functions, and methods applied. The cooperation with the certifier TÜV worked very well. They stated that about 90% of their checklist was covered in these additional checks during the Automotive SPICE assessments.

The report was done as a normal word based Automotive SPICE report and the TÜV delivered their checklist in parallel.

The biggest problem was the same as in case 1.

5 Conclusions

At the moment the safety working group is re-defining the integrated Automotive SPICE and Safety assessment concept so that the content becomes more clear so that more assessors and safety experts can understand the model and do integrated assessments.

A new release of the assessment system is planned for autumn 2012. If you plan to join the SOQRATES working party please contact the author Dr Richard Messnarz, rmess@iscn.com.

Acknowledgements & SOQRATES Membership. We are grateful to the experts who have contributed to the Design AK and Safety AK: O. Bachmann (SIBAC), S. Habel, Leo Ross, I. Sokic (CONTINENTAL), F. König, A. Koundoussi, Heidi Galle (ZF), K. Dussa-Zieger (Methodpark), A. Riel (Grenoble INP), Alexander Much (Elektrobit), J. Unterreitmayer (SQS), D. Ekert, and R. Messnarz (ISCN).

References

1. Bachmann, V., Messnarz, R.: Improving the Software-Development for multiple Projects by applying a Platform Strategy for Mechatronic Systems. In: Proceedings of the EuroSPI 2009 Conference, Wiley SPIP Journal (August 2010)
2. 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)
3. Messnarz, R., et al.: Assessment Based Learning Centers. In: Proceedings of the EuroSPI 2006 Conference, Joensuu, Finland (October 2006); also published in Wiley SPIP Journal (Software Process Improvement in Practice) 12(6), 505–610 (2007)
4. Messnarz, R., et al.: Better Software Practice for Business Benefit. IEEE Computer Society Press, Washington (1999)
5. Messnarz, R.: From process improvement to learning organisations (pp. 287–294). Wiley SPIP Journal (Software Process Improvement in Practice), 11(3), 213–335 (2006)
6. 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)
7. SOQRATES Safety Team, Messnarz, R., Ross, H.-L., Habel, S., König, F., Koundoussi, A., Unterreitmayer, J., Ekert, D.: Integrated Approach for Automotive SPICE and ISO 26262 Assessments. In: Proceedings of the EuroSPI 2009 Conference, Wiley SPIP Journal (August 2010)

8. Messnarz, R., Sokic, I., Habel, S., König, F., Bachmann, O.: Extending Automotive SPICE to Cover Functional Safety Requirements and a Safety Architecture. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 298–307. Springer, Heidelberg (2011)
9. Automotive SPICE, http://www.automotivespice.com
10. ISO 26262, Road vehicles — Functional safety
11. SOQRATES Initiative, http://www.sowrates.de
12. HIS, http://www.his-automotive.de

A Way to Support SPI Strategy through CertiBPM Training and Certification Program in Romania

Anca Draghici, George Draghici, Cristian Olariu, and Alexandru Canda

"Politehnica" University of Timisoara, Romania
{adraghici,gdraghici}@eng.upt.ro,
{olariu_cristian83,alexandru.canda}@yahoo.com

Abstract. Through this article we intent to demonstrate how a qualification and certification program, like CertiBPM, developed and implemented with the support of European Certification and Qualification Association (ECQA), can be use to better support the SPI efforts and strategies into Romanian companies. The research results show only the preliminary opinions (collected through the trainees/employees feed-back) about the SPI efforts and how could the knowledge pool generated by the CertiBPM training program, to support these efforts. First, there will be explained briefly the SPI concept and its impact upon the modern companies. Also, the SPI Manifesto initiative will be described as a good practice, a Community of Practice that have been developed in relation with the EuroSPI series of conferences of ECQA, to support SPI general trend and standard implementation. The context of this paper is defined by the CertiBPM project activities of research and development of a qualification and certification program using the ECQA schema and guidelines. Feed-back was collected (after each training unit, element) after 6 days of training sessions in Business Process Management field, with a large group of employees and students (67 people trained) from the West Region of Romania. The debates and employees answers analysis allow us to make an inventory and to describe the initiated process improvement (SPI efforts and strategies) in important companies from the West Region of Romania. The presented paper is linked with the research activities of the project: CertiBPM - Certified Business Process Manager LLP-LdV/TOI/10/RO/010, founded with support from the European Commission. This paper and communication 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.

Keywords: SPI Effort, SPI Strategy, SPI Manifesto, Business Process Management, Marketing Survey, Training, Certification.

1 Introduction

In the knowledge base society, the use of information and communication technologies becomes a necessity both for professional and social, entertainment purposes. To support the massive computerization increasing, the need for on-time, cost effective and high quality software is ever increasing. In addition, software re-engineering

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 276–287, 2012.

exercises are being actively pursued in enterprises of all magnitudes worldwide. The software process improvement (SPI) program is among one of the contemporary approaches commonly adopted for rationalizing the capability and performance of software development and maintenance in an enterprise [1].

Today companies are challenged with progressively complex business processes that represent opportunities and/or threats and they require agility increasing. Enterprise process improvement is an achieved goal if an ongoing process improvement program is followed but also, refinement and up-date by considering the new turbulent global market conditions. Extensive experience and expertise in achieving institutionalization of process improvements are desirable pre-condition to gain/generate a positive impact in any organization. Sometime, companies built up specific improvement capabilities to better optimize their business process. Such capability is linked with issues as: process assessment, process design and deployment, process knowledge management, quality management, systems and software quantitative solutions etc. When growing their businesses, organizations (in the field of software development or those that are based on complex Enterprise Resources Planning solutions, such as SAP or having Knowledge Management Systems) should not only put effort into developing and executing their business strategies, but also into managing and improving their internal software development processes and aligning them with business growth strategies. It is only in this way they may confirm that their businesses grow in a healthy and sustainable way [2], [3].

In organizations that operate with a Business Process Management (BPM) system as an integral part of their mission and objectives, it becomes clear that they are focused on targets for process improvement, but this obvious link has still not being explored explicitly. From the an empirical perspective, it has been found that continuous improvement activities can be a key source of environmental improvement [4] by taking into consideration the social responsibility framework, too.

BPM represents a holistic management approach (or a modern management system) based on continuous improvement processes, change management theories, and support by modern information and communication technologies/systems. This approach aim to attend an optimal balance between organization’s external environment dynamics and its internal processes functionalities based on innovation, flexibility and agility [5]. In additional, BPM represent and it is also associated with a complex project related to continuous improvement and adaptation of the organizations relations from the perspective of the internal and external environment.

The present articles objective is to present the effects generated by the *CertiBPM - Certified Business Process Manager* (LLP-LdV/TOI/10/RO/010) - CertiBPM training program in Romania. The main focus of the debates will be on demonstrating the Romanian companies System, Services, Software Process Improvement. The associated research scenario is linked with the feed-back collection and process, from the trainees (employees from important companies from the West Region of Romania) that were involved in CertiBPM training programs on March 2012. After this brief introduction, the paper is deployed in three parts: (1) a short description of the CertiBPM project that define the motivation and the source of the process improvement; (2) The marketing research methodology and results that make an inventory of the initiated process improvement in important companies from the West Region of Romania; (3) Conclusions and lessons learned.

2 The CertiBPM Project – SPI Motivation and Source

The motivation for the CertiBPM project lies on the Romanian market training needs satisfaction in BPM. The competencies improving processes for the employees of Romanian companies will have a positive impact upon their results. One of the steps to this improvement is to educate employees, industry representatives, future employees (master and PhD students) in the BPM field of knowledge. Various BPM courses, books and literature from different vendors exist, which show different aspects BPM, but there is no coherent course for the specific Romanian market. The proposed approach is related to the transfer of innovation process from Slovenia and Austria to Romania. It consists of: Transfer of Education, Training and Certification concept of ECQA into the university and manufacturing domain in Romania. Aspects of the project impact are: CertiBPM will build on BPM knowledge and competencies in different industrial sectors, which are represented by the consortium members due to their activities with national and international enterprises; transfer of the ECQA concept and platform into the Romania; implementing the CertiBPM job-role and examination committees for certifying training bodies and exam tools in multilingual environment; the important knowledge of CertiBPM and the important system of an uniform European certification will be very useful for Romanian manufacturing industry, especially for automotive and telecommunication industries [6]. The core of the results envisaged is a skill set which clearly fit the competencies required for becoming a BPM specialist (basic level and advanced level). For all the skill elements training material will be provided in several languages (English, Slovenian, Romanian, and German) and will be upload on an e-learning system. A pool of test questions will be defined, which provides the basis for the trainees' certification process.

The CertiBPM qualification and certification addresses itself to employees from companies departments related to: Quality management (as TQM), CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), SCM (Supply Chain Management), and enterprise information system's specialists etc., who want to complement and/or certify their advanced BPM skills. The target group typically has abilities for self-development and self-learning, creativity, innovative initiatives etc. One of the biggest challenges is to conceive a training program that covers the complete skills set that better satisfy the target group specific needs.

The original training program and materials have been developed trough creative, synergetic energy of the project members' interactions (during face-to-face project meetings, and virtual meeting using Skype conference facilities, from December 2010 till December 2011). Training Material version 2012 (Table 1) was developed within the international consortium of the project.

The CertiBPM training materials content and structure were elaborated with respect and according to the guidelines and requirements of the European Certification and Qualification Association (ECQA, www.ecqa.org). According to these requirements, issues that are mandatory for defining a new profession are: the skill set description, the learning units and elements, and the performance criteria for each learning element (this is a typical tree structure of the training program). This basic structure is shown in Figure 3 (after the methodology described in [7]). If this structure is well define and the units and elements are well described and consistent in relation with the performance criteria, the profession is recognized by the ECQA (the

Job Role Committee – usually representatives from a Leonardo da Vinci (that assure the financial support for this development) project partners - that join together specialists and personalities in the specific professional field is also recognized by ECQA as a valid body) and they could support the examination process in order to generate ECQA certificates for the corresponding profession (the Job Role Committee can start prepare the examination question pool that is related to the corresponding performance criteria, in accordance with the ECQA Examination Guide) [8].

Table 1. The CertiBPM Training/Qualification Program – Learning Units and Elements

Units and Elements Code	Learning Units and Elements Title	No. of Performance Criteria
BPM.U0	**Informative package**	5
BPM.U0.E0	Introduction	2
BPM.U0.E1	ECQA and certification information (demonstration on www.ecqa.org and the Learning Portal)	2
BPM.U0.E2	Conclusions and References data base	1
BPM.U1	**Process Oriented Management**	24
BPM.U1.E1	Management System (ISO 9001:2008)	5
BPM.U1.E2	Managing BPM projects	5
BPM.U1.E3	BPM and Modeling	5
BPM.U1.E4	Documenting Business Process	4
BPM.U1.E5	Process Simulation & Analysis basics, tools and techniques	5
BPM.U2	**BPM and Information Technologies (IT)**	16
BPM.U2.E1	Choosing your BPM tool/platform	5
BPM.U2.E2	BPM tools/platform	7
BPM.U2.E3	BPM and Enterprise architecture	2
BPM.U2.E4	BPM systems and IT integration	2
BPM.U3	**BPM human aspects. Frameworks and Standards**	12
BPM.U3.E1	Human factors in BPM	4
BPM.U3.E2	Motivating people for process change	3
BPM.U3.E3	BPM models, frameworks and standards	5
BPM.U4	**BPM Specializations**	-
BPM.U4.E1	BP manager for IT processes	-
BPM.U4.E2	BP manager for core sales & marketing processes	-

The examination questions (for the on-line examination procedure by login on the www.ecqa.org, section: *Certification and Examination – Exam Registration*) are developed as multiple choice one (maximum 4 types of answers). The examination process has to be very well understood for the participant with respect to the ECQA procedure. The following rules have to be applied [9], [10]:

- The start and end time of the exam is set by the exam organization (after the agreement of the members of the Job Role Committee that have contributed to the development of the training materials and that were involved in the organization of some training sessions). All the participants at an examination process have to be registered on the web portal. The user can attend the exam at any time after the examination period has started and before it will end.
- During the exam, the participant can log in and log out from the system. The answers are stored in the database. If an internet connection break occurs (the wireless connection drops, the internet does not work etc.), a warning message will be displayed. The user has to log out. After the connection has been re-established, the user can log in and has to re-check if his/her last answers have been correctly saved. In case any problems occur, the person supervising the exam should be informed.
- The results of the exam are displayed after the exam is closed by the exam organization. If the user finishes the test earlier, he/she will not be able to view the results. The results are calculated and displayed per learning element. To pass a learning element, at least 66% coverage of the respective learning element is required.
- 10 random questions are selected out of each learning element. Each question has at least one correct answer. If the participant selects a wrong answer, the question is scored zero (0%). All questions are equally weighted; the results are calculated with the average algorithm (total scores of all questions per element / number of questions per element).

After this brief explanation of the CertiBPM training, qualification strategy and content, together with the examination, certification procedure we intent to describe the marketing research methodology and results that make an inventory of the initiated process improvement in important companies from the West Region of Romania.

3 Marketing Survey – CertiBPM Impact on SPI Politics and Strategies in Romanian Companies

3.1 The Research Context and Design

This marketing survey was organized and develops in the end of each CertiBPM training sessions that take place in March 2012 (two training sessions, six days of in-class trainings and some webinars on specific important issues). The objectives of each training session (workshops with the formed target groups) are related to the training needs that were identified [11]. *Project managers* (as project managers, CIOs, quality managers, organization managers, process owners, unit leaders, business analysts) wish to: document business processes; model business processes; manage business process projects; change the way their employees work and implement some specific policies and the strategy in the SPI field. *Users of different BPM solutions* (as employers trainers, quality supervisors, IT supervisors, administrators or consulting, human resources managers) implemented in companies want to: better understand the users' perceptions and to correct their misunderstandings by improving the software solutions; avoid misunderstandings and develop better user's guides; find the best support solution for the change management process when the BPM software solution is implemented or updated or when they are confronted with new users (accelerate the learning process). They want to know how specific SPI strategies will be effect by the BPM system implemented and who they could help and better support the change strategy related to SPI. *PhD and master students* want to understand the BPM process in its complexity and to develop studies, researches regarding different issues related to using, exploiting and optimizing the software solutions implemented in different companies. Also, from their perspective as future users of a specific BPM solution or system, they are interested in getting familiar with the concept and with its main functionalities.

The structure of the training group (trainees target group from the West Region of Romania, participants at the CertiBPM training program, Table 2), together with some feed-back impressions regarding the pedagogical method used for in-class and on-line training and the content of the CertiBPM training program, are presented in Figure 1. The group of trainees is representative for the Western Region of Romania, if the economic specificity is taken into consideration. After each training unit (during each training session) the feed-back was collected with the purpose of: (1) identifying what trainees like and do not like during the in-class interaction regarding: the trainer presentation style and his/her rhythms of training; (2) the content of the element that was taught and the usefulness of the knowledge gained by the trainees when they go to apply or to exploit these knowledge in their companies or organization, including their impact upon the SPI politics and strategies.

For these reasons a focus group (qualitative marketing research) was initiated in the end of each unit training session and the trainees were asked to express their opinions. The acceptance and the utility of the CertiBPM training program has been shown in Figure 1 second graphic, but for the CertiBPM knowledge impact upon the SPI politics and strategies in different trainees companies, the Mintzberg model ([12], followed by Christenson [13]) have been used to collect information (Table 3).

Table 2. The Group/Sample Structure (per type of companies)

Type of company in the sample	No. of companies / %	No. of trainees / %
Automotive	8 / 36.36%	38 / 56.50%
Logistics	6 / 27.27%	6 / 8.75%
Software development	4 / 18.18%	7 / 10.25%
University (master and PhD. students)	1 / 4.54%	10 / 15.50%
Other industries	3 / 13.65%	6 / 9%
TOTAL	22 / 100%	67 / 100%

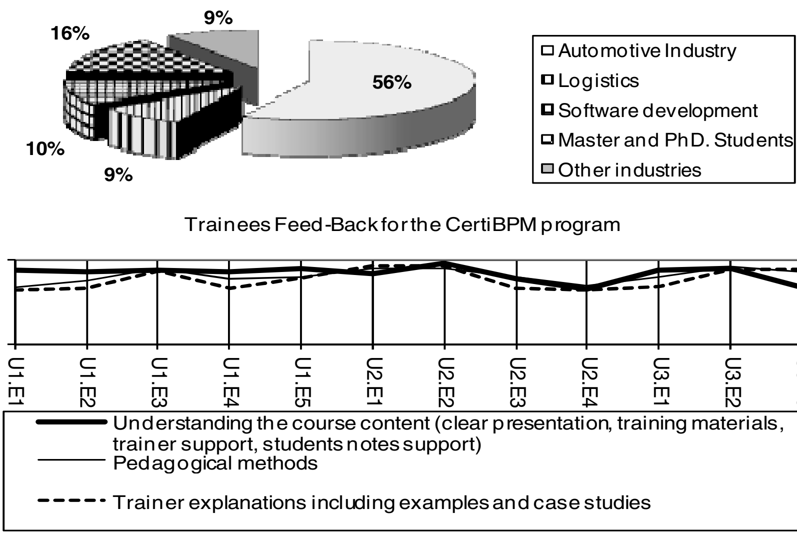


Fig. 1. The Target Group and their Feed-Back

Table 3. Mintzberg Critical Elements as Criteria for Effective SPI Strategy

No.	The Critical Element	The Key Questions
1	Clear, decisive objectives	Are all efforts directed towards clearly understood, decisive and attainable overall goals?
2	Maintaining the initiative	Does the strategy preserve freedom of action and enhance commitment?
3	Concentration	Does the strategy concentrate superior power at the place and time likely to the decisive?
4	Flexibility	Has the strategy purposely built in resource buffers and dimensions for flexibility and man-oeuvre?
5	Coordinated and committed leadership	Does the strategy provide responsible, committed leadership for each of its major goals?
6	Surprise	Has the strategy made use of speed, secrecy and intelligence to attack exposed or unprepared opponents at unexpected times?
7	Security	Does the strategy secure resources based and all vital operating points for the enterprise?

3.2 The Survey Results

The survey results are linked with the main CertiBPM training program teaching subjects that were the most attractive because of their practical implication (employees were interested to up-grade their knowledge in these particular subjects and they rise-up aspects to be optimize related to this subjects): (a) Management system (basic from

ISI 9000); (b) BP definition and description (documentation and specific software tool using for modeling and simulation); (c) Human resources aspects (motivating people for the BPM implementation process, change management, frameworks for BPM). In relation with these subjects the SPI effort and strategy particularities have been debated.

The focus group debates has underlined that *approaching SPI as a change management process*, makes it clear that a SPI effort has the main characteristic features of a change process, through which the practice of software work is object for change. Trainees expressed that the focus of improvements should be moved from *resources* to *process,* and then to *practice* and the improvement efforts should be organized as a *change management program* rather than a process improvement project. The trainees and trainers debates conclude that in a SPI program more attention has to be given on defining expected effects, planning, organizing and managing changing behaviors and knowledge rather than only improving processes.

From the trainees in-put, trainers have understood that in Romanian companies, an SPI effort in itself can be optimize from the perspective of an *organized and planned effort* which is based on gathering feedback information about the processes from the field of practices (based on the define, model, analysis and simulation of specific processes, with the available tools introduce by the CertiBPM training program).

SPI can further be seen as an incremental based change process rather than a revolutionary one. A SPI effort does not happen in an ad-hoc way (as it is seen in some Romanian companies but not in the case of multinational one). It is not an expert oriented effort; it is rather based on practitioners' ideas and ideals (most of them imported from other culture as the case of multinational companies). SPI should not be pushed into the organization (the change management support is vital for the implementation phase). The change process in SPI is a combination of stepwise, planned, organized and controlled effort that is built based on practitioners' commitment and ideas. But, still in many Romanian companies, SPI process is mainly focus on processes and very little on change strategy. A typical experience in Romanian companies, the trainees shows that SPI effort starts with an assessment (often a Capability Maturity Model - based) to establish the current maturity level of the organization. Here the focus is on establishing the maturity level of the software processes in the organization.

Romanian trainees explain that approaching SPI as a change program requires additional assessment to find out the maturity level of peoples' disposition to change in order to understand and establish the maturity of peoples' ability, willingness and openness to change. People in all levels: management, project managers, and software engineering staff have to be motivated to align their actions (behavior) to support the specific SPI strategy (in strong relation with BPM approach). Romanian employee involved in the CertiBPM training program debate that an important step in an SPI effort is focused on identifying *which processes to improve*, when to do what and assigning people to the different process improvement tasks needed to be done (that have to be demonstrate and support by the attractiveness of the Process Model and Simulation training activities and also by the attractiveness of the U3 - human aspects, frameworks and standards).

Trainees agree that, this effort will be structured through a SPI plan which shows the detail of the SPI project (connected with the ideas presented during the managing BPM projects training element – trainees express a great satisfaction for this element). A SPI effort on the other hand, is a knowledge creator mechanism through which knowledge will be captured, modified and transferred to different organizational levels (the main advantages seen by the trainees based on understanding the practical utility of the CertiBPM training program).

In this context, a SPI plan should be expanded to cover not only the details of process improvement efforts but also, the choice of change and knowledge management efforts in the SPI program. Approaching *SPI through a change and knowledge management* point of view recommends having a change strategy instead of an SPI plan which connects people related efforts to process related activities in order to make change happen in practice. The change strategy should be clear in its focus, detailed, communicated, accepted and agreed by people in different organizational levels (management, organizations, and working teams) for being most effective.

Other research results based on the trainees feed-back are presented in Table 4. For each answer to the questions below, a Likert scale with 5 point was use to quantified the trainees answers (1 – categorical no; 2 – no; 3 – sometime, partial no and yes; 4 – yes; 5 – strongly support).

The research results were strongly affected by the large group (38 employees involved in the CertiBPM training program) of the automotive industry companies and it can be considered that these answers are also, characteristics of these companies' strategy development and implementation in Romania West Region.

Table 4. Mintzberg Criteria for Effective SPI Strategy related to BPM activities – Research Results

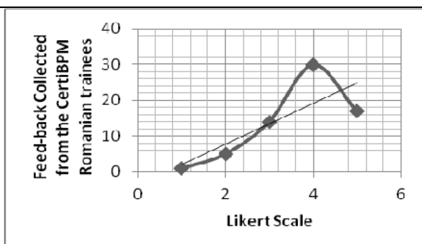
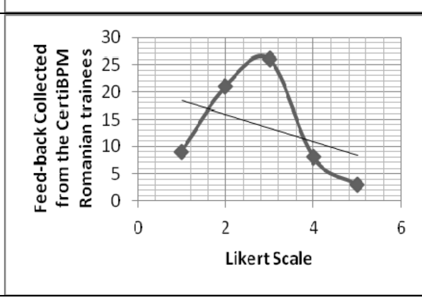
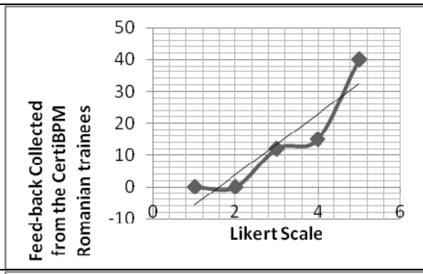
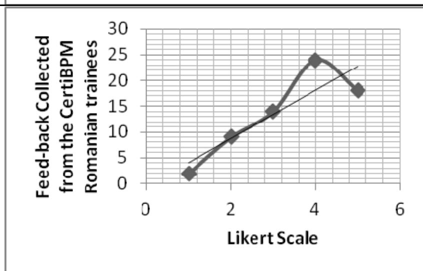
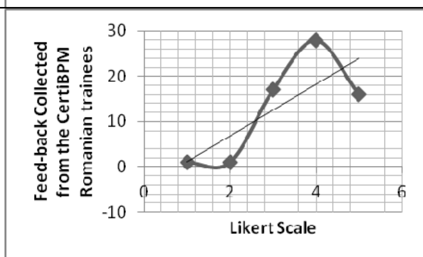
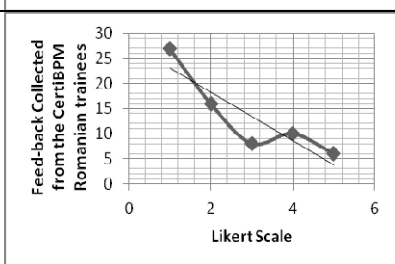
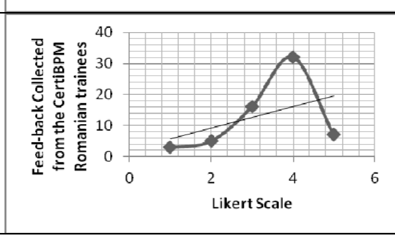
No.	The Critical Element	The Key Questions	Answers given by the target group
1	Clear, decisive objectives		The answers tendency shows there is a strongly support of the management teams for clear and decisive objectives
2	Maintaining the initiative		The answer shows that there is a decrease tendency for the strategy to preserve freedom of action and enhance commitment.

Table 4. (*Continued*)

3	Concentration		The employees' answers show a positive trend for the strategy to concentrate superior power at the place and time likely to the decisive.
4	Flexibility		The trainees' answers demonstrate that their company strategy is built in resource buffers and dimensions for flexibility and manoeuvre.
5	Coordinated and committed leadership		The answers show the increase tendency of the strategies to provide responsible, committed leadership for each of its major goals.
6	Surprise		The trend curve of the answers shows that strategy surprise, attack exposed or unprepared opponents at unexpected times less. Most of the employees were not surprised!
7	Security		The answers show that most of the strategies secure resources and all vital operating points for the enterprise.

4 Conclusions and Lessons Learned

Through this article we intent to demonstrate how a qualification and certification program, like CertiBPM, developed and implemented with the support of ECQA, can be useful for optimizing the SPI efforts and strategies into Romanian companies. The research results show only the preliminary opinions (collected through the trainees feed-back) about the SPI efforts and how could the knowledge pool generated by the CertiBPM training program, to support these efforts. First, in the introduction, there have been explained briefly the SPI concept and its need for the modern companies. Also, the SPI Manifesto initiative have been describe as a good practice, a Community of Practice that have been developed in relation with the EuroSPI series of conferences of ECQA, to support SPI general trend and standard implementation. The context of this paper is defined by the CertiBPM project activities of research and develops a qualification and certification program using the ECQA schema and guidelines. Feed-back was collected (after each training unit, element) after 6 days of training sessions in BPM field, with a large group of employees and students (67 people trained) from the West Region of Romania. The debates and employees answers analysis allow us to make an inventory and to describe the initiated process improvement (SPI efforts and strategies) in important companies from the West Region of Romania.

The marketing survey have underline that for Romanian companies approaching SPI from a change management point of view in the light of strategy help them to view the whole effort in a wider perspective. Therefore, Romanian managers focus moves from improving processes to improving practices, from planning for a project to planning for a program, from assessing the process maturity level of the organization to assessing the people's ability and willingness to change, from identifying the gaps and process improvement activities to capturing and creating new knowledge, modifying and transferring it to different organizational levels. In this context, the identification and describe efforts can be focused on identifying issues related to e.g.: resistance to change, from integrating processes to integrating change and knowledge management efforts, and from implementing new processes to cultivating a culture of change openness and management. Through this approach trainees from Romanian companies underline the role of a detailed, clear, well documented, agreed, accepted, and communicated SPI strategy that becomes more visible. This strategy should define a path for addressing the three main parallel streams of an SPI program, i.e. Practice improvement, Knowledge Management, and Change Management. The marketing survey, general conclusion suggests that an SPI effort should be based on an SPI strategy including answers for how to improve practice of software work through organizational change and knowledge management efforts. The strategy should be effect driven, fit to the organization's requirements and be developed and approved before the SPI efforts starts.

Acknowledgements. The CertiBPM project is financially supported by the European Commission in the Leonardo da Vinci part of the Lifelong Learning Programme under the project number LLP-LdV/TOI/10/RO/010.

References

1. Fung, R.Y.K., Tam, W.T., Ip, A.W.H., Lau, H.C.W.: Software process improvement strategy for enterprise information systems development. International Journal of Information Technology and Management 1(2-3), 225–241 (2004)
2. Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R., Musat, D., Castaño, V., Calvo-Manzano, J. A.: Systems, Software and Services Process Improvement. In: Riel A., O'Connor R., Tichkiewitch S., Messnarz R. (Eds.) CCIS, vol. 99, pp. 280–289. Springer, Heidelberg (2010)
3. Pries-Heje, J., Johansen J.: SPI Manifesto. Version A.1.2.2010 (2010), http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf
4. Hanna, M.D., Newman, W.R., Johnson, P.: Linking operational and environmental improvement through employee involvement. International Journal of Operations & Production Management 20(2), 148–165 (2000)
5. Brocke, J., Rosemann, M.: Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture. In: International Handbooks on Information Systems, vol. 1. Springer, Berlin (2010)
6. CertiBPM project. Certified Business Process Manager, contract no. LLP-LdV/TOI/10/RO/010 (2009)
7. Riel, A.: EU Certificates and Knowledge Communities in Europe: An unbeatable Symbiosis. Keynote at EQN Founding and Dissemination Conference, Krems, Austria. CD-ROM (2006)
8. 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 (2010)
9. Messnarz, R., et al.: Assessment Based Learning Centers. In: Proceedings of the EuroSPI 2006 Conference, Joensuu, Finland (2006); also published in Wiley SPIP Proceeding (June 2007)
10. Messnarz, R., et al.: The EQN Guide, Graz, Austria (2008)
11. Draghici, A., Mocan, M., Draghici, G.: On-Line Training and Certification Solution for Business Process Managers. In: Cruz-Cunha, M.M., Varajão, J., Powell, P., Martinho, R. (eds.) CENTERIS 2011, Part I. CCIS, vol. 219, pp. 380–389. Springer, Heidelberg (2011)
12. Mintzberg, H., Quinn, J.B., Ghoshal, S.: The Strategy Process, the revised edn. Grate Britian by The Bath Press, Bath (1998)
13. Christenson, C.R., Andrews, K.R., Bower, J.L.: Business Policy: text and Cases. Richard D. Irwin, Homewood (1978)

EU Project BPM-GOSPEL – Applying Compliance Management Scenarios in Business Process Modelling for Trusted Business Coaching Programs

János Ivanyos[1], Richard Messnarz[2], József Roóz[3], and Oliver Hammrich[4]

[1] Memolux Ltd., Budapest, Hungary
ivanyos@content.hu
[2] ISCN, Bray, Ireland
rmess@iscn.com
[3] Budapest Business School, Budapest, Hungary
rooz.jozsef@bgf.hu
[4] Methodpark, Erlangen, Germany
oliver.hammrich@methodpark.de

Abstract. The well established and recognized control frameworks and process reference models could be used for effective and efficient enterprise governance, if only the management established its own governance related objectives. Unfortunately, structures and literatures of control frameworks and reference models are not easily interpretable by enterprise management for setting their own business' specific governance objectives. This article gives an overview of how compliance management scenarios implemented by using business process modelling and process assessment tools provide best practice based resources for vocational education in enterprise governance domain with advancing ECQA certified Governance SPICE Assessor training programs.

Keywords: Compliance Management, Business Process Modelling, Governance Capability, ISO/IEC 15504 (SPICE), Trusted Business, Coaching.

1 Background and Applied Methodology

The BPM-GOSPEL Project

The objective of the BPM-GOSPEL - Business Process Modelling for Governance SPICE and Internal Financial Control - project implemented with the financial support of the Commission of the European Communities under the LEONARDO DA VINCI Programme (2010-2012, Project number: LLP-LdV-TOI-2010-HU-001) is providing ready to use compliance management scenarios for enterprises and best practice based vocational trainings demanded by both sides of the labour market.

The "Governance Model for Trusted Businesses" [1] – a publicly available deliverable of the BPM-GOSPEL project – establishes reference processes for mapping operational process management to compliance and audit management implemented by the "Stages" platform (www.methodpark.com/en/product.html) with interfaces to external workflow management and audit management tools.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 288–299, 2012.

Adding business driven case studies into training programmes (like www.training.ia-manager.org) supports understanding of the competencies needed and best practices relevant for business practitioners. Employers are interested in on-the-job trainings where the acquired skills and knowledge can be directly tested and certified by applying the offered methodology and tools in live environment.

The platform system "Stages" is adapted as a multi-layer BPM master example for coaching practical implementation of internal financial controls with IT support applied by private and public sector companies following internationally recognized control frameworks like COSO [2], COBIT [3], Enterprise SPICE [4] and related assessment (audit) approach (Governance SPICE).

Extending the Governance SPICE Assessor Skill Card

Using the internationally recognized terminology outlined in skill definition models, the following skill hierarchy for the "Governance SPICE Assessor" job-role [5] already certified by the European Certification and Qualification Association (ECQA) has been extended by the "Governance Objectives" Skill Unit:

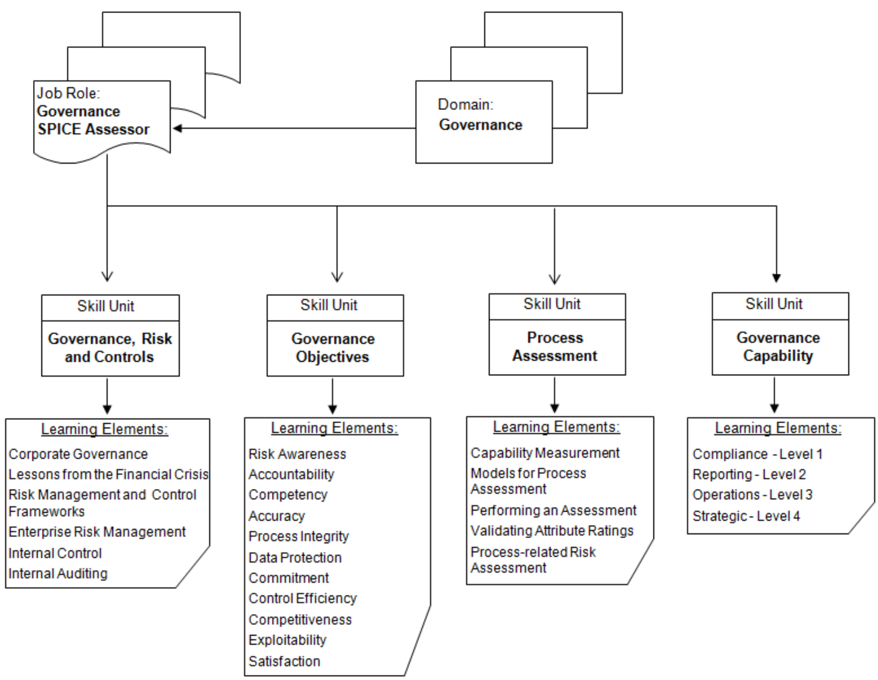


Fig. 1. The updated Governance SPICE Assessor Skill Card

The paper presents how compliance management scenarios have been implemented for supporting acquiring the knowledge elements of the "Governance Objectives" Skill Unit.

Using Governance Capability Assessment (Governance SPICE)
The term of "Governance Capability Assessment" [6] is used in context of Governance, Risk Management and Internal Control processes based on different concepts:

- Corporate Governance Principles (OECD)
- Recognized Control Frameworks and Reference Models (like COSO, COBIT, Enterprise SPICE, etc.)
- Risk Tolerance and Risk Appetite (as of COSO ERM)
- Performance Measurement (as of COBIT)
- Process Capability Assessment (ISO/IEC 15504-2:2003) [7]
- Evaluating Process-related Risk (ISO/IEC 15504-4:2004)

Internal and external audit standards (like IIA and ISA) recommend system based evaluation of existing internal controls against internationally recognized control frameworks like COSO (Internal Control – Integrated Framework) and COBIT (Control Objectives for Information and related Technology). The contents of these frameworks are applicable to set up Process Reference Models in compliance with ISO/IEC 15504-2 requirements.

The selected processes from the COSO and COBIT based Process Reference Models and the Enterprise SPICE model associated with the process attributes defined in ISO/IEC 15504-2 provide a common basis for performing assessments of governance capability regarding internal controls and reporting of results by using a common rating scale. ISO/IEC 15504 offers not only transparent method for assessing performance of relevant governance processes, but also tools for assessing control risk areas based on the gaps between target and assessed capability profiles.

Governance Objectives
The well established and recognized control frameworks and process reference models could be used for effective and efficient enterprise governance, if only the management established its own governance related objectives. Unfortunately, structures of control frameworks and reference models are not easily interpretable by enterprise management for setting their business' specific governance objectives. Furthermore, the external and internal audit standards and literatures are also not really supportive in these terms.

The "Governance Model for Trusted Businesses" keeps both enterprise management and audit assurance logics in mind by presenting governance processes in line with the objectives relevant for enterprise management, together with an exact mapping to processes of control frameworks (reference models) accepted and used by auditors for compliance attestation. The reference to applicable ISO/IEC 15504 conformant processes allows management and auditors to use governance capability profiles in context of the governance objectives.

The "Governance Model for Trusted Businesses" interprets the following governance objectives for determining governance processes as special applications of the recognized reference models (COSO, COBIT and Enterprise SPICE) and trusted business principles [8]:

- Supporting Organization's Internal Control System
  - Risk Awareness
  - Accountability

  - Competency
  - Accuracy
  - Process Integrity
  - Data Protection
  - Commitment
  - Control Efficiency
- Supporting Business Sustainability
  - Competitiveness
  - Exploitability
  - Satisfaction

Using the “Governance Model for Trusted Businesses” have some immediate evidential benefits. At first, it provides ready to use structure for implementing selected or all elements of the recognized control frameworks and generic enterprise models. Regulatory or voluntary compliance requirements could be looked through clear business driven governance objectives helping better understanding and meaningful design and operation by enterprise management. Besides the less implementation efforts, this structure unburdens the internal and external audit activities in concluding opinion about the fulfilment of compliance requirements.

At second, the “Governance Model for Trusted Businesses” offers sufficient set of practices to determine the enterprise specific control objectives. The management can easily select and communicate those minimum requirements which are considered as crucial for running business on the specific market (composing the risk appetite). This decision is a clear message to all stakeholders, including potential customers, that which operational risks are planned to be mitigated by the enterprise management, and which risks remain unattended.

Furthermore, setting target governance capability levels to the governance processes or even to the applied entity-level control processes helps to interpret qualitative and quantitative measures for design and operational effectiveness conclusions. By this way, the business process management solutions, like workflow systems, project toolkits, reporting tools, etc. can be configured for automatically collecting and providing performance information both about the process-level and entity-level controls, based on the manually and/or electronically processed business activities.

2 Business Contexts of the Business Case Implementation

Business Environment’s Expectations for Service Organization’s Control

“Many companies function more efficiently and profitably by outsourcing tasks or entire functions to service organizations that have the personnel, expertise, equipment, or technology to accomplish these tasks or functions. Examples of such services include cloud computing, managed security, health care claims management and processing, sales force automation etc. Although user management can delegate these tasks or functions to a service organization, they are usually held responsible by those charged with, customers, shareholders, regulators and other affected parties for establishing effective controls over those outsourced functions.” [9]

The so called *Service Organization's Control* (SOC) reports provide user management with the information they need about the service organization's controls to help assess and address the risks associated with an outsourced service.

The business process management solutions, like workflow systems, project toolkits, reporting tools, etc. can be configured for automatically collecting and providing performance information both about the process-level and entity-level controls, based on the manually and/or electronically processed business activities. Tracking of the evidences for governance process performance by a process management suite, which is also able to map these evidences to process assessment models – like in the case of the "Stages" for Trusted Business – provides solution to automate formal assessment of governance capability over a period of time. These assessment results certified by a qualified issuer can be published directly by the assessed enterprise, or via a "trusted business" promotion portal.

Within the Controlled Operation application category of the "Governance Model for Trusted Businesses" there are 8 processes applying the practices of all the 20 COSO Internal Control over Financial Reporting Principles and 3 selected COBIT processes, together also covering criteria for the Security, Availability, Processing Integrity, Confidentiality Principles and the Generally Accepted Privacy Principles. By assuring compliance with the 8 governance processes of Controlled Operation, service organization management is able to provide assertions that the service controls and related control objectives relevant for user entities' are suitably designed based on the applicable criteria and operating effectively over a period of time.

The Baseline Business Case Description

Memolux Ltd. is a Hungarian privately owned SME established in 1989. It provides high level outsourcing services covering the full scope of business administration to more than 200 clients. Memolux has more than 20 years experience of providing payroll services for a wide range of user entities. Memolux is the partner of ADP Employer Services International in Hungary to provide ADP Streamline®. ADP Streamline® is a Global Payroll and HR administration outsourcing service designed for international organizations.

The 20 people staffed Payroll Department has been using in-house developed web-based payroll system for payroll related data maintenance, calculation and reporting for about 100 current clients. The system provides many client-specific interfaces to banking systems and management information systems (like SAP) used in Hungary or world-wide. Software Development unit has been participating in European process improvement experiments since 1995 and maintaining ISO 9001 compliant quality controls.

Since 2011 a workflow management system has been implemented for supporting management controls over the huge number of parallel payroll processing cycles. At first the monthly payroll calculation process was modelled and implemented for starting trials. The workflow management tool has also the role of providing evidences for management assertions of process-level service organization's controls.

Management assertions for process-level service controls have been identified by responding control risks at activity steps of the monthly payroll calculations in the areas covered by the risk-based selected application practices offered by the "Governance Model for Trusted Businesses". Day-to-day operation of monthly payroll processing cycles provides automatic logs, checklists, generated control data and reports within the workflow management system.

"Stages" process modelling platform is used for mapping workflow based control evidences to generic process reference model, like the "Governance Model for Trusted Businesses". For that purpose both the reference model and the monthly payroll calculation business process have been configured. The configured processing and control requirements are useful for not only management or audit scope, but they are also applied in knowledge sharing during in-house trainings or informing new employees.

Compliance Workbench functions of "Stages" allow at first to select the relevant set of governance practices and even work products as company specific scope of the reference model. This is the result of the risk assessment performed by the management concerning to the governance objectives (based on business goals and business environment's expectations). At second by using Compliance Workbench functions, the elements of the business processes can be mapped to the scoped governance objectives and can be referred as management assertions for effective operation of the designed controls. At third, the evidence "pools" generated or maintained by the workflow management system can be hyperlinked to these business process elements.

The audit tool in this case is the "Capability Adviser" platform, which allows looking at the business processes and business units through the glance of the recognized reference models' perspectives. By using "Stages" process modelling platform in between the evidence pool (workflow management) and the compliance audit or even governance capability assessment tool, the audit work can be transparently fastened by getting evidences directly from where the business processes have been performed, and by gaining better understanding of management's risk assessment and risk taking approach necessary for judgements of control design effectiveness.

The Monthly Payroll Calculation Process as Business Case Baseline

Among the full set of the processes related to the payroll business operation the monthly payroll calculation process has been selected as a baseline business case. Naturally this means a limitation concerning to the scope of the governance objectives, as some governance objectives are more related to IT governance, business administration, sales, and management processes. However the selected baseline business process represents the mainstream of the business service operation and supervision, where the Satisfaction, Risk Awareness, Accountability, Accuracy and Process Integrity objectives are applicable. As the capability of the Accuracy objective related controls over the monthly payroll calculation process is the most crucial from the client/user organization's perspective, the related controls are embedded even into the detailed activity levels, as well.

3 Implementing the Compliance Management Scenarios

Selecting the Scope of the Management Assertions

As the first step of scoping of the management assertions for the selected governance objectives, a risk assessment should be performed. For supporting this step the objective related risk tables of the "Governance Model for Trusted Businesses" can be used to understand key risk areas and applicable practices as risk responses. An example of governance objective (Risk Awareness) related risk table is presented as follows:

Table 1. Sample risk table of Risk Awareness governance objective

Key Risk	Risk Factors	Responses	Applicable COSO processes	Application Practices
Relevant governance risks are not considered	Governance objectives for business processes are inadequately established	Management sets clearly defined objectives for governance including risk tolerance and risk appetite	Governance (Financial Reporting) Objectives (COSO)	Management specifies governance objectives relevant for financial reporting and trusted business operation with sufficient clarity and criteria to enable the identification of risks to the achievement of the governance objectives relevant for financial reporting and trusted business operation.
	Inconsistency in risk assessment	Risk assessments are periodically performed by considering the time horizon of the governance objectives, risk tolerance and risk appetite	Governance (Financial Reporting) Risks (COSO)	The organization identifies and analyses risks to the achievement of governance objectives relevant for the organization's financial reporting and trusted business operation as a basis for determining how the risks should be managed.
	Risks relevant for organizations' internal control system are not addressed	Control activities developed by reflecting to all assertions relevant for organization's internal control system	Integration with Risk Assessment (COSO)	Actions are taken to address risks to the achievement of governance objectives relevant for financial reporting and trusted business operation.

Management assertions are setting links between the business processes (and activities) and the governance objective-driven processes (and base practices) offered by recognized reference models (like COSO, COBIT and Enterprise SPICE). Management should reflect to their own business goals and business environment's expectations by setting the limits of tolerable deviations (risk tolerance) from the achievement of the governance objectives.

Governance Capability Assessment (Governance SPICE) methodology provides measurement framework for targeting and assessing effectiveness of governance processes (defined by ISO/IEC 15504 compliant reference models) in achievement of governance objectives, however it should be carefully considered at the assessment scope definition, that a governance process might be implemented and applied by more than one business processes (and their instances) within a business operational unit.

For example in case of Accuracy objective concerning to the Monthly Payroll Calculation, the management should determine the management and control activities necessary and sufficient to limit the negative consequences of the inherent business risk related to inconsistency in payroll processing data architecture and disclosure elements. Theoretically the management should apply a measure for determining impact on business goals, however at most cases this measure can't derived directly from "natural" business measures like income, profitability, stock listing rate, market-share, etc. This problem could be resolved by targeting application of governance practices by business process activities. The more governance practices are applied, the negative consequences are the more likely minimized. By doing so, the Governance Capability (SPICE) measurement framework can be used.

At corporate level the COSO, COBIT and Enterprise SPICE reference models based governance processes are applicable for targeting and assessing them for even higher - above level 2 - governance capability. However from a business process view, as in the case of Monthly Payroll Calculation, the assessment target is mostly limited to Compliance (1 - Performed) level, as this is the level where the application of specific process outcomes and base practices are investigated. Targeting the Reporting (2 - Managed) level for the governance process would be applicable, if only the investigated business process were the only instance within the business unit for assessing the capability of the Accuracy objective related governance process.

For example by mapping the Monthly Payroll Calculation business process activities with the Accuracy objective related governance practices, company management sets assertions for achieving Accuracy governance objective in a specific manner, which is adequate to the management risk taking behaviour and decisions related to potential inconsistency in payroll processing data architecture and disclosure elements. Of course, the Accuracy objective related governance practices are also applied by other business processes within a payroll service, so these assertions are closely linked with others e.g. defined for in-house software development and IT infrastructure management.

Implementing the Monthly Payroll Calculation Business Process in “Stages”

“Stages” platform for Trusted Business enables professionals to define and enact business processes, control their standard compliance to governance models and measure process performance. This efficiently stabilizes the foundations for high governance capability in complex corporate business environments.

The key idea behind the “Stages” process management system is to bring process theory and business practices together. Stages is optimized for complex but creativity-driven processes and integrates with a large number of tools. It focuses on the end users of processes and provides them with easy access to process descriptions, allowing them to understand both transparent end-to-end processes and role-centric process details.

“Stages” for Trusted Business contains the preconfigured “Governance Model for Trusted Businesses”, as a reference model for compliance related activities. Business process implementation started with defining multiple level activity layers. Within the business process definition at first the supervision (by operational management) layer was set up. This layer presents the operational management’s high level supervision view of the process.

At this activity layer, there is a simple presentation of the main activities of the Monthly Payroll Calculation process with their interdependences and the key inputs and outputs. The workflow schedule and logs provide technical means for management to effectively supervise a Monthly Payroll Calculation process instance and to act immediately if an activity delay or deviation occur. The “Payroll Cycle Supervision” is implemented as an activity at the supervision layer.

Each of the Monthly Payroll Calculation activities has also a more detailed view extended with activity order links, roles, and next activity layer containing related management assertions.

These sub-layers present the operational staff’s view on the business process activities extended with order links to other staff level activities and activity specific inputs and outputs. The business process activities should be performed by keeping requirements set by the related assertions (derived from the governance objectives). The assertion type “activities” contains reference to the assigned governance objective, weight within assessment scope, mitigated risk area, control description and features, and might have an html link to evidence pool.

Using Compliance Workbench of “Stages”

In order to gage the quality of your business processes, a method is needed to evaluate and quantify them. The evaluation must fulfil criteria with respect to objectivity, reproducibility and comparability, as only then is the evaluation meaningful. To achieving such an objective process evaluation, the business processes should be mapped against one or more reference models (e.g. maturity or capability models developed as a quality gage by industry and standardization committees, standards or even in-house standards within an organization). In our case the business process is mapped to the Accuracy objective related elements (base practices) of the “Governance Model for Trusted Businesses”.

By mapping your business process against elements of one or more reference models, you can obtain an evaluation overview that makes visible any deviations as well as the necessary corrective action for improving quality.

Reference models need to be available within “Stages” in order to be able to map business processes against a reference model. These reference models may either be generated from existing processes implemented by “Stages” or they may be imported from external sources.

The definition of a scope in a reference model allows only the required scope to be selected from the entire reference model. As those elements which are not necessary for mapping purposes may be disregarded, the mapping process takes less time. The scope might be limited based on specific assessment objectives by referring to only a subset of the reference model elements or even element types.

By exporting the scopes and mappings between business processes and reference models, external validation tools can be also used for independent evaluation and qualification of the achievement of governance objectives.

External Validation by “Capability Adviser”

The “Capability Adviser” tool supports ISO/IEC 15504 conformant process capability assessments - based on recognized models like ISO/IEC 15504-5, Automotive SPICE™, COSO, COBIT and others - over the Intra-/Internet, creates Profiles, Assessment Reports and integrates Learning Management Systems to distribute already established best practices in the company.

The interface between “Stages” platform and “Capability Adviser” tool extends the control over business processes by allowing external assessors/auditors to effectively perform compliance audits (Level 1 assessments) and capability assessments (Level 1-5) for process improvement (consulting) and capability determination (assurance) purposes. While “Stages” platform provides modelling solution focusing on the business-driven process descriptions and their implementation with useful mappings to elements of the configured generic models or standards, “Capability Adviser” is more scoping on the assessment work based on the recognized ISO/IEC 15504 conformant process assessment models and using the instances of the business process elements as evidence references.

By using the interface between “Stages” and “Capability Adviser”, the following workflow is followed:

- Business process owners are working on the “Stages” portal. They are uploading/storing their business process descriptions, control information, mappings to reference models, and the links to the workflow and documentation management and transaction processing systems.
- The “Capability Adviser” tool is used to perform an assessment on the compliance with reference models and governance capability evidenced by the performed business processes.
- A major part of the assessment work is collecting the evidences. They proof and describe how the company works according to the defined processes. The assessed/audited team usually needs to upload and assign their work products as evidences in the “Capability Adviser”. To improve the evidence collection

phase, the documents and references (links) already implemented and used by "Stages portal" are electronically imported as URL links and assigned to the correct process elements into the "Capability Adviser".

- Then the assessors are able to directly open the evidences from "Stages" during the assessment.

Once the evidence collection is completed, the assessors rate the process attributes, so that the assessed business unit will receive formal capability level and attribute rating profiles with notes to highlight areas where business unit demonstrated a level of competence or missed competence with an explanation why evidences are not sufficient. "Capability Adviser" provides assessment logs and reports including all ratings and comments.

4 Conclusions

By using the baseline business case of a payroll outsourcing service process, the Trusted Business coaching programs are able to use case studies supporting knowledge sharing focusing on the selected learning objectives related to the "Governance Model for Trusted Businesses". The specific business environment has internationally standardized (SOC 1 and SOC 2 replacing former SAS70) control requirements which should be carefully considered by small business companies providing local services to multinational clients, whose compliance managers, internal and external auditors are making great demands on local service providers and raising difficulties for these companies by increasing requested control and audit efforts and costs. At most cases these demands are driven by the multinational organizations' global compliance or audit requirements, so they are not really "customized" for the local conditions.

The implemented compliance management scenarios present how a local small business organization can efficiently implement compliant control frameworks with respect of its real business needs and risks, and how the implementation results can be exhibited for external evaluation or audit in a cost effective way.

The presented compliance management scenarios can be followed for implementing and evaluating application practices evidencing achievement of all relevant governance objectives. However at a business unit or organization level, the selected governance objectives might be achieved and evidenced through assertions existing at more business processes, so one business process is typically not providing a baseline for all governance objectives. Furthermore, some of the governance objectives are supported by entity-level control processes, while others by process-level controls or by a combination of these levels.

The qualification process of a business unit's compliance to its unequally customized governance objectives - defined by the specific scoping of the governance practices from the "Governance Model for Trusted Businesses" - should cover all those business processes and information sources, which provide the sufficient evidences for management assertions concerning to the effective and efficient implementation of the Trusted Business governance processes defined by the Model. Based on the implementation scenarios referred by this paper, the aimed Trusted Business coaching programs will cover the following elements:

1. Introduction to the relevant learning elements based on the “Governance Objectives” unit of the ECQA certified Governance SPICE Assessor Skill Card.
2. Summary of the business environment’s expectations concerning to the baseline business process in context of the selected governance objectives.
3. Scope setting of the management assertions in context of the governance objectives.
4. Use-case modelling by the “Stages” for Trusted Business.
5. Model based evidence collection by using “Stages” Compliance Workbench.
6. Evaluation of compliance and governance capability profile by using “Capability Adviser” assessment reports.

References

1. Governance Model for Trusted Businesses, BPM-GOSPEL Deliverable (2011)
2. The Committee of Sponsoring Organizations of the Treadway Commission (COSO):
   - Internal Control - Integrated Framework (1992)
   - Enterprise Risk Management - Integrated Framework (2004)
   - Internal Control over Financial Reporting - Guidance for Smaller Public Companies (2006)
3. COBIT - Control Objectives for Information and related Technology, COBIT 4.1 © 2007 IT Governance Institute (2007), http://www.itgi.org
4. Enterprise SPICE - An Integrated Model for Enterprise-wide Assessment and Improvement. Technical Report – Issue 1 September 2010. The SPICE User Group (2010)
5. Governance SPICE Assessor and Internal Financial Control Assessor Skill Cards, ECQA Committees (2011)
6. Ivanyos, J., Roóz, J., Messnarz, R.: Governance Capability Assessment: Using ISO/IEC 15504 for Internal Financial Controls and IT Management. In: The MONTIFIC Book, MONTIFIC-ECQA Joint Conference Proceedings (2010)
7. ISO/IEC 15504-1:2004 Information technology – Process assessment – Part 1: Concepts and vocabulary (2004)
   ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment
   ISO/IEC 15504-2:2003/Cor 1:2004
   ISO/IEC 15504-3:2004 Information technology -- Process assessment -- Part 3: Guidance on performing an assessment
   ISO/IEC 15504-4:2004 Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination
8. Trust Services Principles, Criteria, and Illustrations Copyright © 2009 by the American Institute of Certified Public Accountants, Inc., and Canadian Institute of Chartered Accountants
9. Users and User Entities - Service Organization Control (SOC) Guides, Copyright © 2006-2012 American Institute of CPAs, http://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/pages/users.aspx

There Is No Knowledge without Terminology: Key Factors for Organisational Learning

Blanca Nájera Villar and Diana Brändle

TermNet, Austria
projects@termnet.org

Abstract. Any company or organisation in the world has to manage its knowledge. For some it may be sufficient to have the knowledge in their brains. But as products and services are subject to constant innovation, most of the companies and organisations acting in the global market or on an international level find themselves confronted with the task of constantly updating, referencing, tracking and managing knowledge. Marketing material, speeches on conferences and sophisticated tools make us believe that it is easy to keep track of information in a company or organisation, to use single source publishing, touch each content only once, have it centrally stored and reuse it for multiple purposes. How many of you have really achieved this goal or know organisations, where all content-related processes run smoothly and are fed by a central knowledge base? Is it actually realistic to follow the ideal of ONE central knowledge base in a company / organisation or just an utopian idea? We will have a closer look at challenges and possible solutions for innovative organisations and companies when managing units of knowledge represented in the form of technical words or „terms“ in one or more languages.

Keywords: Terminology, multilingual knowledge base, single source publishing, organisational learning, learning organisation, terminological database, organisational knowledge, globalisation, multilingual, terms.

1 Current Situation

The knowledge [3] related to a product or service is, if at all, stored in many different IT systems, serving different needs, targeting different user groups. The data have different formats, are assigned with different metadata, set up in various languages and are mostly stored at different places. Only certain users or user groups have access to specific data.

In lager companies or organisations it may even happen that people working in different departments or subsidiaries are not aware of existing inventories [4] in the other department or subsidiary, even if access to this information would be helpful for them. Obstacles for exchanging information are not only different formats, contents, tags, etc., but sometimes simply lacking knowledge of existing information [5]. Consequently, a lot of information is available redundantly in different systems;the effort for maintaining data in various systems costs a lot of time and money; and there is a high risk of inconsistent or even incorrect data.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 300–309, 2012.

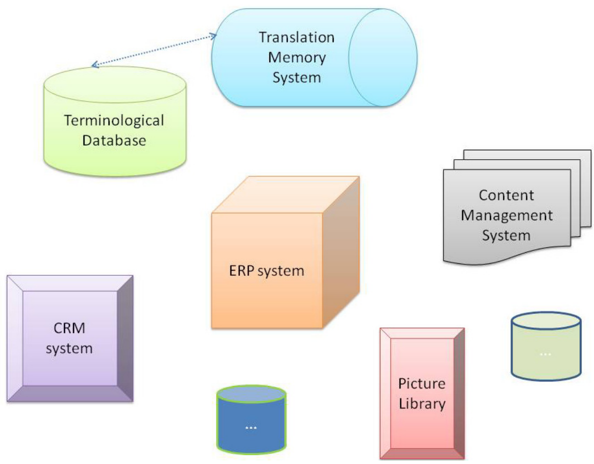


Fig. 1. Corporate knowledge stored in independent systems

2 One Central Knowledge Base or Several Systems?

For a long time terminologists were convinced that the ideal solution to manage corporate or organisational knowledge would be to put a terminological database [1], [7] at the centre of the IT landscape and feed other systems and contents with its data.

But this approach has proved to be unrealistic as the effort to manage all organisational or corporate knowledge with all its specific formats, target groups, languages, codes, pictures, texts, symbols, types of information, different development stages and workflow status, etc. in one system is unmanageable.

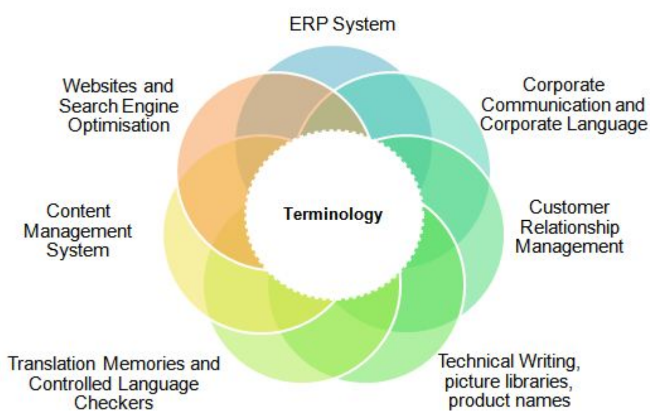


Fig. 2. Outdated: Ideal to have a terminological database in the centre

The ideal of a central term base, however, still does make sense from a translation department's view:

Fig. 3. Central term base for translation departments or language service providers

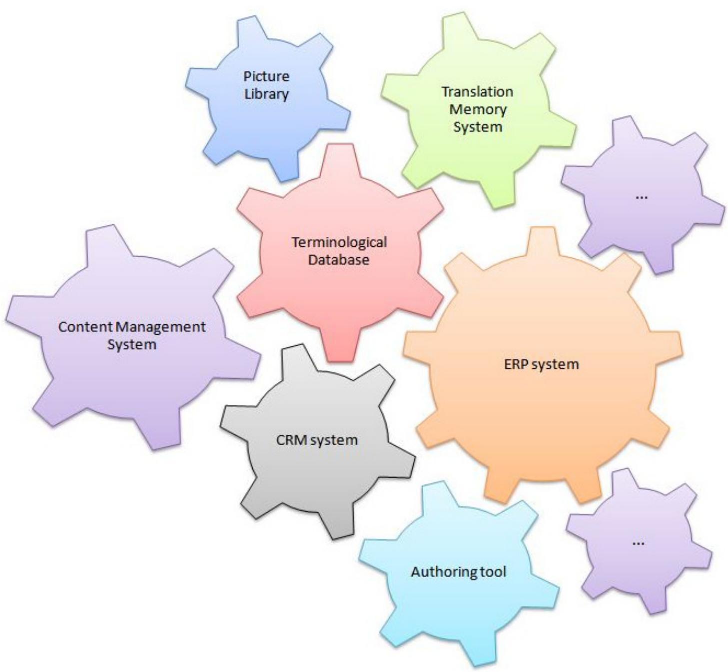


Fig. 4. Realistic approach: interlinked solutions and systems

Translation departments are dependent on input from other departments and have to exchange information and data that are relevant for translating various types of texts and contents. From this point of view it makes sense to put a terminological database in the centre of a language service provider's IT landscape.

To keep an eye on the bigger picture, the solution for larger companies and organisations will be a still heterogeneous IT landscape, except that exchanging data and using information for multiple purposes is made easier by increased automation that simplifies complex, interrelated processes involving many departments and persons. Single source publishing is certainly possibly, but it also depends on different systems which – at least so far - cannot all be replaced or merged.

3 Terminology in the Innovation Process

No matter, whether a new machine is being developed or a new service offered, innovative processes **are always accompanied by communication and documentation. An idea has to be given a name, its** features and functions have to be explained, its parts must be listed and all this knowledge should be easily accessible for all people involved.

3.1 Examples from the Car Industry

Car models are modified every now and then. These so-called "facelifts" include mostly smaller changes of the head or rear lamps:

Fig. 5. Facelift Audi S4 (© Audi AG)

A new generation of a car model is produced on average every 4 to 6 years. In the process of this so-called model changeover, comprehensive revisions and innovations concerning design and technology are carried out:

Fig. 6. Evolution of the AUDI A3 radiator grill (© Audi AG)

The above figure shows how the radiator grille of one individual car model changed over not even a decade. The terms for the individual model generation changed in German from “Ziergitter” over “doubleframe” to “singleframe”.

The long-term effect of classifying information using a terminological database linked e.g. with an Enterprise Resource Planning system, a Customer Relationship Management database, spare parts and price lists, picture libraries, etc. will be more transparent knowledge that can be used for multiple purposes.

In a terminological term base information on the AUDI A3 radiator grille may look like this:

Entry no.: 1	
Subject field:	Automotive
Model:	A3
Model year:	2004 - 2011
German:	**Singleframe-Kühlergrill**
Part of speech:	noun
Gender:	nt
Definition:	…
Def-Source:	…
Source:	…
English:	…
French:	…

Fig. 7. Terminological entry no. 1

Entry no.: 2	
Subject field:	Automotive
Model:	A3
Model year:	2003 – 2004
German:	**Doubleframe-Kühlergrill**
Part of speech:	noun
Gender:	nt
Definition:	…
Def-Source:	…
Source:	…
English:	…
French:	…

Fig. 8. Terminological entry no. 2

Entry no.: 3	
Subject field:	Automotive
Model:	A3
Model year:	1996 – 2003
German:	**Zierrahmen-Kühlergrill**
Part of speech:	noun
Gender:	nt
Definition:	…
Def-Source:	…
Source:	…
English:	…

Fig. 9. Terminological entry no. 3

Fig. 10. Audi paint "Eisvogelblau, Perleffekt" (© Audi AG)

The above picture shows an Audi A3 paint, called "Eisvogelblau, Perleffekt". It was only available until model year 2011 and since then has not been replaced. If you were a customer or retailer of Audi and would like to order a car in this colour, but could not find it, would it not be interesting for you to know that this model was available with this paint until 2011, but not any more? If you were a customer or retailer of Audi in an English-speaking country, how would you find information unless you knew that the English equivalent for “Eisvogelblau, Perleffekt” was not – as you might possibly have expected “Kingfisher Blue, Pearl Effect” -, but “Glass Blue, pearl effect”?

If we look at the table below, we have an idea how complex it is not only to keep track of changes, but also to classify information (set up concept and classifications systems) in larger companies or organisations:

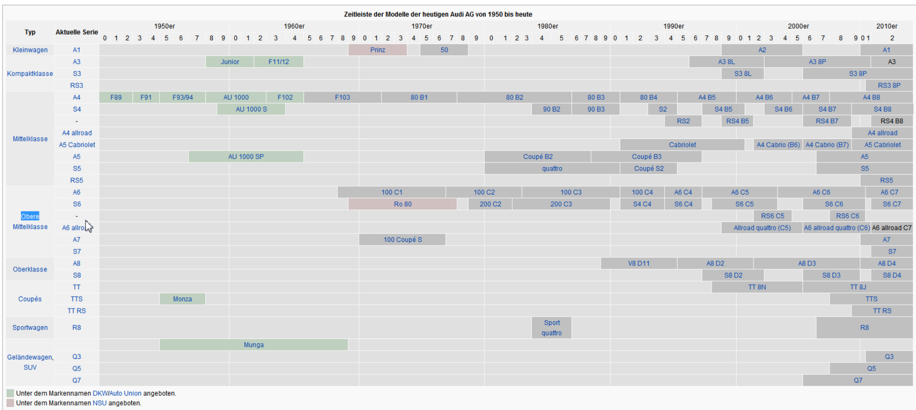


Fig. 11. Timeline of Audi’s models from 1950 until today (© Audi AG)

One aspect on the way to achieve the goal of more transparent information and structured knowledge is to include terminology work right from the start in innovative processes, set up term creation rules based on the individual in-house classification system.

4 Demands on Organisational / Corporate Knowledge

As there are normally several or even many people, different departments, possibly different subsidiaries or contractors in various countries with cultural differences and several languages involved, information, documentation and communication has to meet various requirements [2], [6]:

It must be

- easily accessible (technically and as necessary in different languages)
- up-to-date
- comprehensive
- clear
- comprehensible
- traceable

4.1 Need for Metadata

Knowledge and information can only be tracked, referenced and managed if

- the idea behind a product or service is explained
- a definition of what separates this function, product or service from others is given and
- the designation(s) in the languages needed are recorded and contain sufficient meta data.

A knowledge base in the form of a terminological database can not only contain terms, but also illustrations, multi-media-files, symbols and icons – with the required meta data for correct use. Terminology forms an integral part of organisational learning, as it contains key information, on, for example, the following topics:

- Track changes - Who created the entry or term and when; Who changed the entry or term and when; Who created a certain field and when; Who changed a certain field and when.
- Usage - Which term is preferred or even prescribed or forbidden, for which application, in which type of text, for which product, company, etc.
- Locale-specific variances - Which term or spelling is used in which language and country.e.g. color (USA) vs. colour (UK)
- Geopolitical and cultural issues in Localisation - e.g.: Take great care when using flags, i.e. the flag of Taiwan for information in China

While in Germany, every do-it-yourselfer keeps a locksmith's hammer [figure 13] in the toolbox, the Anglo-Saxon counterpart has traditionally a claw hammer [figure 14] not only to hit nails into the wall, but also to remove them as necessary.

Accordingly, people from both sides of the Channel have different concepts of a hammer. And a German who is asked to use a hammer to remove a nail, will be amazed and grab for a set of pliers instead of a hammer.

5 Conclusions

A learning organisation [2], [6] must enable everybody involved in development, innovation and learning to have easy and quick access to relevant information, to have all important facts available (including meta data), no matter in which language, for which country, which product/service is concerned, to track history of a product/service and thus to prevent duplication efforts, repetition of mistakes and higher costs. Moreover, terminology, terminology management and the use of the reliable terminology resources guarantee quality of a product, protect customers and save costs [3], [7].

- Terminology helps to guarantee the quality of products and processes

- All companies and organisations produce terminology in written and oral form.
- Terminology is a part of the process and, also, of the product.

- Terminology management is crucial at the production of source texts (corporate knowledge) and technical documentation and should be implemented throughout the process (see figure 9).
- All stakeholders in an organisation or company should be included in terminology policies and planning.
- Terminology is an asset for any company helps it to stand out from competitors.

- Terminology helps to protect customers

- Clarity and safety are key premises in technical documentation – not only in the medical industry.
- Bad terminology management can influence the quality of the product and lead to legal claims.
- Technical documentation and, as a consequence, terminology are part of the product.
- Quality assurance is a key issue in the industry and technical documentation.
- Wrong terminology in technical documentation can cause damages to employees, customers, users, etc.

- Terminology management saves costs

Terminology management in technical documentation can lead to:

- 5% reduction of the translations costs,
- 10% reduction of the general costs through 100% matches in translation memories,
- 10% reduction of the work-load,
- 50% less in translation work,
- reduction of 60% in the translation questions and queries.

(Schmitz und Straub, 2010)

Competitive business environments with dynamic development processes (i.e. SRCUM method) make it necessary to constantly maintain and update corporate/organisational knowledge and to make it accessible for changing users and user groups. A key factor for the long-lasting success of learning organisations is professional terminology work.

References

1. Arntz, R., Picht, H.: Einführung in die Terminologiearbeit (Introduction to terminology work). Georg Olms Verlag, Hildesheim (1991)
2. Biro, M., Messnarz, R.: Success Factors for Business Based Improvement. Software Quality Professional 2(2) (March 2000)
3. Budin, G., Swertz, C., Mitgutsch, K.: Knowledge Organization for a Global Learning Society (2006)
4. Edwards, K.: The importance of content inventories. In: MultiLingual, Annual Resource Directory (2012), `http://www.multilingual.com`

5. Galinski, C., Reineke, D.: Vor uns die Terminologieflut. In: eDITion, vol. 2 (2011)
6. O'Keeffe, T.: Towards Zero Management Learning Organisations (2006)
7. Zenk, W., Brändle, D.: Terminology to suit the master plan. In: tcworld (April 2010), http://www.tcworld.info/tcworld/technical-communication/article/terminology-to-suit-the-master-plan

Process Improvement for the Small and Agile

Marion Lepmets[1] and Tom McBride[2]

[1] Public Research Centre Henri Tudor, 29 J.F. Kennedy, Luxembourg
marion.lepmets@tudor.lu
[2] University of Technology, Sydney
15 Broadway, Ultimo, New South Wales, Australia 2007
tom.mcbride@uts.edu.au

Abstract. In order for process improvement to better respond to industry expectations there needs to be a value-centric thinking about processes so that all processes in the organization are contributing to the creation of value for the customer and for the organization. To achieve this, organization's strategic goals need to be aligned with process goals on operational level so that all work contributes to the strategic goals of the organization. Similarly, the performance measurement system needs to be revised to direct efforts toward reaching the broad goals of the cross-functional processes rather than small tasks and compliance to set rules. In this paper we describe potential problems process improvement has today and discuss possible concepts that might help revising the process improvement thinking. These changes clearly reflect the needs and characteristics of small and very small organizations.

Keywords: process improvement, goal alignment, systems thinking, value-creation system, self-organizing teams, flexibility, VSEs.

1 Potential Problems of Process Improvement Today

Process improvement carries with it the promise of increasing productivity through better quality, thus reducing or removing the costs of rework. Today, it is also expected that it helps changing the organization for the better through motivating people in having a shared vision [1] and aligning the organization's goals to the operational level goals of processes [2]. Unfortunately, this expectation has not turned into reality and therefore the concepts of process improvement in its current form need to be revisited and reconsidered to better respond to the changed expectations [3].

The term "process improvement" implies that it is the process itself that is improved, denying the possibility that the improvements in developing software or delivering services could arise from changes that lie outside the scope of processes. Naturally this depends on how a "process" is defined. The most common definition of a process is an interrelated set of activities that uses resources to transform inputs into outputs [4]. Such a definition concentrates on the transformation activities, the work practices themselves to the exclusion of contributions that can be made by the

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 310–318, 2012.

organization structure, the production ecosystem, the competence of the process performers [5] or the goal alignment between organizational goals, process performers and the process itself [2].

This paper describes the underlying concepts of process improvement, how these concepts might not respond to the industry needs today, and the challenges process improvement faces in order to change and better respond to the industry expectations. These changed expectations are corresponding well to the characteristics of micro, small and medium sized enterprises (SME) [6] that need to be agile, adaptable and rely heavily on the competence and skills of their staff rather than the process descriptions [7]. As 85% of Europe's IT sector companies employ from 1 to 10 employees [8], process improvement should better respond to the needs of these micro organizations [6] also known as Very Small Entities (VSE), and learn from their characteristics.

The paper is comprised of three sections, where the first one describes potential problems process improvement is facing today. The second section provides a discussion on possible solutions to these problems in terms of concepts from various domains that could help change process improvement to better respond to industry expectations. Finally, we propose a set of research problems for further investigations.

1.1 How Organizations Are Viewed

Most organizations today still have a mechanistic view of their operations that treats the organization as a type of factory, leading to a high level of rigidity and inertia [9, 10], in denial of the ever-present internal and external dynamics, complexity and the associated uncertainty [11]. Such a view does not recognize or understand that organizations are systems and, consequently, do not realize the implications for how organizations should be planned, designed and managed, which in turn means the value of process concepts continues to be misconstrued and misapplied [12].

Departments of a larger organization can be viewed as SMEs or VSEs, with some autonomy over their activities but with obligations to share the strategic vision of the organization and, therefore, a greater obligation to arrange their activities to achieve those goals. When the VSEs are roughly the size of a single team (10 or less) then their manager and the project manager can be regarded as equivalent. It would be their task to determine and communicate the client's goals to the team, preferably with the involvement of the team, to determine how those intentions are to be achieved.

As environmental uncertainty increases organizations should become more organic [13, 14], decentralizing authority and responsibility closer to the point of application [15], encouraging employees to take care of problems by working directly with one another, encouraging teamwork, and taking an informal approach to assigning tasks and responsibility [11]. Complexity, dynamics and the associated uncertainty necessitate the organizations' ability to adapt and change, and the creativity to bring that about is crucial for these organizations [5].

An organization is a collection of interactions between its four highly interrelated parts – people, structure, tasks and technology that work together towards a common goal [13]. From this socio-technical system perspective people perform tasks to produce goods and services with the help of technology that provides the infrastructure to perform these tasks, and an overarching organization's structure that includes communication, authority and workflow systems that operate within the organization. In this perspective an organization functions as a socio-technical system where the technical system includes technology and tasks performed to achieve organizational goals, and the social system is comprised of people and their roles, competence, knowledge and skills. These systems are interdependent – what affects one affects the other [13], so it is conceivable that improving the competence of the people [16], improving the technology to be used in the process (automated testing, better compilers, issue tracking systems etc.) or changing the structure of the organization could lead to an overall improvement. It is possible to argue that people, technology and structure are all part of the process, but that would require a much broader definition of process than is prevalent in ISO/IEC 15504 (SPICE) [17] and Capability Maturity Model (CMMI) [18].

1.2 Goal Alignment and Goal Orientation

The process improvement specialists often go to the organizations with a conviction that an increase in process maturity based on an appropriate process reference model will help solve any problem the organization might face. While there might be some truth to it, we do not have a clear mapping of which aspects of the organization will be impacted as a result of improving certain processes. This could also be true if the most pressing problem of all organizations was quality but absolute software quality is not among the list of priorities for software development according to recent reports [3]. Instead there is an assumption that software development whether for mashups, service oriented development of applications (SODA) or application development will achieve an acceptable level of quality. It isn't a challenge any longer but an expectation. The challenge for processes is whether they can support and achieve organization's goals and priorities [14].The current process improvement methods (CMMI and SPICE) aim to reduce the variation and unnecessary errors in processes to improve the quality of the product. At the same time, most organizations today know how to run their operational processes as software engineering has long been systematically taught at schools. Instead of looking solely at the operational level process fine-tuning, we should look at what really distinguishes the successful organizations from the less successful ones. Instead of looking at the operational level processes we should look at the alignment between the goals of different organization's levels and the adaptability of the organization, through governance, shared vision and self-organizing teamwork. Successful organizations show an adaptive, opportunistic development of strategy where instead of brilliant strategic planning there is experimenting, trial and error, opportunism and coincidences [5].

Despite the technological resources of support, human beings are ultimately responsible for realizing organization's goals. Process improvement as external-criteria based activity has long been troubled with the lack of goal alignment and goal-orientation. Weiss [19] argues that goal-orientation helps to reinforce the feeling within each organization that the recommended improvement will create benefit, and eliminates much of the resistance to change that comes from long explanation and discussion of the external assessment criteria. The goals of processes and accordingly the measures for the performance of these processes should be aligned with organization's strategic goals. According to Senge, most organizations have visions that are visions of one person (or one group) imposed on the organization that, at best, command compliance and not commitment [20]. A shared vision is a vision that many people are truly committed to, because it reflects their own personal vision. Hammer concurs with that saying that organizations should emphasize teamwork, personal accountability and customer's importance, and develop their people rather than supervise them to tackle complex projects and know how to redesign processes that support organization-wide change [21]. This change should not only be about small process changes [22] but in enablers that determine how well processes will perform over time, supported by the organizational capabilities that help organizations to put these enablers to place [21] adapted to organization's characteristics [23]. Lots of organizations use techniques such as Six Sigma [24] and Total Quality Management (TQM) [25] to ensure that employees execute processes correctly but the dramatic improvement of process performance and process innovation can only be done through redesigning the processes to eliminate many nonvalue-adding activities that are the source of costs, errors and delays. Together with the redesign of the process, the organization should change the performance measurement system that helps assess the performance of the new processes overtime and reward people for focusing on broad, common goals of the organization.

2 Discussion on Potential Solutions

In order to solve some of the potential problems that process improvement faces today we must consider the involvement of people in decision-making and creating a shared vision which supports the direction and motivation of self-organizing teamwork to have the responsibility for what they do and the trust to do it. These considerations illustrate the characteristics of small and agile organizations and we should consider them to revise how process improvement is carried out. There needs to be a value-centric thinking about processes where nonvalue-adding activities should be eliminated and all processes in the organization contribute to the creation of value for the customer or for the organization. In order to achieve this, strategic goals need to be aligned with process goals on operational level so that everything that is done contributes to the strategic goal of the organization. The measurement system needs to support a reward system for reaching the broad goals of the cross-functional processes rather than small tasks and controlled compliance to set rules.

Paradigms of software development and service delivery have moved away from the command and control paradigm toward the distributed decision paradigm. Where the earlier paradigm assumed a need to specify the methods with which goals were to be achieved, current paradigms typified by agile methods tend to assume that those performing the tasks are competent but need to know the goals they are to achieve. This is particularly true of VSEs because they provide specialist services or specialized skills than can be achieved within a large organization. Because VSEs have highly autonomous teams, these teams needs to clearly identify the goals they are to achieve, then use their own initiative and competencies to achieve them [26]. In these circumstances improvement is more likely to come from better ways of identifying and communicating the client's goals and from developing general competencies rather than from following static process descriptions. Since the team is likely to have to adapt rapidly to the specific circumstances, specific and highly optimized work practices are unlikely to be useful in all circumstances.

2.1 Systemic View of Organizations

Unlike the mechanistic way of organizing, the *organic* way is directed to adaptation, flexibility and the ability to change in light of unforeseen requirements. This is important when an organization is dealing with strategic change [13, 14]. Contrary to the detailed task descriptions, rules, regulations and targets, there is employee freedom based on their competence to act, hence there is room for employee-initiated behavior and creativity. Instead of hierarchical control, focus is on employee involvement that is essential for productivity, quality, service and organizational learning and innovation, also establishes the ability of self-organizing. *Self-organizing teams* stimulate participation and involvement, an effect of this is increased emotional attachment to the organization, resulting in greater commitment, motivation to perform and desire for responsibility [27]. A transition from teams that consist of independently focused self-managing individuals to a high level of individual and group autonomy is needed that will lead to the self-organizing teams where creativity and flexibility is cherished [27]. These teams are given significant authority and responsibility for many aspects of their work, such as planning, scheduling, assigning tasks to members, and making decisions with economic consequences. Self-management brings decision-making authority to the level of operational problems and uncertainties and, thus, increases the speed and accuracy of problem solving [28].

Pourdehnad et al. go further and say that even the organic view does not necessarily capture the purposeful nature of organization's parts and therefore a *systemic* approach should be used that provides a formal awareness of the interactions of a system's parts, recognizing the purpose as the most critical classifying variable used in distinguishing social systems from other types of systems [29]. By engaging all stakeholders in creating a shared vision like suggested by Senge, systems thinking techniques get the entire organization to belong and commit to the change giving its members more freedom and the opportunity to learn from their own mistakes as opposed to being closely supervised [20, 21].

Lean management, as one of the systems thinking approaches, could be considered for better goal alignment together with Lean performance, one of the Lean methodologies. Lean performance is a management strategy that focuses on customer and organization's value, process-orientation, teamwork, motivation to continuous process improvement and measurement of this improvement [30]. Lean performance aims to deploy the policies and strategies regarding the company's mission, markets, products and services to a process level.

The *value creation system* sees any organization like a system operating as part of a super-system that consists of markets, competition, resources, and the general business environment. All activities and complexities of organizations fundamentally produce something of value to someone other than themselves. The systems thinking approach sets the process in the heart of the value creation system, where every modeling, improvement, or management effort of a process must be a contribution to that total value creation system. The addition of a feature to a process that does not add value from the customer perspective is not improving performance of the organization [30].

In order to support an organization's software development needs, the methodologies need to be nimble rather than highly capable, quickly able to respond to the organization's demands to enter a new market or adopt a new technology or respond to a new threat. In this environment, software development must be capable of producing software that is "good enough" [31] while *flexible* enough to rise to almost any challenge. From dynamic capabilities perspective, people (resources and their competencies) form the basis of unique value-creating strategies, and their creation, evolution, integration and recombination generate new value-creating strategies [32]. In high-velocity markets, organizations rely extensively on new knowledge created from new situations, experiential activities like prototyping, real-time information, multiple options, and experimenting. In order to be adaptable, routines are iterative and cognitively mindful instead of linear and mindless.

As happens so often, military thinking has confronted the same challenges of achieving an objective in a rapidly changing and ambiguous environment. Current military doctrine is to rely on "Commander's Intent" to clearly convey the commander's intention but to leave subordinates able to exercise their initiative in light of the circumstances at the time [11]. The concept of "Commander's Intent" has been adopted by Agile community where the overall plan is the clear, concise, focused statement of intent. When the circumstances change and the plan has to change, the development team alters the plan with the intent in mind [26].

The performance measurement system needs to change together with the changed processes in order to reward people for working towards broad, common goals of the organization. Hammer proposed an intriguing framework for business processes that could usefully be adapted for software development, since both have the same goals of high performance [21]. Hammer's framework is in two dimensions; process enablers and organizational capabilities. The process enablers do not form a reference model but could form a measurement scale of process capability that includes not only the process design but process performers, process owners, process infrastructure and process metrics. The CMMI and SPICE process capability measurement scale addresses many of the same concerns but not in such detail and not so clearly mapped to different levels of achievement. Where Hammer departs from the SPICE and CMMI models is his explicit dimensions and scale of organizational capabilities that cover leadership, culture, expertise and governance. The equivalents in ISO/IEC 12207 [33] and ISO/IEC 15288 [4] have been labelled "Supporting Processes". All of these dimensions are known to affect software development and there has been a lot of discussion on their importance. Hammer's scale is the first to articulate a systematic approach to these dimensions in such a way that can be tested to determine the scale's validity. That could take some time. In the interim, pragmatic organizations could consider whether and how these organizational capabilities can boost process performance where other process improvement methods are achieving diminishing returns.

3 Summary and Potential Research Problems for Future

The current process improvement practices are often referred to as enforcing rigidity in organizations. Although we believe that process improvement helps reduce errors in operational practices, additional research could be carried out to understand exactly which practices and underlying concepts in process improvement and TQM compromise agility and what can be done about it.

As a response to frequently changing customer requirements the operational processes are redesigned on the fly in agile teams which maximizes the adaptability of the processes and the work the team conducts. Similarly the project management approaches based on the traditional linear development methodologies are mismatched with the real-world dynamic software development efforts where organizations have to adapt quickly to changing technology, markets and social conditions [11]. Agile methodologies in general and SCRUM in particular, achieve this dynamic process tailoring through reviewing their current state, desired state and what they need to do to get from the current situation to the desired one. Techniques are considered and chosen to achieve the goal as much as possible. Schedules might be changed, personnel might be assigned new or different tasks. All that is being done with customer value in mind instead of following the pre-defined and appointed tasks and activities suggested by process owners. We would recommend conducting further research into the underlying workings of Scrum to understand if goal internalization is a result of frequent meetings where goals are revisited as requirements change and actions being taken based on the expertise of the developers. We would like to understand how do people know where they are in relation to their goals and what do they do if what they are doing is not achieving their process or project goal?

We would also suggest researchers to apply systems thinking approaches like Lean to help process improvement approaches better respond to the problems of the organizations and take the processes out of the isolation into being an interconnected piece in the big system called the organization. In this system, how can we describe processes so that they would support achieving a range of goals depending on the uncertainty of the task?

When the organization has changed its processes and expects its members to focus on broad goals, the performance measurement system should adapt accordingly. Additional work into developing a performance measurement system that would reward employees to focus on broad cross-functional process goals that bring value to both the customer and the supplier is a challenging research problem to solve. This measurement scale could reflect Hammer's process enablers and organizational capabilities with the aim of diversifying the CMMI and SPICE capability scales.

Acknowledgments. The present project is supported by the National Research Fund, Luxembourg and cofounded under the Marie Curie Actions of the European Commission (FP7-COFUND).

References

1. Lepmets, M., Ras, E.: Motivation and Empowerment in Process Improvement. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 109–120. Springer, Heidelberg (2011)
2. Lepmets, M., McBride, T., Ras, E.: Goal Alignment in Process Improvement. Journal of Systems and Software, 20p (2012)
3. Finley, I., Valdes, R., Driver, M., Murphy, T.E., Prentice, B., Wilson, N., Norton, D., Altman, R., Smith, D.M., Murphy, J.: Agenda for Application Management. Gartner (2012)
4. ISO/IEC 15288: Systems engineering - System life cycle processes. 15288 (2008)
5. Hoogervorst, J.A.P.: Enterprise Governance and Enterprise Engineering. Sogeti Netherlands, Diemen, Netherlands (2009)
6. EC: Definition of micro, small and medium-sized enterprises (2003), `http://europa.eu/legislation_summaries/enterprise/business_environment/n26026_en.htm`
7. Alexandre, S., Renault, A., Habra, N.: OWPL: A Gradual Approach for Software Process Improvement In SMEs. In: 32nd EUROMICRO-SEAA 2006 Conference on Software Engineering and Advanced Applications, p. 8. IEEE Computer Society, Cavtat (2006)
8. O'Connor, R.V., Basri, S., Coleman, G.: Exploring Managerial Commitment towards SPI in Small and Very Small Enterprises. In: Riel, A., O'Connor, R.V., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 268–279. Springer, Heidelberg (2010)
9. Benner, M.J., Tushman, M.L.: Exploitation, Exploration, and Process Management: the Productivity Dilemma Revisited. Academy of Management Review 28, 238–256 (2003)
10. Sitkin, S.B., Sutcliffe, K.M., Schoreder, R.G.: Distinguishing control from learning in total quality management: A contingency perspective. Academy of Management Review 19, 537–564 (1994)
11. Augustine, S., Payne, B., Sencindiver, F., Woodcock, S.: Agile project management: steering from the edges. Communications of the ACM 48, 85–89 (2005)
12. Rummler, G.A., Ramias, A., Rummler, R.A.: White Space Revisited: Creating Value Through Process. John Wiley & Sons (2009)
13. O'Hara, M., Watson, R.T., Kavan, C.B.: Managing the Three Levels of Change. Information Systems Management 16, 63–70 (1999)
14. Nerur, S., Cannon, A., Balijepally, V., Bond, P.: Towards an Understanding of the Conceptual Underpinnings of Agile Development Methodologies. In: Dingsøyr, T., Dybå, T., Moe, N.B. (eds.) Agile Software Development, pp. 15–29. Springer, Heidelberg (2010)
15. Woodward, J.: Management and Technology. In: Pugh, D.S. (ed.) Organization Theory. HMSO, London (1958)
16. Humphrey, S.W.: Team Software Process. (2000)
17. ISO/IEC 15504-2: Information technology - Software process assessment - A reference model for processes and process capability, 15504, vol. 2 (2004)
18. CMMI v1.3: CMMI® for Development, Version 1.3. Software Engineering Institute (2010)

19. Weiss, D.M., Bennett, D., Payseur, J.Y., Tendick, P., Zhang, P.: Goal-Oriented Software Assessment. In: International Conference on Software Engineering, pp. 221–231 (2002)
20. Senge, P.M.: The Fifth Discipline: the Art and Practice of the Learning Organization. Random House, London (1990)
21. Hammer, M.: The Process Audit. Harvard Business Review, 17 (April 2007)
22. McBride, T., Lepmets, M.: The Many Forms of Process Improvement - Results of an International Survey. Software Quality Days, 7 (2012)
23. Tuisk, A., Karpištšenko, A., Lepmets, M.: Integrated Process Improvement Approach: Case Studies in Skype Technologies Ltd. In: Mas, A., O'Connor, R.V., Mesquida, A.L. (eds.) SPICE 2012. CCIS, vol. 290, pp. 13–25. Springer, Heidelberg (2012)
24. Schroeder, R.G., Linderman, K., Liedtke, C., Choo, A.S.: Six Sigma: Definition and underlying theory. Journal of Operations Management 26, 536–554 (2008)
25. Tennant, G.: Six Sigma: SPC and TQM in Manufacturing and Services. Ashgate Publishing (2001)
26. Lapham, M.A.: DoD Agile Adoption: Necessary Conditions, Concerns and Changes. Crosstalk, 31–35 (January/February 2012)
27. Moe, N.B., Dingsøyr, T., Dybå, T.: Understanding Self-organizing Teams in Agile Software Development. In: 19th Australian Conference on Software Engineering. IEEE Computer Society, Perth (2008)
28. Tata, J., Prasad, S.: Team Self-management, Organizational Structure, and Judgments of Team Effectiveness. Journal of Managerial Issues 16, 248–265 (2004)
29. Pourdehnad, J., Bharathy, G.K.: Systems Thinking and Its Implications in Organizational Transformation. In: 3rd International Conference on Systems Thinking in Management, Philadelphia, PA (2004)
30. Carroll, B.J.: Lean performance ERP project management: implementing the virtual supply chain. St. Lucie Press, Boca Raton, Florida (2002)
31. Bach, J.: Good enough quality: beyond the buzzword. Computer 30, 96–98 (1997)
32. Eisenhardt, K.M., Martin, J.A.: Dynamic capabilities: what are they? Strategic Management Journal 21, 1105–1121 (2000)
33. ISO/IEC 12207: Information technology – Software life cycle processes. ISO/IEC 12207 (2008)

A Practical Approach to Project Management in a Very Small Company

Edgar Caballero and Jose A. Calvo-Manzano

Departamento Lenguajes y Sistemas Informáticos e Ingeniería del Software
Universidad Politécnica de Madrid,
28660 Boadilla del Monte, Madrid, España
ecaballero@bolesfactory.com,
josentonio.calvomanzano@upm.es

Abstract. This article shows how a very small company has tailored Scrum according to its own needs. The main additions made were the "sprint design" phase and the "sprint test" phase. Before the sprint 0, the requirements elicitation and the functional specification were made in order to meet deadlines and costs agreed with clients. Besides, the introduction of an agile project management tool has supported all the process and it is considered the main success factor for the institutionalization of the Scrum process.

Keywords: Scrum, process improvement, very small company, project management.

1 Introduction

Due to the global economic crisis, organizations are forced to adapt their business strategies in order to stay in the market [1]. Spain and particularly the software industry are not the exception. Moreover, an inappropriate quality and project management in software organizations generate cost overruns, low quality and cancelled projects [2]. Organizations must improve their competitiveness through improvements in productivity in order to survive in a market weakened by the crisis [1].

At this scenario, organizations need a more efficient resources management. The goal is producing more in less time with the same or less cost is a key factor. A process improvement is necessary to achieve this goal [3], but a very small organization does not have enough time or resources to invest in it [4].

Several process improvements (e.g. "COMPETISOFT" [5]) have been proposed for small organizations based on defined process control [6] and empirical process control [7]. Agile methods are based on empirical process control and provide a good performance in small teams [7]; therefore they are suitable for process improvements in very small organizations.

This article shows how a very small company has implemented an agile method such as Scrum and how it was adapted, supported by an agile management tool, according to its own needs.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 319–329, 2012.

In the following sections it will be described the context, the tailored process and the agile management tool. Finally, conclusions will be summarized.

2 The Context

2.1 The Organization

Bolesfactory is a very small company of software development with a staff of 14 people [8].

At 2010, due to the actual crisis, the organization started to get worried about productivity and quality. In order to improve them, the organization got interested in agile methodologies and decided to introduce one of them. After evaluating several alternatives (Scrum, Extreme Programming, Kanban), the management decided to use Scrum.

For the past 3 years the organization has been working with Scrum and has some lessons learned about it, mainly with the problems detected.

Initially Scrum was applied without any change, but after some experiences, the organization began to make adjustments according to the needs that were emerging.

As in previous experiences with other methodologies (e.g., Team Software Process), productivity was affected by the lack of specialized support tools in the implementation of the process. For this reason, the organization decided to use an agile project management tool that achieves the expected levels of productivity by improving the visibility of the project.

2.2 Scrum

Scrum is an agile process framework that allows organizations be focused on business value through the frequent and regular delivery of high quality software [8].

Scrum is based on an empirical process control model rather than the traditional defined process control model, which regularly inspects activities to monitor what is happening and adapts them to produce the desired and predictable outcomes [9].

Scrum allows the team to apply any specific method or technique oriented to the software development.

The main criteria for selecting Scrum were:

- Several characteristics and principles were similar to the previous iterative process of the organization (Team Software Process).
- Hypothesis: Scrum increases productivity and reduces time to benefits [10].
- Scrum leads as the most adopted agile methodology [11].

Scrum implements an iterative and incremental process which involves three stakeholders: the Product Owner, the Team, and the ScrumMaster.

The Scrum process defines an initial preparation phase and several iterations called sprints (see Fig. 1). A sprint is a 2-4 weeks period of development time and 4 meetings are held: planning, daily, review and retrospective meetings.

The requirements are collected and prioritized in the product backlog, which is decomposed in tasks on the sprint backlog.

The planning meeting is decomposed in 2 sessions. The first session is focused on explaining the sprint scope, and the second session is focused on tasks identification and effort estimation.

During the execution of each sprint, the team meets daily in the 15-minute meeting to track the work progress answering three questions: What have I done since the last Scrum meeting? What will I do before the next Scrum meeting? What prevents me from performing my work as efficiently as possible? [10].

Fig. 1 shows the Scrum process.

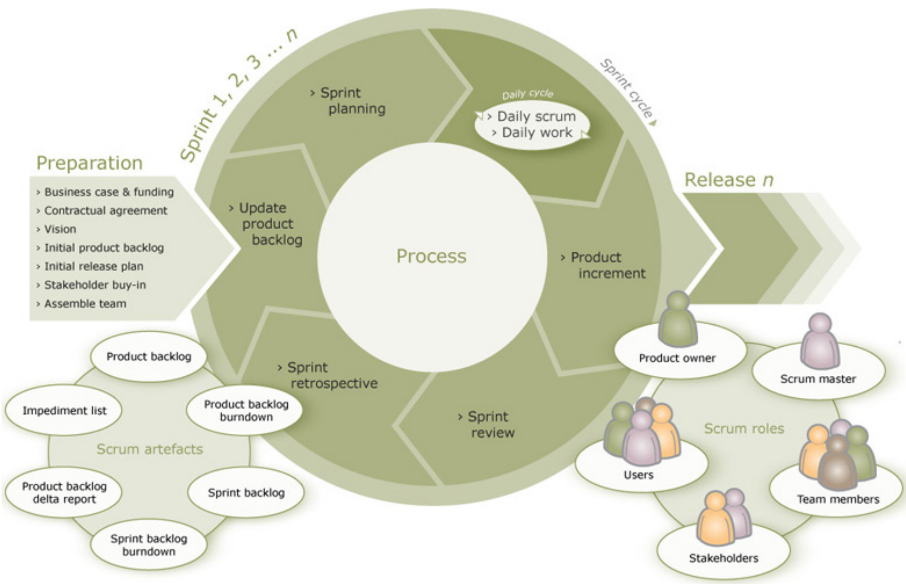


Fig. 1. Scrum process

2.3 Agile Management Tool

The tool selected by the organization was "Target Process" [12]. Its usability and flexibility were the main criteria for the choice.

Target Process is an agile project management software designed with simplicity in mind, Target Process helps software development companies to reduce the complexity of software project management and simplifies planning, tracking and quality assurance activities.

Target Process is a customizable tool. You can create a development process with customizable practices, workflows, terminology and customized fields.

3 Tailoring Scrum in the Organization

After introducing Scrum in some projects, the team members were motivated with the Scrum project management strategy. However the organization detected several problems such as:

- A high number of changes because a low initial requirements definition.
- A low quality in the product during the “sprint review”.
- An unstable product after each sprint caused by the incremental strategy.
- Many wasted hours in getting the project graphics and reports.

In order to solve the previous problems, the organization began to define additions that will be described in the next sections.

3.1 The Tailored Process

In this section, the sprint flow will be described in order to understand the tailored process. Fig. 2 shows the sprint flow highlighting the additions.

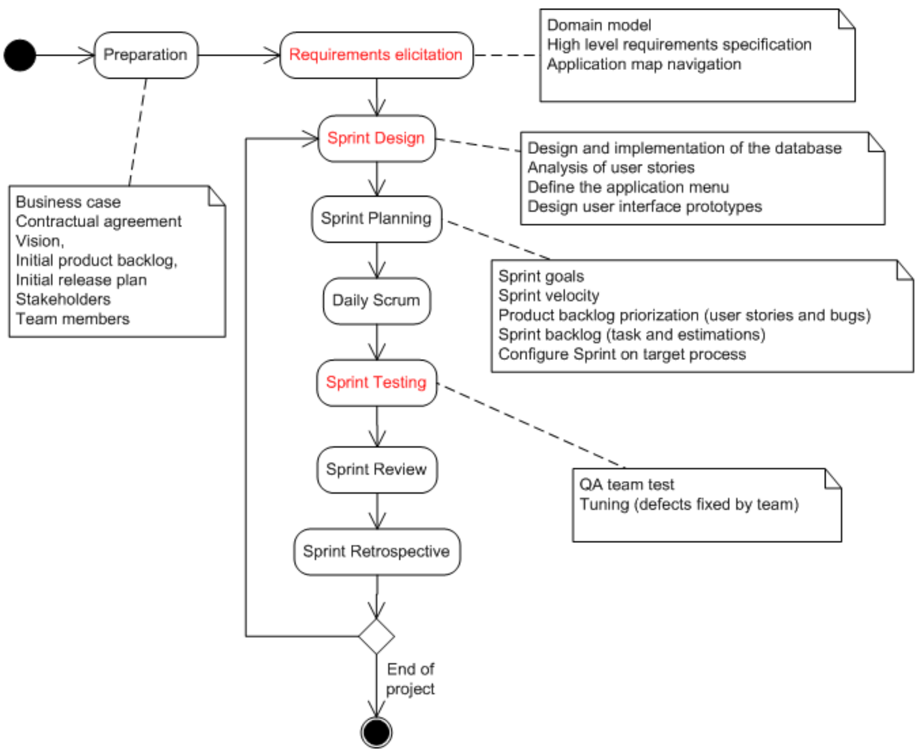


Fig. 2. Sprint flow chart

As shown in Fig. 2, “requirements elicitation”, “sprint design” and “sprint testing” are the additions introduced in the Scrum process.

After the “preparation phase”, the process continues with the “requirements elicitation”. Its goal is to get the domain model, the high level requirements specification and the application map navigation in order to improve the requirements understanding and their cohesion.

The iterative process starts with the “sprint design”. It is focused on increasing the knowledge for improving the estimations during the “sprint planning”, and it includes the data base implementation, the analysis of user stories, the menu application and the main user interface prototypes.

Then the team continues with the “sprint planning”, in which only the second session takes place (tasks identification and effort estimation). The first session with the “product owner” is moved to the “sprint design”.

After that, the “daily scrum” begins. It includes the coding phase and the tests performed by the development team.

Before the “sprint review”, the “sprint test” is started. At this point, the quality team performs the integration test and the product verification. If there is time, the development team uses this time to fix major defects found by the quality team.

Finally, the “sprint review” and the “sprint retrospective” are done without any modification as described by Scrum.

Fig. 3 shows the tailored Scrum process.

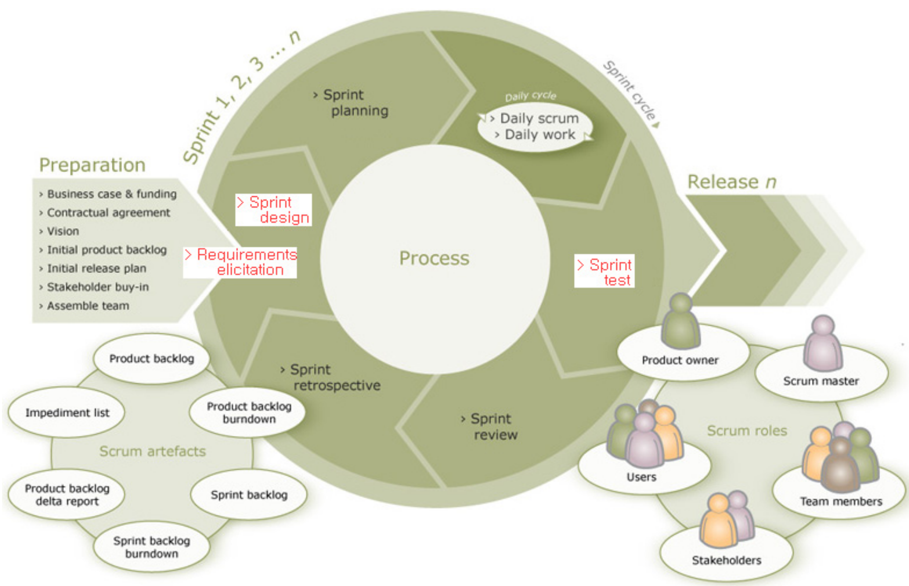


Fig. 3. Tailored Scrum process

3.2 Addition 1: The Requirements Elicitation in the Preparation Phase

The preparation phase (also called sprint 0 or pre game phase) organizes and defines all project needs for starting the Scrum iterations (sprints), but Scrum does not say how to do it. Its goal is to stablish the project scope delimitation and it contains the business case, the contractual agreement, the vision, the initial product backlog, the initial release plan, the stakeholders and the team members.

According to the organization experience, the preparation phase and the first planning session were not enough to get a good understanding of the project, the requirements individually were stable, but low cohesion between them forced to make frequent changes to the initial definition of requirements.

Scrum is only a project management framework and it does not specifies a requirements management method. Each organization has to select one.

At this scenario, the organization decided to introduce a requirements elicitation method in order to get a high level functional specification which includes the domain model, the high level requirements specification and the application map navigation.

This high level functional specification has two goals:

- Improve the requirements cohesion.
- Reduce the deviation of the project in order to meet deadlines and costs agreed with clients.

Based on this functional specification, the prioritized product backlog is created.

3.3 Addition 2: Sprint Design

Scrum suggests that the tasks of analysis, design, implementation and testing are performed during the execution of the sprint, but the decision on how to implement them is up to the development team. In this sense, the organization decided that the analysis and design were implemented before the "sprint planning" meeting in order to improve the requirements knowledge and reduce the estimate deviations.

"Sprint design" is approached as a systematic activity to improve the knowledge of the project before starting to code. Like others scrum activities, the "sprint design" must be a time box activity, in the case of a 4 week sprint, the "sprint design" should last less than 3 days.

The "sprint design" has the following tasks:

- Design and implementation of the database.
- Analysis of user stories.
- Define the application menu.
- Design the user interface prototypes.

Therefore the "sprint design" is not a scrum modification, but a specific addition detailing an aspect not described by the methodology that may or may not be used by another organization.

The user interface prototypes are an excellent tool for an early user validation [12]. Fig. 4 shows an example of a user interface prototype (left side in Fig. 4) designed with Balsamiq Mockups [13], which is very similar to the final implementation (right side in Fig. 4). Their main advantage is the short time needed for its design.

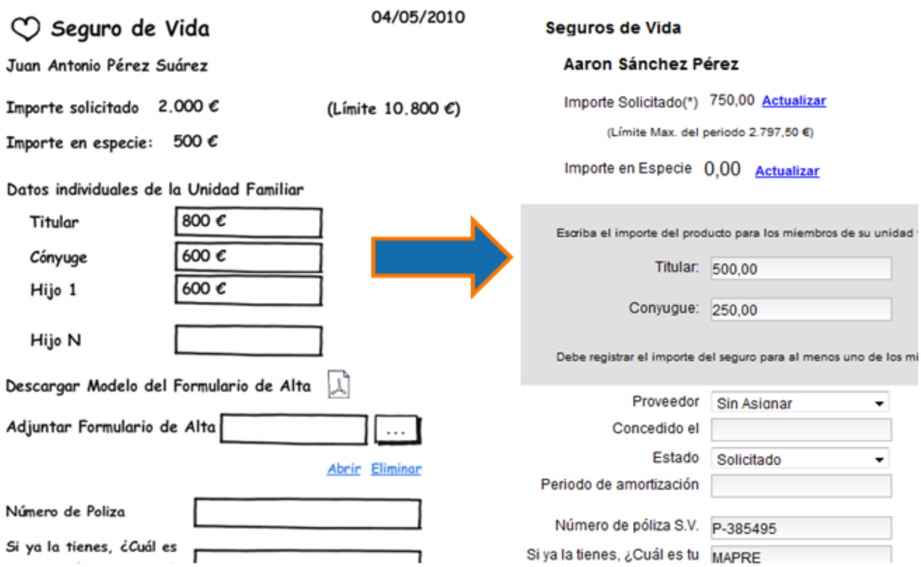


Fig. 4. User interface prototype

3.4 Addition 3: Sprint Test

The product quality after each sprint was not satisfactory. According to the good practices described in [15], the organization decided to create a quality team in order to test the user histories. After some sprints, it was observed in the “sprint review” that the user stories implemented at first had a better quality than the others implemented at the end, especially those that need an integration job.

The conclusion reached was that the user stories with bad quality were not tested enough, mainly because the sprint end date was reached, and often the team changed to “done” the user stories state when in fact they were not. Initially it was decided to allocate more time for the integration testing during the sprint, but it turned curiously detrimental from the standpoint of the project monitoring because the estimated effort for the integration testing distorted the project status during the early days of the sprint. This small and insignificant detail became a factor of discouragement for the team because the sprints ended with delays that were not identified until late in the sprint. To solve this problem, integration tests were separated from development by creating a new phase called “sprint test” to be performed before the “sprint review”. The effort allocated for this activity is 20% of the effort that was initially assigned to each sprint.

The “sprint test” is performed by the quality team. During this phase, the development team can also fix the higher priority defects in order to improve the product quality at the “sprint review”. This fixing activity is called “sprint tuning” and must be adjusted to the time box established for the “sprint test”. The “sprint test” improves product quality but reduces productivity because it is a failure quality activity [16].

4 The Support Tool for Agile Project Management

The tool selected by the organization was "Target Process" [12]. Its usability and flexibility were the main criteria for selecting it. It is free for small teams (upto 5 members). In the next sections the activities supported by the tool will be described.

4.1 Change Management

Although in agile methodologies, the change management process is open and does not require formal approval, it is important to have at least a record of change requests and whoever has done. With "Target Process" the user is the one that registers the request through an email, the tool automatically classifies the request and finally the "product owner" updates its priority.

4.2 Bugs Management

"Target Process" allows tracking all the defects found by the quality team or users. The users can add, prioritize, plan iterations and view quality reports. In the bugs list, users can change bug states, assign bugs to developers and testers and edit bug (change severity, release/iteration, effort, etc.).

4.3 User Histories Management

"Target Process" allows storing requirements as features or as user stories. The user stories can be created on the fly or be imported from a .csv file. A user story could be assigned to two team members in the case of pair programming. One interesting feature is the storage of the initial estimation in order to analyze effort deviations.

4.4 Impediments Management

An impediment (or block) is an obstacle that prevents the sprint completion. "Target Process" provides the functionality needed to manage impediments. It allows adding impediments to user stories, tasks and bugs in order to make them visible for all team members.

4.5 Sprint Planning and Load Balancing

The planning and load balancing is one of the main characteristics in "Target Process". For planning, the user must establish the team members, their velocity, the start day, the duration and the sprint velocity. After that, the tasks are identified for every user story, the team members are allocated and the work load is estimated. Then, from the load balancing panel the user verifies if the user stories and bugs selected for the current sprint are according to the sprint velocity. If the estimated effort that is required for the sprint does not exceed the sprint velocity, the work load of the team members must be verified. If some of them are overloaded, then their tasks or bugs must be reassigned.

4.6 Project Monitoring

"Target Process" supports the project monitoring through the burn down chart, which is drawn automatically considering the initial workload, the variation from baseline (change control) and the projection of the work to do according to the average velocity attained until the day before. The burn down chart shows the sprint progress detailing the iteration progress, the ideal line and the forecasted progress. It only requires that every day the team members update the time remaining for each task in which they had worked during the day.

Fig. 5 shows an example of a sprint burn down chart collected from one of the projects developed by the organization.

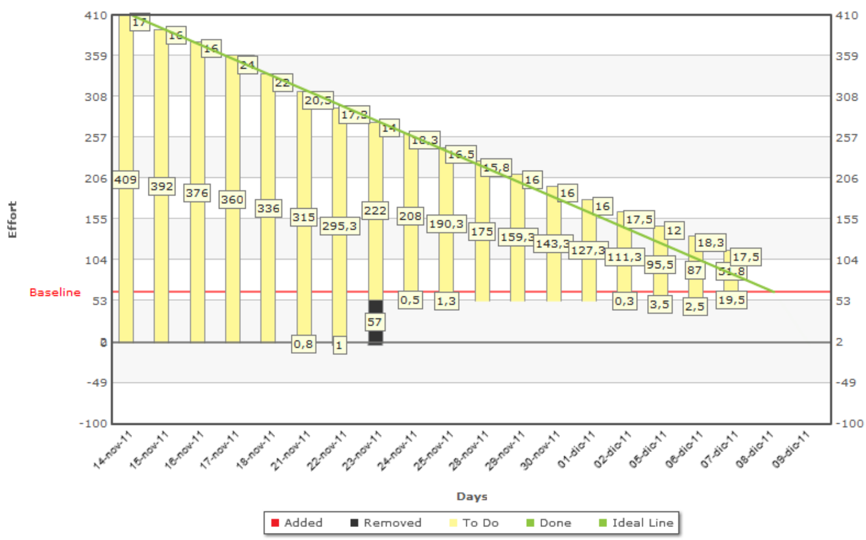


Fig. 5. Sprint burn down chart

4.7 Daily Activities Support

The tool provides two utilities that facilitate teamwork, even if the team members are located in different places.

1. "Team board" allows performing the daily meetings. It shows what is in progress or what was finished yesterday by any team member.
2. "Task board" allows following the bugs and task state. The team members can drag and drop the items in order to update their states.

5 Conclusions

Scrum is a methodology that seems simple but requires discipline and above all ability and experience of the development team. Similarly, as described by its authors, Scrum meets the needs of project management but does not address other important processes such as requirements management.

Therefore, to implement Scrum an organization must make additions based on their own needs and experiences to enhance the process, at least during the first months.

The following are the main conclusions from the study:

- Organizations must be careful not to alter the basic principles of the methodology.
- The definition of requirements achieved with Scrum was not enough to meet deadlines and costs agreed with the clients.
- It is necessary to establish some discipline with some activities because the team was not able to be self-managed or self-organized as Scrum requires.
- An expert is necessary when the team is new in the process.
- To achieve the productivity indicated by Scrum, the team needs a previous experience and excellent technical skills.
- A project management tool with Scrum features must be incorporated.
- The organization must be patient in order to reach a high productivity level and avoid modifying the heart of Scrum. There is a high probability of cancellation in the first sprints.

Some benefits identified with the additions introduced to Scrum are:

- Adjust the sprint tasks for improving the estimation.
- Reduce the ambiguity of the project requirements.
- Allow users to validate the user interface prototypes.
- Minimize the probability of changes.
- Increase the product quality with a specific test phase.

5.1 The Unfinished

The instability of the product due to increments generated by the sprints is the highest priority in the process improvement inside the organization. Different techniques have been identified to improve this aspect but they have not yet been implemented. Adopting an agile methodology requires to be prepared not only technically but also mentally, because it is passed to a model where the change in a user requirement is not a trouble. This is logical when the value provided to the customer is what prevails over any other criteria, and in fact is a competitive factor. But not all organizations are ready to assume the economic consequences of these changes when the impact is not easily measured and cannot be controlled. Continuous changes and increments generated by the sprints require the establishment of a regression test strategy. Applying Scrum specific practices should be strengthened including some "extreme programming" techniques. In this sense, the organization is currently working on "test driven development" (based on unit tests) and "continuous integration" (in order to support regression testing). These techniques will improve code quality and will reduce the probability of destabilizing the product due to frequent deliveries and continuous changes. Once these techniques will be implemented, the next step is the automation of the user interface testing and code generation to continue increasing productivity.

Acknowledgements. This work is 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. Schwab, K.: The Global Competitiveness Report 2010–2011. World economic forum (2010), http://www3.weforum.org
2. Standish group: CHAOS Manifesto (2011), http://standishgroup.com/newsroom/chaos_manifesto_2011.php
3. Glazer, H.: Time to Market vs. Process Discipline. In: SEPG Conference, Nashville, TN, USA (2006)
4. Richardson, I., Wangenheim, C.: Why are Small Software Organizations Different?, Guest editors' introduction. IEEE Software 24(1), 18–22 (2007)
5. Oktaba, H.: Software Process Improvement: The Competisoft Project. IEEE Computer Magazine 40(10), 21–28 (2007)
6. Pino, J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Software Quality Journal 16(2), 237–261 (2007)
7. Tore, D., Torgeir, D.: Empirical studies of agile software development: A systematic review. Information and Software Technology 50(9-10), 833–859 (2008)
8. http://www.bolesfactory.com
9. Schwaber, K.: Agile Project Management with Scrum. Microsoft Press (2003)
10. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (2002)
11. West, D., Grant, T.: Agile Development: Mainstream Adoption Has Changed Agility. Forrester Research (2010)
12. http://www.targetprocess.com
13. Kamalrudin, M., Grundy, J.: Generating essential user interface prototypes to validate requirements. In: 26th IEEE/ACM International Conference on Automated Software Engineering, USA (2011)
14. http://www.balsamiq.com
15. Kniberg, H.: Scrum and XP from the trenches. InfoQ (2007)
16. Humphrey, W.: A Discipline for Software Engineering. Addison-Wesley (1995)

Software Project Management in Very Small Entities with ISO/IEC 29110

Rory V. O'Connor[1,2] and Claude Y. Laporte[3]

[1] Lero, The Irish Software Engineering Research Centre, Ireland
[2] Dublin City University, Dublin, Ireland
[3] École de technologie supérieure, Montréal, Canada
Rory.OConnor@computing.dcu.ie,
Claude.Y.Laporte@etsmtl.ca

Abstract. The recently published ISO/IEC 29110 standard Lifecycle profiles for Very Small Entities has at its core a Management and Engineering Guide [1] which are targeted at very small entities (enterprises, organizations, departments or projects) having up to 25 people [2], to assist them unlock the potential benefits of using standards which are specifically designed to address their needs. This paper discusses the role and structure of Project Management in the emerging ISO/IEC 29110 standard Software Process Lifecycles for Very Small Entities as well as its practical implication. This paper will also focus on the design and development of project management support documentation and their associated usage in early trials of ISO/IEC 29110.

Keywords: VSE, ISO/IEC 29110, ISO, Standards, Project Management.

1 Introduction

Projects are the cornerstone of all business activities in small and very small companies. Firms must complete various projects to achieve their financial goals and obtain information. Business owners and managers have only one attempt executing a project successfully. Hence, the process must be carefully thought out and planned. In their study into why software projects fail [3] have shown that software specialists spend about 40 to 50 percent of their time on avoidable rework rather than on what they call value-added work, which is basically work that's done right the first time.

Administering software development is usually achieved through the introduction of a software project management process. However, implementing software project management controls in very small software companies is a major challenge. This paper introduces the project management practices in the newly published ISO/IEC 29110 [1] standard Software Process Lifecycles for Very Small Entities. The following sections discuss the role of project management in general, the structure of ISO/IEC standard and its project management practices. Finally the paper focuses on the design and development of project management support documentation and their associated usage in early trials of ISO/IEC 29110.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 330–341, 2012.

2 Software Project Management

Many software products fail not because there is no market, but because the cost of creating the software far outstrips any profit. Currently approximately half a million project managers worldwide are responsible for in the region of one million software projects each year, which produce software worth USD$600 billion. It is now accepted that many of these projects fail to fulfill customers' expectations or fail to deliver the software within budget and on schedule [4]. Putnam suggests that about one-third of projects have cost and schedule overruns of more than 125% [4].

Software project failure is often devastating to an organization. Schedule slips, buggy releases and missing features can mean the end of the project or even financial ruin for a company. Some of the major reasons for projects failing are [3]: Unclear objectives; Unrealistic or unarticulated project goals; Inaccurate estimates of needed resources; Poor reporting of the project's status; Unmanaged risks; Poor communication among customers, developers, and users; Poor project management. Many of these clearly relate to project management.

While there are many reasons why software projects fail, one of the most important is incorrect management of the project. Good project management cannot guarantee project success, however bad project management usually results in project failure. Furthermore, software is delivered late, costs more and fails to meet its requirements. Clearly, by using effective project management techniques a project manager can improve the chances of success.

A study by Capers Jones [5] of approximately 250 software projects between 1995 and 2004 shows an interesting pattern. When comparing projects that successfully achieved their cost and schedule estimates against those that ran late, were over budget, or were cancelled without completion, six common problems were observed: poor project planning, poor cost estimating, poor measurements, poor milestone tracking, poor change control, and poor quality control. By contrast, successful software projects tended to be better than average in all six of these areas. Perhaps the most interesting aspect of these six problem areas is that all are associated with project management rather than with technical personnel.

There are many ways to make large software systems fail. There are only a few ways of making them succeed. It is commonly agreed that project management is the key factor that tends to push projects along either the path to success or the path to failure. Among the most important project management practices leading to success are those of planning and estimating before the project starts, absorbing changing requirements during the project, and successfully minimizing bugs or defects.

Successful projects always excel in these critical activities: planning, estimating, change control, and quality control. By contrast, projects that run late or fail typically had flawed or optimistic plans, had estimates that did not anticipate changes or handle change well, and failed to control quality [5].

With the low-cost tools available today for small-scale project management, and the value of project management being increasingly recognized by many in the government and in corporate sectors, many small and very small organizations choose not to take advantage of formal project management techniques and tools. [6]. Given the competitive nature of the current business environment, it may be argued that the need to initiate the right projects and achieve the desired results is just as critical if not more critical for the small business, as it is for the large business. Furthermore it can be argued that project management offers value for any size business, and does not require a large investment of cash capital to establish. In order to minimize risk and set a small business up for success, we contend that such businesses can benefit from some form of formal project management.

3 ISO/IEC 29110 Standard

The ISO/IEC 29110 standard "Lifecycle profiles for Very Small Entities" [1] is aimed at addressing the issues identified above and addresses the specific needs of VSEs [7, 8, 9] and to tackle the issues of poor standards adoption by small companies [10, 11, 12]. The approach [2] used to develop ISO/IEC 29110 started with the pre-existing international standard ISO/IEC 12207 dedicated to software process lifecycles. The overall approach consisted of three steps: (1) Selecting ISO/IEC 12207 process subset applicable to VSEs of up to 25 employees; (2) Tailor the subset to fit VSE needs; and (3) Develop guidelines for VSEs.

The basic requirements of a software development process are that it should fit the needs of the project and aid project success [13]. And this need should be informed by the situational context where in the project must operate and therefore, the most suitable software development process is contingent on the context [14]. The core situational characteristic of the entities targeted by ISO/IEC 29110 is size, however there are other aspects and characteristics of VSEs that may affect profile preparation or selection, such as: Business Models (commercial, contracting, in-house development, etc.); Situational factors (such as criticality, uncertainty environment, etc.); and Risk Levels. Creating one profile for each possible combination of values of the various dimensions introduced above would result in an unmanageable set of profiles. Accordingly VSE's profiles are grouped in such a way as to be applicable to more than one category. Table 1 illustrates a Profile Group which contains three profiles (labeled A, B and C) that are mapped to nine combinations of business models and situational factors.

Table 1. Allocating VSE characteristics to profile groups

	Profile Situational Factors		
Business Models	**Critical**	**User Uncertainty**	**Environment Change**
Contract	*Profile A*	*Profile A*	*Profile A*
In-House	*Profile C*	*Profile B*	*Profile A*
Commercial	*Profile B*	*Profile A*	*Profile A*

Profile Groups are a collection of profiles which are related either by composition of processes (i.e. activities, tasks), or by capability level, or both. The "Generic" profile group has been defined [9] as applicable to a vast majority of VSEs that do not develop critical software and have typical situational factors. This profile group does not imply any specific application domain, however, it is envisaged that in the future new domain-specific sub-profiles may be developed in the future. Table 2 illustrates this profile group as a collection of four profiles, providing a progressive approach to satisfying the requirements of profile group.

Table 2. Graduated profile of the Generic profile group

	Generic Profile Group		
Entry	Basic	Intermediate	Advanced

To date the Basic Profile [1] has been published, the purpose of which is to define a software development and project management guide for performing one project at a time.

3.1 Engineering and Management Guide

At the core of this standard is a Management and Engineering Guide (ISO/IEC 29110-5) [1] focusing on *Project Management* and *Software Implementation* as illustrated in figure 1. The purpose of the *Project Management* process is to establish and carry out in a systematic way the tasks of a software implementation project, which complies with the project's objectives in terms of quality, time and cost. *Project Management* generates a *Project Plan* to direct the software project. During the execution of the project *Change Requests* may cause revisions to the *Project Plan*. The project is the subject of *Project Assessment and Control* during the lifetimes of the project until the *Software Implementation* is complete and *Project Closure* occurs. Software Implementation (SI) produces a specified software system implemented as a software product or service. This process starts with the establishment of *Software Requirements*, after which *Architectural and Detailed Design* are produced. Software is the *Constructed* and verified using *Integration and Test* procedures. The final staged being *product d*elivery to the customer.

Within ISO/IEC 29110, the purpose of the Project Management process is to establish and carry out in a systematic way the Tasks of the software implementation project, which allows complying with the project's Objectives in the expected quality, time and costs. It is intended to be used by the VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the VSE organization or project needs.

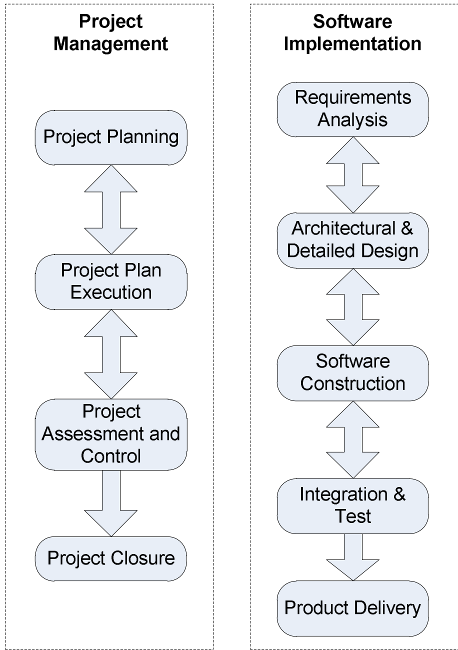


Fig. 1. ISO/IEC 29110 Basic profile Process Diagrams

3.2 ISO/IEC 2910 Project Management Objectives Practices

Figure 2 shows the flow of information between the Project Management Process activities of the Basic profile including the most relevant work products and their relationship.

The objectives of the ISO/IEC 29110-5-1-2 Project Management Process are:

- The *Project Plan* for the execution of the project is developed according to the *Statement of Work* and reviewed and accepted by the *Customer* and the *Tasks and Resources* necessary to complete the work are sized and estimated.
- Progress of the project is monitored against the *Project Plan* and recorded in the *Progress Status Record*. Corrections to remediate problems and deviations from the plan are taken when project targets are not achieved. Closure of the project is performed to get the *Customer* acceptance documented in the *Acceptance Record*.
- The *Change Requests* are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule and technical impact.
- Review meetings with the *Work Team* and the *Customer* are held and agreements are registered and tracked.
- Risks are identified as they develop and during the conduct of the project.

- A software *Version Control Strategy* is developed, where items of *Software Configuration* are identified, defined and baselined, and releases of the items are controlled and made available to the *Customer* and *Work Team*.
- *Software Quality Assurance* is performed to provide assurance that work products and processes comply with the *Project Plan* and *Requirements Specification*.

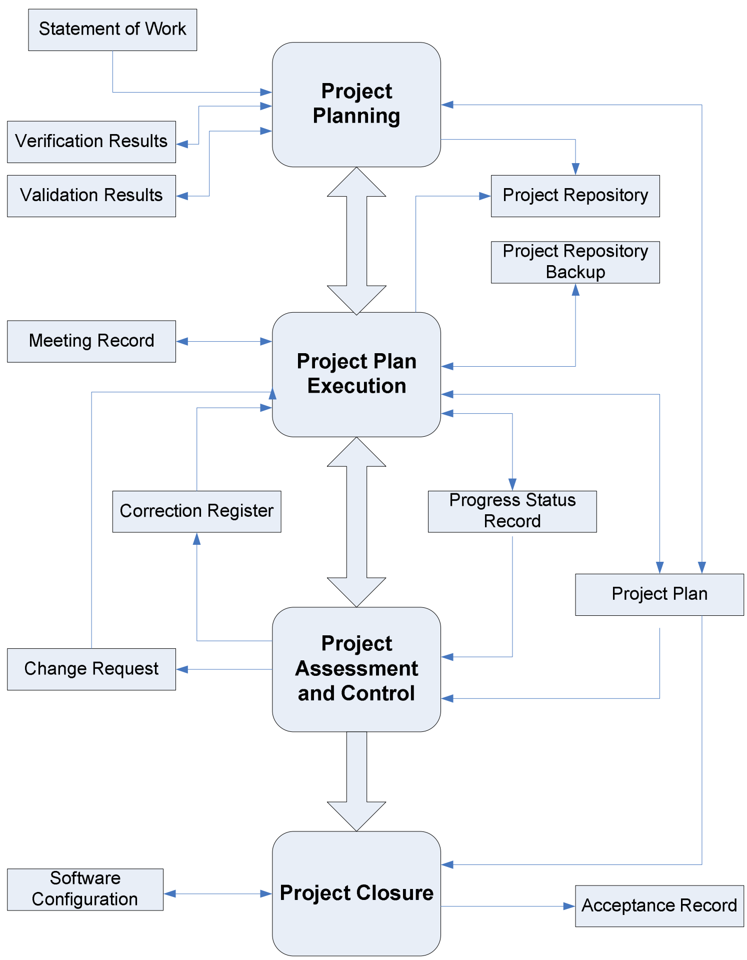


Fig. 2. Overview of ISO/IEC 29110 Project Management Practices

The four activities of the Project Management Process of ISO/IEC 29110-5-1-2 are:

- **Project Planning** - The primary objective of this process is to produce and communicate effective and workable project plans. This process determines the scope of the project management and technical activities, identifies process outputs, project tasks and deliverables, establishes schedules for project task conduct, including achievement criteria, and required resources to accomplish project tasks".

- **Project Plan Execution** - To implement the actual work tasks of the project in accordance with the project plan. Ideally when the project plan has been agreed and communicated to all teams members, work of the development of the product, which is the subject of the project, should commence.
- **Project Assessment and Control** - purpose is to determine the status of the project and ensure that the project performs according to plans and schedules, within projected budgets and it satisfies technical objectives. This process includes redirecting the project activities, as appropriate, to correct identified deviations and variations from other project management or technical processes. Redirection may include re-planning as appropriate.
- **Project Closure** - typically involves releasing the final deliverables to the customer, handing over project documentation to the business, terminating supplier contracts, releasing project resources and communicating project closure to all stakeholders. Often a final step is to undertake a Post Implementation Review (post-mortem) to identify the level of project success and note any lessons learned for future projects.

4 Deployment Assistance

In order to assist with the deployment of ISO/IEC 29110 and to provide guidance on the actual implementation of ISO/IEC 29110-5 in VSEs a series of *Deployment Packages* and *Implementation Guides* have been developed to define guidelines and explain in more detail the processes defined in the ISO/IEC 29110 profiles [15].

A set of *Deployment Packages* (DP) (which are freely available from [16]) are a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. A DP is not a process reference model (i.e. it is not prescriptive). The elements of a typical DP are: description of processes, activities, tasks, roles and products, template, checklist, example, reference and mapping to standards and models, and a list of tools. Packages are designed such that a VSE can implement its content, without having to implement the complete framework at the same time. The table of content of the project management deployment package is illustrated in figure 3.

In addition a series of *Implementation Guides* have been developed to help implement a specific process supported by a tool and are freely available from [16]. To date a small number of implementation guides have been developed. These include:

- Version Control with CVS
- Version Control with SVN
- Project Management with GForge
- Issue tracking with GForge
- Software Process Improvement with OpenOffice Calc.

1. Technical description
Importance of project management
Project management success and failure
2. Definitions (generic and specific definitions)
3. Relationships with ISO/IEC 29110
Project management process
Tasks and roles
4. Detailed description
Roles, products and artifacts
5. Templates
WBS, Project status template, etc.
6. Examples
Project management lifecycle practices, etc.
7. Checklists
Project plan review checklist, etc.
8. Tools
9. Reference to other standards and models
ISO 9000, ISO/IEC 12207 and CMMI for Development
10. References
11. Deployment package evaluation form

Fig. 3. Table of Content of a Project Management deployment package

5 Pilot Projects

The working group (ISO/IEC JTC1/SC7 WG 24) behind the development of this standard is advocating the use of pilot projects as a mean to accelerate the adoption and utilization of ISO/IEC 29110 by VSEs around the world. Pilot projects are an important mean of reducing risks and learning more about the organizational and technical issues associated with the deployment of new software engineering practices. A successful pilot project is also an effective means of building adoption of new practices by members of a VSE. Pilot projects are based on the ISO/IEC 29110-5 Management and engineering guide [1] and the deployment package(s).

To date a series of pilot projects have been completed in several countries utilizing some of the deployment packages developed. For example in Canada a pilot study has been conducted with an IT department with a staff of 4: 1 analyst and 3 developers, who were involved in the translation and implemented 3 DPs: Software Requirements, Version Control, Project Management. In Belgium a VSE of 25 people started with a process assessment phase aiming to identify strengths and weaknesses in development related processes. This company is now working on improvement actions mainly based on the following Deployment Packages: Requirement Analysis, Version Control, and Project Management. In France, a pilot study [17] was conducted with a 14-people VSE that builds and sells counting systems about the frequenting of natural spaces and public sites. In addition a further series of pilot projects are currently underway in Canada, Ireland, Belgium and France, with further pilot projects planned in the near future.

5.1 Trials to Date

To date we have published [17] the final conclusions and results of one pilot project that conducted with a 14-person VSE based in France, which successfully implemented ISO/IEC 29110 processes practices utilising the available Deployment Packages. From which we have identified some potential additional infrastructure and support process activities and suggestions for future evolution of ISO/IEC 29110 Process Profiles. A further series of pilot projects are currently underway in research laboratories and enterprises in Canada, Ireland, Belgium and France, with further pilot projects planned in the near future.

	Small Project	**Medium project**	**Large project**
Duration of project	Less than 2 months	Between 2 and 8 months	More than 8 months
Size of team	Equal or less than 4 people	Between 4 and 8 people	More than 8 people
No. of engineering specialties involved	One specialty	More than one specialty	Many specialties
Engineering fees	Between 5,000$ and 70,000$	Between 50,000$ and 350,000$	Over 350,000$

Fig. 4. Three categories of engineering projects

At the ETS (École de technologie supérieure, www-eng.etsmtl.ca), a 6,000 student engineering school of Montréal, graduate and undergraduate students have used the project management process of the Basic profile to start-up VSEs and have implemented the activities in existing VSEs. As an example, a graduate student currently employed as a professional engineer in an engineering firm having over 400 engineers, is developing a project management process, using the basic profile, for the very small projects of his enterprise. In his enterprise, projects are divided in three categories as illustrated in figure 4. A formal project management process was already used for medium-size and large-size projects, but only an informal process was used for the small-size projects. The task of the graduate project is to define a project management process for the small-size projects using the basic profile, obtain approval by the vice-president of the engineering firm and help deploy the new process.

In another pilot project, a website is developed by a VSE of 2 people to help travelers throughout the life cycle of a trip from its initial planning to sharing the experience of the traveler with friends. The site will be able to build a custom profile for each user in order to propose relevant items such as travel activities or accommodations. The set of proposed roles of the basic profile has been allocated, as illustrated in figure 5, to the two-people VSE. Where one member of the VSE, team member B, plays the role of the project manager.

Role	Name of team member
Analyst	A
Designer	B
Programmer	A/B
Project Manager	B
Technical Leader	A
Work Team	A/B

Fig. 5. Allocation of roles in a two-people VSE

The project manager uses the project management process of the basic profile to manage the project and produce or review the documents listed in figure 6.

Name of document	Main author	Reviewer (if applicable)
Change Request	A	B
Correction Register	B	A
Maintenance Documentation	B	A
Meeting Record	A	
Product Operation Guide	B	B
Progress Status Record	B	
Project Plan	B	A
Project Repository	B	
Project Repository Backup	B	
Requirements Specification	A	B
Software	A/B	
Software Components	A/B	
Software Configuration	A/B	
Software Design	B	A
Software User Documentation	A	B
Statement of Work	A	B
Test Cases and Test Procedures	A	B
Test Report	A	
Traceability Record	B	A
Verification Results	A/B	
Validation Results	A/B	

Fig. 6. Allocation of documents in a two-people VSE

As the VSE grows, the set of roles will be attributed amongst all people of the VSE using the same project management and software implementation processes of the basic profile. If this VSE eventually decides to work on more than one project at a time, it will then use the project management process of the intermediate profile of ISO/IEC 29110.

6 Discussion

As ISO/IEC 29110 is an emerging standard there is much work yet to be completed. The main remaining work item is to finalize the development of the remaining three profiles: (a) Entry – a six person-months effort project or a start-up VSEs; (b) Intermediate - Management of more than one project and (c) Advanced - business management and portfolio management practices. In addition the development of additional Profile Groups for other domains such as critical software, game industry, scientific software development are being studied

Recently, working group 24 was mandated to develop a standard for VSEs developing systems. A system may include material, computer programs, firmware and technical documentation. The new standard for VSEs will use ISO/IEC 15288 System life cycle processes standard [18] as the main framework. The objective of the working group is to develop a systems engineering basic profile which will match the software engineering basic profile. The working group will use the actual project management process of the software basic profile as the baseline to modify or add new tasks required by systems engineers. As an example, since most systems have material components, the project manager of a VSE must decide if the material components will be developed and built internally or subcontracted. This 'make or buy' task was not a task of the software project management process, it will therefore be added to the systems basic profile [19].

Acknowledgments. This work is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

References

1. International Organization for Standardization (ISO): ISO/IEC TR 29110-5-1-2 Software engineering - Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile, Geneva (2011), available at no cost from `http://standards.iso.org/ittf/PubliclyAvailableStandards/c051153_ISO_IEC_TR_29110-5-1_2011.zip`
2. Laporte, C.Y., Alexandre, S., O'Connor, R.V.: A Software Engineering Lifecycle Standard for Very Small Enterprises. In: O'Connor, R., et al. (eds.) EuroSPI 2008. CCIS, vol. 16, pp. 129–141. Springer, Heidelberg (2008)
3. Charette, R.N.: Why Software Fails, pp. 42–49. IEEE Computer Society, Spectrum (September 2005)

4. Putnam, L.H., Myers, W.: Industrial Strength Software: Effective Management Using Measurement. IEEE (1997)
5. Jones, C.: Software Project Management Practices: Failure Versus Success, CrossTalk (October 2004)
6. Hamilton, G., Byatt, G., Hodgkinson, J.: Project management for the small business 6 misconceptions about project management are addressed. CIO (April 2011)
7. O'Connor, R.V., Laporte, C.Y.: Deploying Lifecycle Profiles for Very Small Entities: An Early Stage Industry View. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 227–230. Springer, Heidelberg (2011)
8. O'Connor, R.V., Laporte, C.Y.: Using ISO/IEC 29110 to Harness Process Improvement in Very Small Entities. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 225–235. Springer, Heidelberg (2011)
9. O'Connor, R.V., Laporte, C.Y.: Towards the provision of assistance for very small entities in deploying software lifecycle standards. In: Proceedings of the 11th International Conference on Product Focused Software (PROFES 2010). ACM (2010)
10. Coleman, G., O'Connor, R.V.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software 81(5), 772–784 (2008)
11. O'Connor, R.V., Coleman, G.: Ignoring 'Best Practice': Why Irish Software SMEs are rejecting CMMI and ISO 9000. Australasian Journal of Information Systems 16(1) (2009)
12. O'Connor, R.V.: Evaluating Management Sentiment towards ISO/IEC 29110 in Very Small Software Development Companies. In: Mas, A., et al. (eds.) SPICE 2012. CCIS, vol. 290, pp. 277–281. Springer, Heidelberg (2012)
13. Clarke, P., O'Connor, R.V.: The Meaning of Success for Software SMEs: An Holistic Scorecard Based Approach. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 72–83. Springer, Heidelberg (2011)
14. 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)
15. Laporte, C.Y.: Contributions to Software Engineering and the Development and Deployment of International Software Engineering Standards for Very Small Entities. PhD thesis of the Université de Bretagne Occidentale, Brest (2009), `http://tel.archives-ouvertes.fr/tel-00483255/fr/`
16. ISO/IEC JCT1/SC7 Working Group 24 Deployment Packages repository, `http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html`
17. Ribaud, V., Saliou, P., O'Connor, R.V., Laporte, C.Y.: Software Engineering Support Activities for Very Small Entities. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 165–176. Springer, Heidelberg (2010)
18. ISO/IEC 15288:2008, Systems and software engineering - System life cycle processes, International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland (2008)
19. Laporte, C.Y., Fanmuy, G., Ptack, K.: The Development of Systems Engineering International Standards and Support Tools for Very Small Enterprises. In: 22nd Annual International Symposium of the International Council on Systems Engineering, Rome, July 9-12 (2012)

Author Index