Software Process Improvement Framework

Natalja Nikitina

Doctoral Dissertation in
Information and Communication Technology
Stockholm, Sweden, 2014
Abstract

Many software development organizations today are keen on improving their software development processes in order to develop software products faster, cheaper or better. For that reason, Software Process Improvement (SPI) has received significant attention from the research community over the last few decades. Process maturity models have become widely known for benchmarking software processes against predefined practices and for identifying processes to be improved or implemented, whereas process improvement approaches were developed for guiding the actual process improvement process. However, despite a wide number of provided guidelines on how to standardize the processes and how to run process improvement efforts, only a few SPI initiatives have succeeded. About 70% of the SPI initiatives fail and a significant number do not even get started. Many studies argue that the success of the SPI initiatives is dependent on the organizational, social and managerial aspects of process improvement. Those aspects however are not sufficiently covered by the existing SPI approaches and models. The little knowledge on organizational, social and managerial aspects of SPI that is available is mostly scattered across the domain. Hence, there is lack of a holistic overview of the current SPI domain that provides sufficient coverage of organizational, social and managerial aspects of SPI.

This thesis has explored the organizational, social and managerial aspects of SPI and placed them into the context of the SPI domain. Its main research result is **Software Process Improvement Framework (SPIF)**. The framework provides an overview of the SPI domain and positions theories representing organizational, social and managerial aspects of SPI in the context of existing SPI approaches, models, methods and practices. SPIF is based on the existing theoretical framework for SPI environment proposed by Sami Zahran. The SPIF framework has been additionally complimented with four additional outcomes of this study. Those are: 1) a list of organizational, social and managerial factors facilitating SPI effort, 2) a list of contextual factors impacting process change, 3) a process model for guiding software method adoption, and 4) a checklist representing the properties of successful and sustainable SPI projects.

The research was based on a strong industrial cooperation. As many as thirty software development organizations were involved in this research. Methodologically, the research was conducted in line with the inductive reasoning, which guided the research into building the knowledge from empirical studies. However, at some stages of this research, literature studies were incorporated. The main research methods of this study are action research and case studies, whereas data collection methods are primarily structured interviews, participatory observations and surveys.

The thesis concludes that implementing a recommended software development processes or practices using well defined SPI approaches is not enough. In order to implement successful and lasting process improvement, organizations also need to consider organizational, social and managerial aspects of SPI. The SPIF framework and other results of this thesis may significantly benefit software development organizations that plan to conduct software process change, or have already done it. These organizations may use SPIF for getting an overview of the process improvement process and the theories, methods and tools that should support it. The other results of this thesis can be used for: 1) incorporating organizational, social and managerial aspects in process changes, 2) for adapting process improvements in various organizational contexts, 3) for guiding adoptions of new software development methods, and finally 4) for evaluating and improving process improvement efforts.

**Keywords:** software process, process improvement, SPI, software method adoption, organizational change, change management, organizational aspects, social aspects, managerial aspects, SPI success factors, SPI checklist.
To my son, Sebastian, who was in my tummy while I wrote this thesis
Acknowledgement

This thesis, like any other achievement in my life, would have never been accomplished without help and support of many people. During this five year long journey, I have met, interviewed and worked together with many people that have contributed in many ways to my intellectual, professional and personal growth. For this I am thankful to you all.

First of all I would like to thank my main supervisor, Associate Professor Mira Kajko-Mattsson for making the time of my studies a continuous learning experience. To say that her commitment and dedication to the profession and to her work is enormous would be an understatement. Mira, I will always be thankful for your dedication to my work and for all the time you invested in guiding me to become a researcher. I have learned a lot from you.

I also would like to express my gratitude to all the people who made it possible for me to study the state of the art in the software industry. Without the industrial knowledge that I gathered from many many people, the contribution that this research makes to the field of Software Process Improvement would not be significant. To begin with, I would like to thank Andy James Nolan, for sharing his knowledge and for allowing me to learn from his considerable experience. His support to my study, occasional encouragements, and incredible patience and interest to my work allowed me to develop this research much further. The numerous interviews and chatty discussions that grew from them made us friends, which I appreciate even more. I would also like to thank Magnus Stråle and Pelle Niklasson for their time and support. Our fruitful cooperation allowed me to study the industrial case of software process improvement effort in real time and significantly learn from it; this experience benefited my research to great extent. Furthermore, I would also like to thank other IT professionals whose contribution and involvement in this research has been significant. My special thanks to Aðalsteinn Rúnar Óttarsson, Marcus Lagergren, Selim Erten and Torbjörn Karlsson for sharing their experience and evaluating the results of this study.

During the course of conducting this research I interviewed many software engineers. I would like to thank all of them for dedicating their time and for sharing their knowledge and experience with me. Among them are: Fredrik Tjärnberg, Paul Smith, Linus Ekström, Johan Björnfot, Per Gunsarfs, Tuva Palm and Mathias Axelsson. I would also like to thank my friends for participating in the interviews. Many thanks to: Emmanouil Rovithis, Mikael Magnusson, Ana Gabriela Márquez, Dinusha Gunaratne, Dennis Wickramasurendra, Pamela Vicelja, Buster Olsborn, Diana Razera and Qing Guan. I would also like to thank my friends for participating in the interviews. Many thanks to: Emmanouil Rovithis, Mikael Magnusson, Ana Gabriela Márquez, Dinusha Gunaratne, Dennis Wickramasurendra, Pamela Vicelja, Buster Olsborn, Diana Razera and Qing Guan. To of all you and many other people, who took part in the interviews, I am immensely grateful. Your experiences and valuable comments gave me crucial information for the development of my work.

I would like to thank the internal reviewer of this thesis, Associate Professor Anne Håkansson, for her feedback on the early version of this thesis. Her comments helped me to improve the thesis and present my work in the better way.

I am also grateful to all the colleagues from the ICT school of Royal Institute of Technology who have been supportive in one way or another. My research group fellows, Salman Ahmad Khan and Gudrun Jeppesen, for active discussions on the topic and valuable feedback. To Katharina Rasch, my office mate, for the occasional chats on topics beyond work and for her patience during the countless phone interviews that I made from our office. To my colleagues and friends in the department, Esmiralda Moradian, Dan Wu, Kathrin Dannmann, Håkan Olsson, Vladimir Vlassov, Nima Dokoozadeh, Shahab Mokarizadeh, for encouraging...
discussions in the corridors, at “fikas” and lunches. To the head of the Software and Computer Systems unit, Thomas Sjöland, for his strong support and for always being able to solve any administrative issue in a professional and kind manner. To the members of the administrative staff, Marianne Hellmin and Sandra Gustavsson, for being always available and for answering all of my questions with a smile.

Above all I would like to thank my family, who has always supported me. My mother, Valentina, for her unconditional love and care throughout my life. My father, Sergej, for encouraging my interest in science from early days. To my sister, Tatjana, for being my first formal teacher who taught me how to read and write. To my brother, Pavel, for being a role model and the reason why I became a software engineer. Finally, I would like to thank my husband, Camilo, who has always been my biggest supporter. Thank you, mi amor, for all your love, for all your care, for always being by my side, and for always believing in me. “Te amo todo”. 
# Table of Contents

Part I PhD Thesis ................................................................. 3

1 Introduction .................................................................. 5
   1.1 Research problem .................................................. 6
   1.2 Research goal and objectives .................................. 6
   1.3 Contribution .......................................................... 7
   1.4 Software organizations involved ......................... 7
   1.5 Target audience ..................................................... 7
   1.6 Research scope and limitations ............................. 8
   1.7 Structure of the thesis ........................................... 8
   1.8 List of publications ................................................ 9

2 Background and related work ....................................... 11
   2.1 Software development processes, process models and methods ........ 11
   2.2 Software process improvement ............................. 17
   2.3 Framework for SPI environment ........................... 25
   2.4 Organizational change management .................. 26

3 Research approach ...................................................... 33
   3.1 Research paradigm .............................................. 33
   3.2 Research strategy ................................................ 34
   3.3 Research process ............................................... 35
   3.4 Research method ............................................... 43
   3.5 Data collection and analysis .................................. 44
   3.6 Organizations involved ....................................... 47
   3.7 Validity .............................................................. 50

4 Research contribution ................................................... 53
   4.1 Summary of research papers ................................ 53
   4.2 Other contribution ................................................ 62

5 Research results .......................................................... 63
   5.1 Terminology ........................................................ 63
   5.2 Software process improvement framework .............. 64
   5.3 Organizational, social and managerial factors facilitating SPI effort .... 72
   5.4 Contextual factors impacting process change ........... 77
   5.5 SMA process model ............................................. 79
   5.6 SPI checklist ....................................................... 82
6 Conclusions.............................................................................................................. 85
   6.1 Future work................................................................................................. 88

References............................................................................................................... 91

Appendix A............................................................................................................. 99
   Appendix A.1.................................................................................................... 99
   Appendix A.2.................................................................................................. 101
   Appendix A.3.................................................................................................. 106
   Appendix A.4.................................................................................................. 109
   Appendix A.5.................................................................................................. 110
   Appendix A.6.................................................................................................. 111
   Appendix A.7.................................................................................................. 112
   Appendix A.8.................................................................................................. 113
   Appendix A.9.................................................................................................. 118

Appendix B............................................................................................................. 119

Appendix C.......................................................................................................... 123

Appendix D.......................................................................................................... 125

Part II Publications............................................................................................. 127
Part I
PhD Thesis
Chapter 1

Introduction

Today, software development organizations are keen on developing and maintaining their software products in an efficient and effective manner. Therefore, they strive towards having well-organized and suitable software development processes. While “in theory”, well-organized software processes can be achieved by simply implementing one of the widely known and accepted software development methods, it may not be so “in practice”. Due to different organizational contexts, certain software development methods can be suitable for some organizations but not for others. Moreover, software organizations often lack expertise on and knowledge of how to identify the most suitable software development methods, and most importantly, how to adapt them to the organizational context in a correct manner.

It is widely known that there is no “silver bullet” software development method that is suitable for all (Sommerville, 2011). Software organizations have their own cultures, characters, identities, formality levels and needs. Therefore, they cannot just utilize any predefined existing software development method. They have to adapt it first to their individual needs and contexts.

Adapting a software method is, however, not a one-off activity. Since in today’s highly changing and competitive business environments, the organizations, their cultures, identities and needs continuously change and evolve and so do their software development processes. Therefore, software development processes need to be continuously refined and improved, which can be achieved by implementing Software Process Improvement (SPI).

SPI aims at the continuous refinement of software process by means of understanding the existing software processes, identifying problems or weaknesses in them and by continuously changing them (Sommerville, 2011). Its overall goal is to increase process quality, and thereby, achieve at least one of the following: 1) improved product quality, 2) reduced development time, or 3) reduced development cost (Paulish & Carleton, 1994), (Sommerville, 2011).

Software process improvement can be performed in three ways, by for instance: 1) adopting a new software development method, 2) identifying and eliminating the weaknesses and defects of the existing software processes, or 3) by benchmarking the process against and changing it towards existing reference models. The adoption of a new software development method is often done in a company specific way since there are no standard procedural guidelines for it. The two other ways to improve software process, however, are supported by instrumental, procedural, and occasionally, by organizational SPI guidelines. Identifying and eliminating the weaknesses of the existing software processes can be supported by inductive SPI approaches such as Quality Improvement Paradigm (QIP) (Basili, 1985), and Six Sigma (Bhote, 1989). Benchmarking the software process against existing reference models is driven by process standardization and based on the implementation of prescribed processes and best practices advocated by a certain reference model. The best known reference models in software engineering industry are process capability and maturity models, such as CMMI (Capability Maturity Model Integration) (CMMI Product Team, 2002) and SPICE (Software

A large number of approaches, standards and models is associated with SPI. However, despite the provided guidelines on how to standardize the processes and how to run SPI efforts, only a few SPI initiatives succeed. The research shows that about 70% of the SPI initiatives fail and a significant number do not even get started (Ferreira & Wazlawick, 2011). The studies on the SPI success factors argue that success of the SPI initiatives is dependent on the organizational, social and managerial aspects, such as senior management commitment, staff involvement, training and mentoring, just to mention a few (Niazi, et al., 2006b). The need for incorporation of the organizational culture and tightly integrated in it social issues to the SPI initiatives has been recognized by a number of researchers (Kautz, et al., 2001), (Kautz & Nielsen, 2004), (Sahota, 2012), (Muller, et al., 2008), (Heikkila, 2009), (Muller, et al., 2010), (Ferreira & Wazlawick, 2011), (Mathiassen, et al., 2005), (Kandt, 2003). However, those aspects are not sufficiently covered by the existing SPI approaches and models.

1.1 Research problem

Even though research on SPI is not new and a large number of studies have been dedicated to process improvement, there is lack of a holistic overview of the current SPI domain that provides sufficient coverage of organizational, social and managerial aspects of SPI. Most of the existing SPI approaches and models focus only on technical, instrumental and procedural aspects of SPI (McFeeley, 1996), (ISO/IEC, 2004), (Basili, 1985). A few SPI models, such as P-CMM (Curtis, et al., 2001) and Six Sigma (Bhote, 1989) address some of the organizational and social aspects, but do not cover them sufficiently. The little knowledge on organizational, social and managerial aspects of SPI that is covered by a few SPI approaches is mostly scattered across the domain. To the knowledge of the author of this thesis, there is just one model that provides a general overview of the SPI field and, to some extent, considers organizational, social and managerial aspects of SPI (Zahran, 1998). The model however is outdated.

1.2 Research goal and objectives

This thesis aims to explore the organizational, social and managerial aspects of SPI and place them in the context of the SPI domain. Therefore, the main research objective of this thesis is to provide an overview of the SPI domain and position theories representing organizational, social and managerial aspects of SPI in the context of existing SPI approaches, models, methods and practices. The research is complimented by four secondary research objectives. Those are the following.

- RO-1) to elicit SPI success factors that relate to organizational, social and managerial aspects of SPI;
- RO-2) to elicit contextual factors that impact the design of a software process change;
- RO-3) to explore how to conduct software method adoption as part of software process improvement and organizational change processes;
- RO-4) to explore how to assess the status of the SPI effort and provide recommendations for its improvements, considering organizational, social and managerial aspects of SPI.
1.3 Contribution

Aligned with the main research objective, the main contribution of this research is Software Process Improvement Framework (SPIF). The framework aims to give a general overview of the complexity and diversity of SPI, present the existing SPI approaches and models, incorporate theories of the organizational change, and position other research results of this thesis in relation to the existing theories in SPI. In line with the four secondary research objectives of this study, the other research results includes the following outcomes: 1) a list of organizational, social and managerial factors facilitating SPI effort (originated from SPI success and sustainability factors), 2) a list of contextual factors impacting process change, 3) a process model for Software Method Adoption (SMA model), and 4) SPI checklist.

First, we listed SPI success and sustainability factors leading to the successful and sustainable SPI efforts. From those factors, we selected the organizational, social and managerial factors facilitating SPI effort by choosing the factors that relate to organizational, social and managerial aspects of SPI. Second, we listed contextual factors that impact the design of process change. Those factors may help organizations to choose a suitable change strategy, or tailor a change process to the organizational needs. Third, we developed the SMA model, a process model for conducting a software method adoption as part of software process improvements. The process model is to be used for guiding adoption of a new software method, of a set of new practices, or an adoption of significant process changes or transformations. Last, we created an SPI checklist that represents the properties of successful and sustainable SPI projects. It can be used for guiding the SPI effort towards successful and sustainable results, as well as for providing the recommendations for the improvements of the SPI effort.

1.4 Software organizations involved

Due to a strong relevance of the research problem to the software industry, we based this research on a strong industrial cooperation. As many as thirty software development organizations were involved in this research. Their level of involvement, however, differed depending on the stage in which they were involved and the purpose of the study. Three software organizations intensively supported our research and allowed us to study their software development processes in detail. Out of them, one organization allowed us to participate in their SPI initiatives. In addition, industrial representatives from twelve software organizations helped us to identify SPI success factors, and representatives from fifteen other organizations evaluated the significance of the initial SPI checklist items. Finally, one organization was involved in the evaluation of the SPI checklist and three organizations were involved in the evaluation of the SMA model. Representatives of those organizations helped us to map the content of the SPI checklist and the SMA model to the industrial SPI and method adoption projects. Detailed descriptions of the software organizations involved in this research are presented in Section 3.6.

1.5 Target audience

There are three target groups that can benefit from the results of this thesis. The main target group is software development and engineering organizations that plan to conduct software
process change, or have already done it. These organizations may use the SPIF framework for structuring their process improvement efforts, for choosing suitable change strategies, for guiding the adoption of new processes or methods, for evaluating and improving their process improvement efforts, and finally, for incorporating organizational, social and managerial aspects into the process change. Additionally, those organizations may significantly benefit from the results of this research by increasing their understanding of the complexity of the software process change, and by acknowledging the importance of organizational, social and managerial aspects of it.

The second target group of this thesis is the academic community. Researchers studying the area of software process improvements can benefit by reviewing the contributions of this thesis and by incorporating its outcomes to the existing SPI methods, models and approaches. The results of this thesis have the largest potential to contribute to the research aiming at building a comprehensive overview of software process improvement.

Finally, the remaining target group consists of organizations that conduct assessment, and standardization of the organizational software processes, as well as guide their continuous improvements. Those organizations can be especially interested in the SPI checklist, which can be used for evaluating the status of SPI and improving it.

1.6 Research scope and limitations

Software process improvement is a highly complex and diverse process. It embraces a myriad of various technical, instrumental, procedural organizational, social and managerial aspects to be considered, such as clear directions, full commitment, sponsorship, resources, and knowledge just to mention a few. The structured and holistic overview of the process improvement process covers several scientific fields; among them are software engineering, organizational science, and psychology. Therefore, presenting a holistic view of the process improvement effort could not have been elucidated within the scope of a doctoral thesis. Hence, this study and the SPIF framework that resulted from it do not provide a comprehensive, nor holistic view of SPI. The results of this thesis, however, contribute to the broader understanding of the software process change. Thus, the focus of the research is on the understanding of the complexity of SPI effort, the recognition of its diversity, and the incorporation of the organizational, social and managerial aspects of SPI.

1.7 Structure of the thesis

This thesis consists of two major parts. Part I presents the background of this research, the research approach and its results. Part II, on the other hand, presents the ten research papers, which resulted from this study.

Part I includes the presentation of the research problem and research goal, description of the related work, and detailed report on the research method used and the results achieved. It consists of six chapters and four appendices, as follows.

- Chapter 1 – Introduction: provides a brief introduction into the research domain, presents the research problem, research goal, and overview of the research contributions.
Chapter 2 – Background and related work: provides a short overview of the concepts and related work in the field of software development processes and process improvements, and presents theories from the organizational change management that have been incorporated in or influential to this research.

Chapter 3 – Research approach: describes the research paradigm and research strategy that guided this research, and presents the research process that was followed to conduct this study.

Chapter 4 – Research contribution: gives an overview of the research produced, and presents summaries of ten research papers written during the course of writing this thesis.

Chapter 5 – Research results: presents the Software process improvement framework and the other outcomes of this study.

Chapter 6 – Conclusions: summarizes this thesis and presents conclusions of this research.

Appendix A: presents data collection instruments, such as questionnaires and surveys, used during the research process.

Appendix B: presents SPI success and sustainability factors identified during this research and lists their literature and empirical sources.

Appendix C: presents the results of the evaluation of the SMA model within six industrial method adoption projects.

Appendix D: provides the overview of the SPI checklist and lists the importance of its items based on the results of evaluation of ten finalized SPI projects.

Part II comprises the ten research papers representing the main research contribution of this work. The publications are listed in Section 1.8, and their summaries are provided in Section 4.1.

1.8 List of publications

The research presented in this thesis has been reported on in ten research papers that have been published in or submitted to peer-reviewed scientific conferences and journals. All the published papers illustrate the research process and research work conducted during the course of writing this thesis. Paper 6, Paper 8 and Paper 10, however, present the main results of this thesis, as presented in Chapter 5.

In the moment of writing this thesis, two out of ten papers were still under review process. Below we list the papers and specify the fora they were published in or submitted to.

Paper 1. Impact of Growing Business on Software Processes

Paper 2. Impact of Corporate and Organic Growth on Software Development

Paper 3. Historical Perspective of Two Process Transitions
Paper 4. Developer-driven Big-bang Process Transition from Scrum to Kanban

Paper 5. From Scrum to Scrumban: A Case Study of a Process Transition

Paper 6. Factors Leading to the Success and Sustainability of Software Process Improvement Efforts

Paper 7. Software Process Improvement Health Checklist

Paper 8. Successful Process Improvement Projects are no Accidents


Natalja Nikitina, Mira Kajko-Mattsson. Submitted to ICSSP 2014

The author of this thesis, Natalja Nikitina, is the main author of all the ten included papers. This implies the responsibility for designing and conducting the research reported on in the papers, as well as conducting most of the writing. The second author of all the included papers, Mira Kajko-Mattsson, supervised the research work and contributed to the structure and content of the papers. Paper 5 and Paper 8 were written in cooperation with industrial representatives, Magnus Stråle and Andrew James Nolan, respectively. The industrial representatives provided data for the research and confirmed the correctness of the understanding of the subject studied.
Chapter 2

Background and related work

Software development concerns all the aspects of software production from early stages of software requirement specification to maintaining the software after it has gone to use (Sommerville, 2011). In addition, software development is to a certain degree a social phenomenon constrained by organizational culture and structure, and the way of working and collaborating of stakeholders (Maciaszek & Liong, 2006). The processes of software development are formed by the technologies and tools available for the software developers, but they are to a large extent impacted by the way the work is organized, and the way stakeholders interact and collaborate (University of Bergen, u.d.). The processes of software development are both technological and social processes that are run by requirement engineers, developers, testers and other technical staff. Therefore, the change of the software development processes impact a myriad of both technological and social aspects which are tightly integrated in the organizational culture.

To aid in understanding the social and technological aspects of software process and SPI, as well as organizational change, this chapter provides a short overview of the used concepts and related work in the field of software development processes, software process improvements, organizational change and change management. First, Section 2.1 presents the concepts of software development processes, process models and methods. It describes the software development processes and presents best known software process models and methods. Second, Section 2.2 presents the state of the art of SPI field and describes best known SPI approaches and models. Third, Section 2.3 describes a framework for SPI environment, which was used as a base to outline the SPIF framework, that resulted from this research and that is presented in Section 5.2 of this thesis. Finally, the theories of organizational change management that can be incorporated to SPI are covered by Section 2.4.

2.1 Software development processes, process models and methods

This section gives an overview of the software development processes, process models and methods. Section 2.1.1 clarifies the terms of software processes, software process models, and software development methods and contributes to the understanding of those concepts. Then, Section 2.1.2 presents software development processes and the activities included in them. Finally, Section 2.1.3 describes the best known software process models and software development methods.

2.1.1 Process, process model and method

The terms process, process model, and method are often seen as synonyms and the distinction between them is not clear for many. It is important, however, to clarify the difference in their meanings. The general definitions of process, process model, and method are the following.
Since a process is defined as a series of actions leading to an end, software development process can be defined as a series of actions leading to a software product. In this thesis, we refer to a software development process, or software process as a series of actions, procedures or activities performed by the organization in order to design and produce a software product. The formal and more detailed definition of software development process is the following:

- **Software development process:** “a process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use.” (IEEE Computer Society, 1991)

The most common way to describe a software process is to organize it as a model containing key process features (Pfleeger & Atlee, 2006). This way of representing a process is called, a software process model. In this thesis, we refer to a software process model as a description of a software process organized in a model. The definition of the software process model is the following:

- **Software process model:** “a simplified description of a software process which is presented from a particular perspective.” (Sommerville, 2011)

There is a very thin line of difference between software method and software process model or software process. By its definition method implies incorporation of procedures and techniques in the systematic, organized and structured manner. Unfortunately, the software development method lacks its own official definition. In this thesis, we refer to a software development method or a software method as following:

- **Software development method:** a collection of prescriptive activities, procedures and techniques that should be performed in a systematic way, resulting in or contributing to a production of software product.

### 2.1.2 Software development processes

Software development process, commonly referred also as software process, is a set of activities and guidelines that lead to the production and maintenance of a software product. It includes the basic software engineering activities related to requirements engineering, design, implementation, testing, and maintenance, as well as any other activities that result in software products such as software prototyping, software modification, reuse, and system re-engineering. There are many different software processes but most of them build on the fundamentals of software engineering, the software lifecycle. The software lifecycle has general
Figure 2.1  **Software lifecycle** (Pfleeger & Atlee, 2006), (Maciaszek & Liong, 2006)

representation, consisting of the following main phases 1) requirements analysis, 2) system
design, 3) implementation, 4) integration and testing, 5) deployment, and finally 6) operation
and maintenance (Pfleeger & Atlee, 2006), (Maciaszek & Liong, 2006), as shown in Figure 2.1.
Despite the fact that the general view of the **software lifecycle** is widely accepted and practiced
by the academic and industrial audience, there is no agreement on a detailed software process
that would fit all. The reason for it is great variations among the software products, and
organizational cultures of the software companies.

Software process contains large amount of activities concerning both engineering and
management disciplines. The main engineering activities concern **requirements engineering,**
**software analysis and design,** **implementation and testing** (Robillard, et al., 2003). The main
management activities of software process concerns **configuration and change management,** and
**project management** (Robillard, et al., 2003). The software processes rely on people following
them, making decisions and judgments. Since there is no “ideal” software processes, each
software organization develops its own software process, defining and structuring their
software engineering and management activities in a unique way.

The quality, efficiency and composition of software process activities affect the cost of
software development, ultimate product quality, and time to market of software product. For
this reason, many authors stress the importance of software processes and its quality (Pfleeger
& Atlee, 2006), (Sommerville, 2011). Software process, however, is not a tangible object; hence
it is hard to evaluate its quality. Sommerville suggests software process characteristics that
represent different aspects of quality of the software process, which combined together
correspond to the level of maturity of the process (2011). Some of the process characteristics
can be conflicting. Therefore, the importance of process characteristics needs to be tailored to
the context of specific organization and its needs. The characteristics of software process are
presented in the Table 2.1.

### Table 2.1  **Software process characteristics**, (Sommerville, 2011)

<table>
<thead>
<tr>
<th>Process characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>To what extent is the process explicitly defined and how easy is it to understand the process definition?</td>
</tr>
<tr>
<td>Standardization</td>
<td>To what extent is the process based on a standard generic process? To what extent is the same process used in all parts of the company?</td>
</tr>
<tr>
<td>Visibility</td>
<td>Do the process activities culminate in clear results, so that the progress of the process is externally visible?</td>
</tr>
<tr>
<td>Measurability</td>
<td>Does the process include data collection or other activities that allow process or product characteristics to be measured?</td>
</tr>
<tr>
<td>Supportability</td>
<td>To what extent can software tools be used to support the process activities?</td>
</tr>
<tr>
<td>Acceptability</td>
<td>Is the defined process acceptable to and usable by the engineers responsible for producing the software product?</td>
</tr>
<tr>
<td>Reliability</td>
<td>Is the process designed in such a way that process errors are avoided or trapped before they result in product errors?</td>
</tr>
<tr>
<td>Robustness</td>
<td>Can the process continue in spite of unexpected problems?</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Can the process evolve to reflect changing organizational requirements or identified process improvements?</td>
</tr>
<tr>
<td>Rapidity</td>
<td>How fast can the process of delivering a system from a given specification be completed?</td>
</tr>
</tbody>
</table>
2.1.3 Software development processes models and methods

Due to a wide variety of organizational profiles and specifics of the products produced by different software companies, it is impossible to have one “ideal” prescribed software process that would fit all. This, however, does not mean that the software companies have no guidelines on how to structure their software processes. Instead, the organizations can choose from a wide range of different software process models, and software development methods that are designed to guide towards suitable and efficient software processes.

Software process models and methods started to emerge in early 1970s with the Waterfall model (Royce, 1970). Hundreds of other models and methods have been created since then. However, only some of them have become widely accepted today. Among these are Waterfall model, (Royce, 1970), V-Model, (German Ministry of Defence, 1992), Spiral model, (Boehm, 1986), Iterative model, (Randell & Zürcher, 1968), Rational unified process, (Kruchten, 2003), Model driven development, (OMG, 2001), Agile development, (Beck, et al., 2001), Test driven development, (Beck, 2003), Crystal, (Cockburn, 2004), Scrum, (Schwaber, 1995), XP, (Beck, 2000), Lean (Poppendieck, 2002), and Kanban (Kniberg & Skarin, 2010). Below we present several of the best known and most commonly used software development models and methods.

**Waterfall model**

The first formal software development model originated from manufacturing and construction industries. Its first formal description was presented in 1970 by Winston Royce (Royce, 1970). Royce’s original model consisted of the following phases: requirements specification, design, construction, integration, testing and debugging, installation and maintenance. The process was structured as a cascade from one phase to another. Hence it is known as Waterfall model (Sommerville, 2011). The model defines software development phases, through which one goes in a sequential manner. Only when the previous phase is fully completed, one proceeds to the next one. The initially defined principal phases of the model have been continuously enhanced and modified and as a result today the model presented by different authors can include slight or major variations (Pfleeger & Atlee, 2006), (Sommerville, 2011), (Maciaszek & Liong, 2005). The Waterfall model by Sommerville is presented in Figure 2.2 (2011).
The software development based on the Waterfall model is usually supported by a large amount of documentation of each development phase (Sommerville, 2011). The Waterfall model, however, does not provide enough flexibility to respond to changing customer requirements since commitments are made in the early stage of the process (Sommerville, 2011). Therefore, this model is suitable for larger software engineering projects, where the requirements do not change during system development.

**V-model**

A further development of the Waterfall model resulted in V-model (German Ministry of Defence, 1992). The V-model consists of similar process phases as the Waterfall model, but with a focus on demonstrating how testing and validation activities relate to requirements and design (Pfleeger & Atlee, 2006). As shown in Figure 2.3, the process phases are formed in typical “V” shape. The linkage between the left and right sides of the “V” aims to verify that requirements and design are implemented completely and correctly (Pfleeger & Atlee, 2006).

**Iterative development model**

An iterative software development model assumes that software development is done in a set of short iterations. As shown in Figure 2.4, each iteration represents a small waterfall of typical software engineering activities, resulting in a preliminary software product (Maciaszek & Liong, 2006). The access to the preliminary software product allows the customer to provide feedback on its early versions and refine system requirements for the next iteration. The software product resulted in the previous iteration is always input for the next iteration. The iterations normally continue until the desired software product is fully developed.

**Agile software development**

Agile software development is built on the iterative and incremental software development, allowing quick feedback loops in the process. Thus, the agile development can easily adapt to changing user requirements during the project development. In the agile development, direct

---

**Figure 2.3  V-model, (Pfleeger & Atlee, 2006)**
Figure 2.4 Iterative process model, (Maciaszek & Liong, 2006)

face-to-face communication, people and customer collaboration are preferred to the written
documentation, process and contract negotiation (Beck, et al., 2001). Agile methods are
suitable only for some types of system development and are not recommended for large-scale
projects or critical system development (Sommerville, 2011). A large number of different agile
methods is widely used today. They all follow the mentioned agile principles but are different
in the practices and activities they apply. The agile methods described below are Scrum, eXtreme Programming and Kanban.

Scrum

Scrum is an agile software development method, which was originally defined in 1995
(Schwaber, 1995). It is an adaptive, quick and self-organizing method, which primarily focuses
on process flexibility, adaptability and productivity (Schwaber & Beedle, 2001). It aims at
addressing project uncertainty by combining an empirical approach and consistent control
over the development process (Schwaber, 1995). Scrum requires self-organizing autonomous
teams, and defines specific roles of Team, Scrum master, and Product owner. In Scrum the
software development process is divided into two to four weeks long iterations, called sprints.
During each sprint, the team develops a sprint backlog, resulting in a piece of potentially
shippable software. Scrum is supported by a set of specific meetings, such as Sprint planning
meetings, Daily stand up meetings, Sprint demos and Retrospectives.

eXtreme Programming (XP)

eXtreme Programming (XP) is a widely used agile software development method (Beck, 2000).
It includes a collection of best development practices focused on improving software quality
and customer satisfaction (Sommerville, 2011). The method emphasizes four characteristics
of agile methodology. These are communication, simplicity, courage and feedback (Pfleeger &
Atlee, 2006). Therefore, in an XP process, developers communicate with themselves and with
their customers; they create simple system design; they receive frequent and early feedback
on the system quality and functionality and they openly respond to changing requirements.
XP is based on interconnected practices, principles and values. The characteristics embedded into the XP method, known as twelve facets of XP, are the following: planning game, small releases, metaphor (a common vision of the system), simple designs, writing tests first, refactoring, pair programming, collective code ownership, continues integration, coding standards, sustainable pace and onsite customer (Pfleeger & Atlee, 2006).

The XP project starts with development of high level system requirements, called user stories (Sommerville, 2011). The user stories are then broken down to tasks, and time and effort for their implementation is estimated. After it, the user stories are prioritized by a customer, constituting the main input for the release planning. The development process is structured in two to three week long iterations. To enable frequent feedback loops, the acceptance tests are conducted during the iteration just when a piece of functionality is implemented. After every iteration, the functionality is delivered to the customer for approval.

Kanban

Kanban is based on Just-In-Time and Lean production systems. It has been developed by Toyota and has been used in Toyota Production Line for decades (Ohno, 1988). The method, however, is still reasonably new in software engineering, but is gaining an increasing popularity (Boeg, 2012). In contrast to Scrum or XP, Kanban is less prescriptive. It does not include development iterations, it does not define roles or meetings, and finally, it does not contain process artifacts (Kniberg & Skarin, 2010). Kanban is often seen as a change management method that is focused on the following base principles:

- **Visualize Work**: visualize every major step of the process going from vague concept to releasable software,
- **Limit Work-In-Progress (WIP)**: set limits to the maximum amount of work in progress allowed in each stage,
- **Make Policies Explicit**: define and make explicit the policies you are following,
- **Measure and Manage Flow**: take measurements of the development flow in order to manage it,
- **Identify Improvement Opportunities**: create and adopt a culture of continuous improvements, called Kaizen (Boeg, 2012).

### 2.2 Software process improvement

The importance of quality, efficiency and composition of software process activities has been stressed by many authors (Pfleeger & Atlee, 2006), (Sommerville, 2011). The reason for it is that the quality of the software process affects the cost of software development, ultimate software product quality, and time to market of the product. Therefore, in order to maintain and increase their competitive advantage, many software organizations strive to improve their development processes. They do it by conducting Software Process Improvement (SPI).

The software process improvement is commonly focused on the needs of the software organization and the weaknesses of the existing process (Sommerville, 2011). SPI is often implemented through a series of improvement cycles, when the process is continuously changed and the new practices are continuously added (Robillard, et al., 2003). Even though different authors often disagree on the principal stages of SPI process, most of them agree that
the SPI process should be continuous and cyclical (Zahran, 1998), (Sommerville, 2011), (Dybå, et al., 2004), (Yeakley & Fiebrich, 2007).

The commonly accepted general cyclical model of SPI was developed in 1930s by Walter Shewhart (1939), and further evolved by Deming (1950), Imai (1986), and Ishikawa (1985). It is known as PDCA model or PDCA cycle. The model is named after its four cyclical stages: Plan, Do, Check, and Act. As illustrated in Figure 2.5, the cyclical stages of the model coordinate the continuous improvement effort, by emphasizing careful planning and effective action. The cyclic phases of the PDCA model are the following:

- **Plan** (SPI planning): aiming at defining process improvement goals and vision, as well as identifying process improvement activities and creating a process change plan.
- **Do** (Process change): aiming at changing the process according to the planned improvement on the small scale or a pilot project.
- **Check** (Process review): aiming at assessing/reassessing/measuring the process according to the goals of the process improvement, analyzing the process and its measurements and comparing them to the expected results.
- **Act** (Scaling the improvements): aiming at rolling out the improvements to the larger scale.

SPI initiatives should have a clear direction, goals and vision of the desired/improved process. Those can be expressed by a process improvement roadmap. The importance of the roadmap was illustrated by Watts Humphrey (1989) with a Chinese proverb: “If you don’t know where you are going any road will do”. The process improvement roadmap is important for focusing and guiding the improvement efforts towards the desired process.

Based on the choice of the software process improvement roadmap, the SPI approaches can be divided into two main types: 1) reference model driven SPI, which focuses the SPI effort on the continuous process standardization and introduction of recommended processes and practices, and 2) inductive SPI, which focuses the SPI effort on identifying problems, weaknesses and defects of the process and eliminating them. In addition to the two main types of SPI approaches, in this thesis we propose a third type of improving software processes, which is software method adoption. Since the adoption of the new development method may improve the software process, the adoption of a new method or a set of new practices can be seen as process improvement effort. The three types of SPI efforts are presented in the following sections.

![Figure 2.5  PDCA model](image-url)
2.2.1 Reference model driven SPI

Reference model driven SPI is an improvement effort driven by process standardization and based on the implementation of prescribed processes and best practices advocated by a reference model. The reference model aims to guide the evolution of the processes in the organization towards the processes that are recommended by it. The best known reference models in software engineering industry are process capability and maturity models (CMMI Product Team, 2002) (Dorling, 1993). They are designed to help software organizations to assess the status of their software processes and identify the areas of future improvement.

The process capability and maturity models, however, do not provide guidelines on how to change the process and conduct those improvements. Therefore, the maturity models should be supported by process improvement guidelines and process models. The best known maturity models are CMMI (Capability Maturity Model Integration) (CMMI Product Team, 2002) and SPICE (Software Process Improvement and Capability dEtermination) (Dorling, 1993). They are supported by specifically designed process change models that aim to guide initiation, planning and implementation of the improvement actions. CMMI is supported by a model called IDEAL (McFeeley, 1996), and SPICE is supported by ISO/IEC 15504-4 (ISO/IEC, 2004). Below we present the mentioned reference models and the process change models for their implementation.

Process reference models

Capability Maturity Model Integration (CMMI) was developed by SEI, and was built upon the retiring well-known model called CMM (Capability Maturity Model) (Paulk, et al., 1993). CMMI reference model provides software organizations with a roadmap for process standardization and improvement. The model covers twenty two software development process areas, and has two representations: staged and continuous. Its staged representation allows organization to assess its development processes and assign a maturity level from 1 to 5 across organization’s multiply process areas (CMMI Product Team, 2002). CMMI’s five maturity levels are presented in Figure 2.6, and are the following: Initial, Managed, Defined, Quantitatively managed, and Optimizing (CMMI Product Team, 2002). What maturity level the organization achieves is governed by what processes (process areas) are performed in accordance to CMMI, see Figure 2.6 (CMMI Product Team, 2002). Each maturity level is linked to a set of process areas, governing the order in which the process areas are implemented. The continuous representation comprises six capability levels and allows the organization to focus on the process areas they consider important and assign capability levels to individual process areas (CMMI Product Team, 2002). This allows flexibility of the order in which process areas are implemented or improved.

Software Process Improvement and Capability dEtermination (SPICE) model, also known as ISO/IEC 15504, is another international reference model for process assessment and improvement. The SPICE model provides tools for standardized process assessment and its capability determination (ISO/IEC, 2004). Even though SPICE offers only continuous representation, it has somewhat similar structure to CMMI. In the similar way as CMMI, SPICE is organized into six capability levels that are used to characterize the individual processes. The capability levels are the following: Incomplete process, Performed process,
Figure 2.6  CMMI maturity levels and its corresponding process areas, (CMMI Product Team, 2002)

Figure 2.7  SPICE capability levels and its corresponding process attributes, (Dorling, 1993)
Managed process, Established process, Predictable process and Optimizing process. The capability levels, in turn, are measured by process attributes that list process characteristics that must be reached to obtain a certain capability level, see Figure 2.7. The process attributes in turn further consist of generic practices. The SPICE model guides the process improvement efforts towards reaching higher levels of process maturity. (ISO/IEC, 2004) (Dorling, 1993) (Kinnula, 2001)

Process models for implementation of reference models

The reference models provide a base for process assessment and recommend what software processes need to be introduced to the organization next. They, however, do not provide guidelines on how to introduce the recommended processes. Therefore, they are supported by specifically designed models that aim to guide initiation, planning and implementation of the improvement actions, such as the IDEAL model (McFeeley, 1996), or ISO/IEC 15504-4 (ISO/IEC, 2004).

The IDEAL model is specifically designed to support the implementation of CMM or CMMI maturity models (McFeeley, 1996). The model is named after its five phases: 1) Initialize: start the improvement program, 2) Diagnose: assess the current state of the process, 3) Establish: set the implementation strategy and improvement program, 4) Act: implement process improvements, and 5) Leverage: analyze the improvement effort and revise the approach (McFeeley, 1996). As shown in Figure 2.8, the five phases of the IDEAL model contain activities that together represent the process of continuous software improvement. The IDEAL model can be used as a roadmap for initiating, planning, and implementing the improvement actions (McFeeley, 1996).

ISO/IEC 15504-4 is a guide for process improvements that was published as part of the SPICE/ISO/IEC 15504 reference model. ISO/IEC 15504-4 provides an eight-step process model for continuous software process improvements (ISO/IEC, 2004). The eight steps are the following:

- **Initiating**
  - Set context and establish sponsorship

- **Diagnosing**
  - Appraise and characterize current practice
  - Develop recommendations and document results

- **Establishing**
  - Set strategy and priorities

- **Acting**
  - Plan actions
  - Establish action teams
  - Plan, execute and track installation
  - Define processes and measures

- **Leveraging**
  - Revise organizational approach
  - Document and analyze lessons

Figure 2.8  IDEAL model, (McFeeley, 1996)
1. Examine organization’s business goals
2. Initiate process improvement cycle
3. Assess current capability
4. Develop action plan
5. Implement improvements
6. Confirm improvements
7. Sustain improvements
8. Monitor performance

As shown in Figure 2.9, the eight steps of the model and their relationships represent the high level process of the process improvement and implementation of the SPICE/ISO/IEC 15504 assessment standards. The ISO/IEC 15504-4 model is based on the use of metrics and process assessment results, which in turn derive the improvement action plan (ISO/IEC, 2004). This implies that the process improvement actions target the desired capability profile of the organization, and the results are measured and monitored through metrics.

### 2.2.2 Inductive SPI approaches

Inductive SPI approaches focus on understanding the organization, its process, and its problems and challenges, as well as on guiding the SPI towards resolving process related
problems and reducing software defects. The inductive SPI approaches are also called, problem driven SPI and experience based SPI. Inductive SPI efforts are tailored to organizational needs and goals and usually improve the software process in a customized manner. They also incorporate learning from previous experience and integrate lessons learned when shaping the SPI method.

Unlike the reference models, the inductive SPI approaches do not impose any set of processes to be introduced. Instead, during the inductive SPI effort, the changes to the process are focused on the elimination or minimization of process related problems and defects. Therefore, the inductive SPI approaches provide guidelines or process models on how to change the process and conduct the improvements. The well-known SPI approaches that follow the inductive reasoning are: Quality Improvement Paradigm (QIP) (Basili, 1985), and Six Sigma (Bhote, 1989).

Quality Improvement Paradigm

The Quality Improvement Paradigm (QIP) was developed by Viktor Basili in 1985 (Basili, 1985). This model for continuous process improvement allows the organization to develop and improve their processes through experimenting rather than matching the processes against certain pre-defined reference model (Basili, 1985) (Basili, 1989). The QIP model also encourages learning from experience and capturing and packaging the best practices in a reusable form (Kinnula, 2001). As presented in Figure 2.10, the QIP model, referred to as the QIP cycle, consists of two closed-loop cycles: project cycle and organizational cycle. The project cycle provides feedback to the project during the project, while the organizational cycle provides feedback to the organization after the project completion (Kinnula, 2001).

![Figure 2.10 QIP cycle, (Kinnula, 2001)](image-url)
QIP cycle, the SPI process begins by characterizing and understanding current software processes, based on which the measurable goals for the improvement are set. Then the new processes, techniques and tools for the processes are chosen and implemented as part of project cycle. Finally, the results of the completed project are gathered, analyzed, packaged and distributed across the organization.

**Six Sigma**

Six Sigma is a business management approach for improving engineering and development processes. It originated from manufacturing processes and was first introduced by Motorola in 1985 (Bhote, 1989). In recent years, Six Sigma has been introduced to improve software development processes and has been often combined with Lean (Poppendieck, 2002). Six Sigma is a disciplined data driven approach aiming at improving a development process by identifying and eliminating costs that provide no value to customers: waste costs, such as introducing and fixing defects in the system (Park, 2003), (Pyzdek & Keller, 2010). Using a measurement-based strategy, Six Sigma defines how the process is performing and how it should be improved, in order to achieve three goals: to reduce costs, to improve customer satisfaction, and increase revenue and profits (Park, 2003). Six Sigma provides a process improvement cycle for guiding the improvement efforts, called DMAIC. DMAIC stands from its five phases and aims to improve existing processes. DMAIC’s five iterative phases are the following: 1) Define the goals of the improvement project, 2) Measure the process attributes with respect to its quality and efficiency, 3) Analyze the process or the system, and identify ways to eliminate the gap between the current state and the desired state of the process or the system, 4) Improve the process by eliminating the defined defects or problems, and 5) implement the Control mechanisms for sustaining the achieved improvements (ISixSigma, 2010), (Pyzdek & Keller, 2010). Six Sigma uses another improvement cycle, called DMADV, for creating new process designs (Harry & Schroeder, 2000), (ISixSigma, 2010).

### 2.2.3 Software method adoption

Different software development methods illustrate how to structure software development in an efficient and effective manner. The best known and most commonly used software development methods were presented in section 2.1.3. Since the suitable to the organization software development methods may provide an easy to use guide for the efficient software development processes, many organizations choose to adopt new software development methods in order to improve their existing software processes. Therefore, in this thesis we propose software method adoption as an alternative way of implementing SPI.

The software method adoptions, also called process transformations, allow the organizations to start afresh and adopt new software methods instead of improving the existing ones (Nikitina & Kajko-Mattsson, 2011a). Growing amount of software organizations adopt agile software development methods. Still however, based on (Sahota, 2012) as many as 84% of those organizations experience failure of the adoption projects. Adopting a software method is a highly complex process that includes changes of not only software development process but also changes of the organizational culture and social patterns and behaviors of stakeholders involved (Sahota, 2012), (Laanti, et al., 2011), (Svensson & Höst, 2005).
Despite the large number of software method adoptions, there are very few guidelines that the organizations can use on how to adopt new software methods. There is no widely accepted and clearly defined method for software method adoption. Therefore, the organizations adopt new methods in an ad-hoc manner or use slightly relevant methods for process improvements (Nikitina & Kajko-Mattsson, 2011a), (Nikitina, et al., 2012). In this thesis we propose a Software Method Adoption (SMA) process model for guiding the new method adoptions, which is presented in this Section 5.5.

2.3 Framework for SPI environment

Building on the elements of the SPI effort presented in the previous sections, various aspects of SPI can be best represented by a framework for SPI environment proposed by Sami Zahran (1998). Zahran stresses the importance of effective software process environment which can lead to continuous software process improvement. The effective software process environment should consist of mechanisms for facilitating the establishment of a process culture and infrastructure to support software development (Zahran, 1998). Moreover, he states that to achieve the effective process environment, the following aspects should be covered: process documentation, process training, and process monitoring and enforcement. More specifically, the organizational mechanisms that should be in place to establish the effective process environment are the following: process ownership, process training, measurements of process results, monitoring of process performance, feedback from users, feedback from external environment, and process inspections and enforcement (Zahran, 1998).

The framework for SPI environment proposed by Zahran is shown on Figure 2.11. It consists of four following main components that need to be in place in order to build a continuous SPI environment (Zahran, 1998).

1) Software process infrastructure: Both software processes and SPI efforts need to be supported by infrastructure, which includes both human resources (organizational and management infrastructure), and technical tools and facilities (technical infrastructure). Zahran outlines that organizational and management infrastructure covers roles and responsibilities for establishing, monitoring, and enforcing the SPI (1998). Those roles are the following: sponsorship roles, management roles, coordination roles, and improvement teams role (Zahran, 1998). The technical infrastructure, on the other hand, covers technical platforms, computing facilities and tools to support the software process and SPI teams.

2) Software process improvement roadmap: The improvement roadmap specifies the desired process state to be achieved by software process and criteria for achieving this state. It can be
one of the standard maturity models, such as CMMI or SPICE, or a tailored organizational version to match organizational needs (Zahran, 1998). It is however important that the organization has a clearly defined roadmap and set of goals for the SPI effort.

3) **Software process assessment**: The assessment method should specify the techniques for assessing the organizations' current software process, practices and infrastructure (Zahran, 1998). The assessment method is normally strongly linked to the chosen improvement roadmap. The results of the assessment provide a baseline of the current status of the process and should outcome in the recommendations for improving the process effectiveness (Zahran, 1998).

4) **Software process improvement action plan**: The findings from the process assessment are transformed into specific actions for process change, which have to be undertaken to improve the process infrastructure, discipline and effectiveness (Zahran, 1998). The implementation of those actions is essentially about introducing the change to the organization. The *SPI action plan* consists of the list of the actions (activities), their schedule and milestones, as well as allocated resources, and budget.

This framework does not sufficiently cover aspects of the organizational change. It, however, provides a solid base for a broader perspective to the SPI effort. That is why this framework is used as a base to outline the *SPIF* framework that resulted from this research and is presented in Section 5.2.

### 2.4 Organizational change management

Clearly defined SPI roadmap and method are inevitable elements for succeeding with the SPI initiative. The SPI roadmaps and methods focus on technical, instrumental, procedural and organizational aspects of SPI, allowing the SPI process to be structured, managed and organized. This however is not enough. Other relevant aspects of SPI such as context and people should be considered (Dybå, 2002), (Ferreira & Wazlawick, 2011). The SPI initiatives, regardless of the chosen SPI method, roadmap and strategy, impose changes to the software processes, which in turn impact organization and business, as well as people and their behaviors. This can be considered an *organizational change* (Pries-Heje & Johansen, 2010), (Heikkila, 2009). The importance of the management of the organizational changes, of the incorporation of the organizational culture and tightly integrated in it social issues to the SPI initiatives has been recognized by number of researchers (Kautz, et al., 2001), (Kautz & Nielsen, 2004), (Sahota, 2012), (Muller, et al., 2008), (Heikkila, 2009), (Muller, et al., 2010), (Ferreira & Wazlawick, 2011), (Mathiassen, et al., 2005), (Kandt, 2003). Still however, the literature within SPI seldom incorporates organizational change theory (Aaen, et al., 2001).

The literature from *organizational science* explores *organizational change* and lists many different dimensions of it. According to Nguyen-Huy, *organizational change* contains four basic elements (2001). The elements are the following: 1) *formal structures* implying the official allocation of resources and division of responsibilities among the employees, 2) *work processes* implying the processes under change, 3) *systems and shared beliefs* implying the shared values, norms and beliefs of the employees, and 4) *social relationships* implying the nature and quality of the interpersonal interactions between the employees (Nguyen-Huy, 2001). The recognition
of the first two elements, formal structure and work processes, by the literature within SPI can be seen in a wide variety of the process maturity models and SPI methods. The process change however cannot be performed without introducing the changes to systems and shared beliefs and social relationships, represented by organizational culture.

### 2.4.1 Change management

Common barriers and problems of SPI are related to human and organizational factors (Ferreira & Wazlawick, 2011). In many cases, the process improvements and changes are linked to the people’s resistance to change, frustration and disappointment. Therefore, the success in SPI heavily depends on how the organizational change has been managed and perceived by people involved (Muller, et al., 2010), (Kautz, et al., 2001), (Kautz & Nielsen, 2004), (Ferreira & Wazlawick, 2011).

In order to bring organizational change, successful SPI initiatives require effective change management, which can manage organizational and behavioral changes – changes to the organizational culture, and to the way people work and collaborate (Heikkila, 2009). Effective change management includes established strategy, detailed planning, flexibility and commitment in executing the plans, understanding of the changes to the organizational culture, and consideration of the stakeholders’ attitude towards change (Heikkila, 2009).

The literature on change management comes from organizational science and psychology. Due to a wide range of change management approaches and very limited incorporation of organizational change theory to the SPI, it is difficult to point out the predominant change management approach that would be suitable for SPI (Heikkila, 2009). Nonetheless, the change management approaches that this study found most suitable for the SPI context are listed below.

The commonly accepted model for the organizational change was developed in 1947 by Kurt Lewin (1947). Still, however, the concept remains relevant today. Lewin perceives the organizational change as a social change, focusing on the careful change of the social processes. He highlights the importance of social habits, which may not change due to resistance to change. He also recognizes group standards and group procedures, as a tool for managing the resistance to change. According to Levin, “it is easier to change individuals formed into a group than to change any one of them separately” (Lewin, 1947). Therefore, a successful organizational change includes the following aspects:

- **Unfreezing**: mental unfreezing of group standards, group procedures and organizational policies. It is often done by presenting the provocative problem or situation in order to make people recognize the need for change and to search for solutions.
- **Moving**: developing new behaviors, opinions, experiences and values. It is often done by adopting changes to the process and organizational structure.
- **Freezing**: freezing newly established changes, group standards, group procedures and organizational policies. Unless the newly established changes are frozen and established, the process and social behaviors may return to its original state (Lewin, 1947), (Bartoli & Hermel, 2004).
Building on Lewin’s reasoning, John Kotter proposed an extended approach for the organizational change (1995). Among other things, Kotter put in focus the need for leadership and support for organizational change (1995). Based on the experiences of successful and unsuccessful organizational changes, he proposed an approach for organizational change consisting of eight consecutive steps. The steps are the following:

1. **Establishing the sense of urgency**: by examining market and current situation critically, as well as by identifying major opportunities and solutions;
2. **Forming a powerful guiding coalition**: by creating a team who will lead the change, and by encouraging the cohesiveness of the team;
3. **Creating a vision**: and developing strategies for reaching that vision, in order to guide and direct the change effort;
4. **Communicating the vision**: and teaching new behaviors, using all the available communication tools and patterns as means;
5. **Empowering others to act on the vision**: and eliminating obstacles of change, by encouraging change, innovation and initiatives towards reaching the vision;
6. **Planning for and creating short term wins**: by planning for and implementing changes that bring improvements, and rewarding employees involved in it;
7. **Consolidating improvements and producing more change**: by promoting and developing employees who can implement the vision, and by strengthening the changed process with new projects and change agents;
8. **Institutionalizing new approaches**: by highlighting the results of the changed process, and by developing the means to ensure leadership development (Kotter, 1995).

Kotter’s approach was applied for managing process improvement within few industrial case studies (Hayes & Richardsson, 2008), and was further evolved by Kandt in order to address specific needs of SPI (Kandt, 2003). Kandt outlined the program for managing the organizational and cultural change during the SPI. This program focused on the incorporation of the organizational and cultural issues, and thus, aided to the acceptance of the software process changes by the practitioners (Kandt, 2003). The program consisted of ten following steps:

1. Define an organizational vision
2. Articulate a compelling need for change
3. Define a change effort and vision
4. Obtain management sponsorship and commitment
5. Adopt a change approach
6. Identify and mitigate risks
7. Align the training program with the change effort
8. Align the reward and recognition program with the change effort
9. Communicate the change effort often and effectively
Kandt’s approach highlights the importance of social factors on the change efforts. Therefore, according to Kandt, all the software organizations that change their software processes must adopt four critical principles, which are: 1) create a vision for the future organization, 2) achieve executive commitment, 3) involve practitioners in change definition, and 4) communicate change vision to the organization (Kandt, 2003).

To summarize, when organizations are to change their software processes, the literature points out that they need to convince their employees to buy-in to the process change. The complex process of SPI should be well guided and carefully managed to convince and train people in the future changes, as well as to involve them in the change process. Moreover, the change process needs to have a clear vision and strong leadership. Management support, institutionalization of the changed process, and enforcement of the changed process by means of organizational policies are final elements needed to establish the changes and support the introduction of the social change.

2.4.2 Change of the organizational culture

The management of the organizational change is a key factor for successful introduction of the SPI. The main challenge of the organizational change management lays in changing the systems and shared beliefs and social relationships referred to as organizational culture (Heikkila, 2009), (Nguyen-Huy, 2001). The existing organizational culture needs to suit the newly implemented process, expressed by managing human dynamics of commitment and resistance (Heikkila, 2009). A survey done by VersionOne, shows that inability to change and adopt organizational culture is seen as the biggest barrier to the adoption of the agile methods (VersionOne, 2013). Fifty two percent of the surveyed organizations reported it as the biggest challenge for their organizations (VersionOne, 2013). This however has not been confirmed by other studies done on the barriers of the SPI initiatives (Niazi, et al., 2010), (Niazi, 2009). The reason for it can be that the cultural impacts on SPI are challenging to identify, and the mainstream research in SPI focuses exclusively on the technical, procedural and instrumental aspects, and often disregard the impact of the organizational culture (Sahota, 2012). Despite the recognized importance of organizational culture, there is no mainstream approach on how to change organizational culture in order to support the introduction or improvement of a software process.

The organizational culture or corporate culture was first defined by Edgar Schein as “the accumulated learning that a given group has acquired during its history” (1988). Every organization, including software organizations, has its own culture, character and identity, referred to as organizational culture (Schneider, 2000). Organizational culture is stronger in the larger, older and more successful organizations (Schneider, 2000). The culture consists of organization’s core values, its norms, principles, beliefs and stories (Anderson & Ackerman Anderson, 2010). It gets embedded into the organization over time through its policies, procedures, practices, methods and ways of operating (Anderson & Ackerman Anderson, 2010). Edgar Schein stressed the importance of the organizational culture by saying “if you do not manage the culture, it manages you and you may not even be aware of the extent to which this is happening” (1988).
In order to manage organizational culture and support process change, organizations first need to understand their current culture and then to define how this culture needs to change to support the newly changed software process. For analyzing the organizational culture, one may use the models of the organizational culture. As argued by Sahota (2012), one of the most suitable for the context of software process improvement is Culture model by William Schneider (1994).

Schneider’s Culture model is a framework for understanding and measuring the organizational culture. It is focused around the idea that every successful organization has a core culture, which should be aligned with organization’s strategy and core leadership practices (Schneider, 2000). As shown in Table 2.2, Schneider’s Culture model defines the four following distinct core cultures:

1) Control culture has power as a primarily motive, and is focused on the achievement of the organizational goal. It ensures certainty, predictability, safety, accuracy and dependability,
2) Collaboration culture has synergy as a primarily motive, and is focused around the collective experience of people. It ensures unity, close connection with and dedication to the customer,
3) Competence culture has fundamental motive of accomplishment of conceptual goals. It ensures distinction and uniqueness,
4) Cultivation culture is motivated by self-actualization and further realization of ideas and values. It ensures the growth of the customer and fulfillment of customer’s potential, (Schneider, 2000).

<table>
<thead>
<tr>
<th>Control</th>
<th>Collaboration</th>
<th>Competence</th>
<th>Cultivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market share</td>
<td>Synergic customer</td>
<td>Superiority</td>
<td>Customer growth</td>
</tr>
<tr>
<td>Commodity</td>
<td>relationship</td>
<td>Excellence</td>
<td>Enrichment of customer</td>
</tr>
<tr>
<td>Predictability</td>
<td>Close partnership</td>
<td>Uniqueness</td>
<td>Fuller realization of potential</td>
</tr>
<tr>
<td></td>
<td>with the customer</td>
<td>Create market</td>
<td></td>
</tr>
<tr>
<td></td>
<td>High customization</td>
<td>niche</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constant</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>innovation</td>
<td></td>
</tr>
<tr>
<td>Strategy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authoritative</td>
<td>Team builder</td>
<td>Standard setter</td>
<td>Catalyst</td>
</tr>
<tr>
<td>Directive</td>
<td>Coach</td>
<td>Conceptual</td>
<td>Cultivator</td>
</tr>
<tr>
<td>Conservative</td>
<td>Participative</td>
<td>visionary</td>
<td>Commitment builder</td>
</tr>
<tr>
<td>Commanding</td>
<td>Integrator</td>
<td>Task master</td>
<td>Steward</td>
</tr>
<tr>
<td></td>
<td>Trust builder</td>
<td>Challenger to</td>
<td>Appeal to higher level vision</td>
</tr>
<tr>
<td></td>
<td></td>
<td>others</td>
<td></td>
</tr>
<tr>
<td>Leadership</td>
<td></td>
<td>Convincing</td>
<td></td>
</tr>
<tr>
<td>Certainty</td>
<td>Synergy</td>
<td>Distinction</td>
<td></td>
</tr>
<tr>
<td>Organizational</td>
<td>Experimental</td>
<td>Conceptual</td>
<td></td>
</tr>
<tr>
<td>systematism</td>
<td>knowing</td>
<td>systematism</td>
<td></td>
</tr>
<tr>
<td>Epistemology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertical integration</td>
<td>Diversity</td>
<td>Knowledge-capital</td>
<td>Flexible organization</td>
</tr>
<tr>
<td>Mass customization</td>
<td>Open book management</td>
<td>measures</td>
<td>Empowerment</td>
</tr>
<tr>
<td>Statistical process control</td>
<td>Self-directed teams</td>
<td>Benchmarking</td>
<td>Open book management</td>
</tr>
<tr>
<td>Formal processes</td>
<td>Open offices</td>
<td>Entrepreneurial</td>
<td>Seeking employee commitment</td>
</tr>
<tr>
<td></td>
<td>Group oriented</td>
<td>management</td>
<td></td>
</tr>
<tr>
<td>Management ideas</td>
<td>leadership</td>
<td>Best practices</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Continuous</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2 Schneider’s four core cultures types, (Schneider, 2000)
Schneider argues that if the desired software process and change strategy fits the nature of the organizational culture, then the probability is high that the process change effort would be successful (Schneider, 1994). On the other hand, if the desired software process or change strategy does not suit the existing organizational culture, and the organizational culture has not been adapted to the change, then the change effort would most likely fail in the long run. According to Schneider, it is essential to work with the existing culture rather than fight it (Schneider, 1994). For that, organizations should have a good understanding of their culture and an acknowledgement of its strength.

Schneider’s Culture model was further evolved by Michael Sahota and Agilitrix, and used as an illustrative guide for adjusting organizational culture to support the adoption of agile methods (Sahota, 2012), (Agilitrix, 2011), see Figure 2.12. Sahota aligned the agile development methods to the structure of Schneider’s four core cultures, illustrating the dominant culture of the methods. He also presented an approach for adjusting the organizational culture during the software method adoption (Sahota, 2012). As can be seen in Figure 2.12, he recommends two options for working with the existing culture: 1) Option 1 implying not to change the culture and work with the existing culture, and 2) Option 2 implying to carefully explore the cultures that are similar to the core culture, choose the desired culture (often based on the secondary organizational culture), and start slowly adopting principles of the desired culture. Those solutions go in line with Schneider’s principle of working with existing organizational culture.

As mentioned in this chapter, the SPI roadmaps and methods focus on technical, instrumental, procedural and organizational aspects of SPI. This however is not enough. The
SPI initiatives impose changes to the software processes, which in turn impact organization and business, as well as people and their behaviors (Pries-Heje & Johansen, 2010), (Heikkila, 2009), (Sahota, 2012). Therefore, the strategies and approaches for organizational change, mentioned in this chapter, need to be incorporated to the existing body of knowledge on SPI. This research and the results that emerged from it and are presented in Chapter 5 constitute an effort for doing this.
Chapter 3

Research approach

This chapter presents the research approach used during this study. Section 3.1 and Section 3.2 describe the research paradigm and strategy that guided this research. Then, Section 3.3 presents the process of this research. Section 3.4 and Section 3.5 report on the research method, data collection and analysis approaches used during this study. Section 3.6 lists the organizations involved in this research. Finally, Section 3.7 describes validity threads of this study and how they were minimized.

3.1 Research paradigm

Research paradigms are those that relate to the underlying “knowledge” or epistemology that guides the research towards the most suitable and appropriate research method for answering the research questions (Orlikowski & Bauroudi, 1991), (Myers, 2009). Therefore, in order to find the most appropriate method for this study, we first chose a research paradigm suitable for the research problem and goal. Since the study aimed to add to the solution of an industrial research problem, and the phenomenon being studied was highly impacted by its context, as it is motivated below, the interpretivist paradigm was suggested to guide this study.

Interpretivism assumes that reality can be archived only by means of social constructions. It emphasizes the subjective meaning of the reality and highlights the impact of context and people involved, including the researchers (Myers, 2009), (Orlikowski & Bauroudi, 1991). According to the interpretivist paradigm, the organizations, groups and social systems cannot exist apart from humans and therefore cannot be defined, characterized and measured in some objective and universal way (Orlikowski & Bauroudi, 1991). By using interpretivism, one could achieve a better and comprehensive understanding/view of the phenomena that is being studied.

Due to high significance of the context and the influence of the organization and people involved in SPI process, the interpretivism was chosen as a leading paradigm for conducting this study. It aids to the understanding of social and contextual situations that influence the SPI process. Moreover, since software organizations are the primary target audience of this research and the research problem is an existing industry problem, it is important that the research done within this field considers the role of context and real world setting. This is because it has been recognized that practitioners find more useful studies that provide detailed interpretation of a case in specific organizational and social contexts (Walsham, 1995). The lessons and recommendations given for a specific case can be easily applied by other practitioners, if the context allows it.

Interpretivism guides the research towards the methods which recognize the importance of the context and assume that the reality is subjective. In order to understand the meaning of human and social interactions, interpretivists need to engage in the social settings investigated and learn about the context from the participants’ perspective (Porta & Keating,
2008), using mainly qualitative research methods such as field studies and action research, since these methods examine humans within their social settings (Orlikowski & Bauroudi, 1991). It, however, also makes use of quantitative data collection methods, if needed (Porta & Keating, 2008). Interpretivists mostly work inductively but can combine both inductive and deductive approaches (Porta & Keating, 2008). Driven by our research goal and objectives, we follow an *interpretivist* approach and conduct research in line with *inductive reasoning*. Moreover, the research would be driven by qualitative research methods, to some extent incorporating quantitative data collection techniques, when required. The research strategy is presented in the following section.

### 3.2 Research strategy

*Interpretivism* and *inductive reasoning* guide research into building knowledge from empirical studies. *Inductive reasoning* draws inferences from observations in order to make generalizations (Trochim, 2005). The inductive reasoning model consists of four stages: 1) *Observation* aiming at collecting empirical data through observing the phenomena; 2) *Pattern* aiming at classifying the facts and identifying patterns in the collected empirical data; 3) *Tentative hypothesis* aiming at defining a preliminary conclusions or theory; and finally 4) *Theory* aiming at validating the tentative hypothesis and generating theory (Trochim, 2005).

As illustrated in Figure 3.1, the inductive reasoning model was chosen as a base for the research strategy of this thesis. However, since the field of software processes and SPI contains some already defined knowledge, we decided to incorporate literature reviews and studies to the inductive reasoning approach in some stages of our research. The upper part of Figure 3.1 shows the alterations to the inductive reasoning approach that were done during this study. The yellow boxes in the figure represent additional literature studies conducted. The bottom part of Figure 3.1 shows how the modified inductive reasoning approach was incorporated into the research stages.

The research process consisted of four stages: 1) *Pre-Study* during which the research field was explored by means of literature reviews, 2) *Empirical Study* during which the empirical data on the research questions was collected and analyzed, corresponding to the *Observation* and

![Figure 3.1 Relationship between the inductive reasoning approach and the research process](image-url)
Pattern stages of inductive reasoning approach, 3) Creation of tentative hypothesis during which preliminary conclusions were defined, corresponding to the Tentative hypothesis stage of inductive reasoning approach, and finally, 4) Evaluation during which the preliminary conclusions were evaluated and the theory was generated, corresponding to the Theory stage of inductive reasoning approach. The research process and its stages are further described in the following section, Section 3.3.

3.3 Research process

The research process consisted from the four major stages: Pre-Study, Empirical study, Creation of tentative hypothesis and Evaluation. The research process stages are represented in Figure 3.2 and Figure 3.3, and are described below.

3.3.1 Pre-Study

The first stage of the research process, Pre-Study, aimed at defining the research problem and question, and at providing an overview of the state of the art. As it can be seen in Figure 3.3, Pre-study consisted of two parallel research steps: Research field study and Definition of the research focus. As part of the Definition of the research focus step, we first identified the research problem and defined preliminary research goal and objectives. Then, during the Research field study step, we studied the state of the art within SPI and gathered theoretical knowledge, by means of a literature study. During this step, we studied three major source types: books and standards, scientific articles, and white papers or industrial reports using mainly digital databases and libraries, such as IEEEXplore, ACM, Springer, Scopus and Libris, as well as simple internet-based search. We have explored the state of the art on the software development processes, software development methods and methodologies, software method adoptions, software process improvements, as well as software process improvement tools and models. Finally, we came back to the Definition of the research focus step, we confirmed the research goal and objectives, and defined the research strategy to be followed in this research.

The Pre-Study stage resulted in a defined research goal and objectives, and a better understanding of the state of the art of SPI. The literature study presented the theoretical knowledge of software process, as defined in the books, standards and scientific articles, but did not provide comprehensive information on its application in industry as defined in white papers and experience reports. Therefore, we decided to explore the empirical context of software processes and their improvement efforts, in order to understand how software companies used the knowledge from the state of the art, and what elements of real world were missing in the existing theories.

3.3.2 Empirical study

The second and major stage of the research process, Empirical study, aimed at exploring changes and improvements of software processes in industrial contexts. It consisted of two

Figure 3.2 Research process stages
sub-stages Explorative empirical study and Explanatory empirical study, see Figure 3.3. The Explorative empirical study aimed at gaining an overall understanding of the software development process, its formality level and evolution in time by means of three independently conducted case studies. The Explanatory empirical study, on the other hand, aimed at gaining the understanding of the SPI, its context, benefits and challenges, by studying two real-time SPI projects using an action research method. The two sub-stages are presented below.

Explorative empirical study

The Explorative empirical study consisted of three case studies, which are described as three consecutive steps: 1st, 2nd and 3rd case study, respectively. All the case studies concerned software development organizations that have experienced changes/improvements of their processes, which were triggered by the growth of the organization and/or need of structural and defined software processes.

The 1st and 2nd case studies had a similar structure. They concerned the evolution of companies’ software processes impacted by companies’ expansion, and aimed at identifying
factors that might contribute to process improvement models. During the 1st case study, we studied a medium-sized software company that had been continuously growing during the last seven years. The subject of the 2nd case study, on the other hand, was a product development group that started as a small and independent start-up company, underwent two acquisitions and ended up as part of a multinational software corporation during its eleven-year history.

The 1st case study was conducted by means of participatory observations of the company's software process, methods, and practices. The observations were complemented by a number of unstructured interviews and reviews of the documentation. After it, six employees of the company were interviewed using semi-structured open-ended interviews. The 2nd case study, on the other hand, was conducted by interviewing five company employees via similar semi-structured and open-ended interviews. The interviewees chosen for the interviews in both case studies were senior company employees, who had deep knowledge of software processes and process evolution. The semi-structured interviews in both case studies were based on the questionnaire presented in Appendix A.1.

The 1st and 2nd case studies illustrated the impact of business growth on the software processes, and highlighted the importance of planning and preparing for process changes in advance, as well as of continuous reviewing, changing, and evolving software processes over time. The studies also highlighted the importance of social and organizational aspects on the results of process change. The results of the 1st case study are published in Paper 1 (Nikitina & Kajko-Mattsson, 2010b) and the results of the 2nd case study are published in Paper 2 (Nikitina & Kajko-Mattsson, 2010c).

The 3rd case study concerned studying two software method adoption initiatives and its results in one organization. The organization studied was a small software development company that underwent two software method adoptions. As a result of the adoptions, the organization changed and improved its software processes. We studied the organization by means of participatory observations of its development processes during the period of eight months. The observations were complemented by reviews of documentation and unstructured interviews with the company's employees. After it, we conducted structured interviews with all the company's employees. The interviews were based on the questionnaire presented in Appendix A.2.

By comparing the two adoption initiatives and analyzing their results, the 3rd case study illustrated challenges of SPI. Most of the identified challenges were organization and people related, highlighting the importance of organizational, social, and managerial aspects of SPI. This made us to redefine the research goals and objectives in order to focus them on organizational, social, and managerial factors influencing SPI. The results of this study are published in Paper 3 (Nikitina & Kajko-Mattsson, 2009).

Explanatory empirical study

The Explanatory empirical study consisted of two action research studies, which are described as two consecutive steps: 1st and 2nd action research study, respectively. The action research studies aimed at studying one software development organization that had adopted a new software development method, Scrumban, in its two geographically distributed offices. The
goal of the action research studies was to gain full understanding of software method adoption processes at the studied organization, and gather knowledge on how it should be conducted.

The 1st action research study concerned a software method adoption from Scrum to Scrumban in a big-bang manner. The adoption was conducted in the company’s development office in Sweden. After the success of the first adoption initiative, the company decided to transfer to Scrumban in its second development office, in Vietnam. The second method adoption initiative was conducted in an iterative manner, and it was this initiative that was studied during the 2nd action research study.

During the 1st and 2nd action research studies, both the researchers and the company were engaged in the intended study in action. While the company was undergoing software method adoptions and process improvements, the researchers were actively observing and participating in it, by means of participatory observations. As participant observers, the researchers observed the process during all its stages, participated in all its activities, identified problems and guided and reflected upon the process changes. The participatory observations were complimented by unstructured interviews, workshops with the employees, reviews of the system and process documentation. Other data collection methods, used during this sub-stage, were surveys used for evaluating the employee satisfaction, and semi-structured interviews. The surveys were used several times during the process to assess and compare the differences in employee satisfaction over time. About forty company employees have filled in the surveys, including all the software developers, testers, development managers and product owners. The survey is presented in Appendix A.3. During the semi-structured interviews, altogether we interviewed forty company employees aiming at gathering data for process assessment. Most of the interviews were based on the questionnaire presented in Appendix A.4.

During the 1st and 2nd action research studies, we gained a thorough understanding of how both big-bang and iterative software method adoption initiatives were conducted, what challenges and obstacles were met during them and what conditions might have contributed to a successful software method adoption. The results of the 1st action research study are published in Paper 4 (Nikitina & Kajko-Mattsson, 2011a), whereas, the results of the 2nd action research study are published in Paper 5 (Nikitina, et al., 2012).

### 3.3.3 Creation of tentative hypothesis

During the third stage of the research process, Creation of tentative hypothesis, we aimed at analyzing knowledge gathered during the Empirical study stage, complimenting it with additional knowledge gathered from wider contexts, and finally, building preliminary conclusions for this study. The Creation of tentative hypothesis stage consisted of two sub-stages: Creation of the SPI checklist and Creation of the SMA model. The Creation of the SPI checklist sub-stage concerned creating the preliminary checklist for evaluating the SPI effort. This checklist should contain the success factors and best practices of the SPI. The Creation of the SMA model sub-stage concerned creation of a preliminary model for Software Method Adoption (SMA). The two sub-stages and their steps are presented below.
Creation of the SPI checklist

The Creation of the SPI checklist sub-stage started with a research step called, Elicitation of SPI success factors. During this step, we aimed at creating a basis for identifying factors that contributed to successful and sustainable SPI which, in turn, would aid software companies in defining, planning, monitoring and improving their SPI efforts, and in sustaining their results. The factors were gathered by conducting two independent studies: literature study and interviews.

During the literature study of the Elicitation of SPI success factors step, we reviewed more than forty five publications dealing with SPI projects. These were mainly experience reports and case studies that had been retrieved from IEEE, ACM, Springer, John Wiley and Sons, and other publishers. Out of them, we chose twenty seven empirical reports describing conditions contributing to or subtracting from the success of SPI projects. During the literature study, we identified twenty seven SPI success factors. The factors and their literature sources are listed in Appendix B.

During the interviews of the Elicitation of SPI success factors step, we interviewed forty five software engineers who had been involved in or who had been affected by SPI projects. They came from twelve different medium-sized software organizations. The interviews were based on the questionnaire presented in Appendix A.5. The SPI success factors listed by the interviewees were analyzed and grouped together when concerning the same or similar issues.

As a result of the Elicitation of SPI success factors step, we outlined thirty three SPI success and sustainability factors. The results of this step are published in Paper 6 (Nikitina & Kajko-Mattsson, 2012b). The elicited SPI success and sustainability factors were used in the latter stages of this research as a base to identify organizational, social and managerial factors facilitating SPI effort.

After the SPI success factors were elicited, we used the knowledge collected in them to create a preliminary SPI checklist. The preliminary SPI checklist aimed at listing the conditions necessary for succeeding with the SPI implementations and sustaining their results. It was done in the next research step called Creation of the preliminary SPI checklist.

The Creation of the preliminary SPI checklist started by restructuring the already gathered SPI success factors into the conditions necessary for succeeding with the SPI. Those conditions represented the properties (attributes) of healthy SPI efforts. Next, we created an interview questionnaire that aimed at investigating the significance and completeness of the preliminary SPI checklist. We first conducted two pilot interviews, during which we tested the clarity of the questionnaire and completeness of the checklist items. This resulted in the improved questionnaire and some minor changes to the checklist. The questionnaire is presented in Appendix A.5. Based on the pilot tested questionnaire, we conducted interviews, during which we interviewed seventeen software engineers that were involved in SPI or were affected by it. We interviewed ten engineers and delegated interviews with seven other engineers to students on a master level. Finally, based on the feedback provided in the interviews we added five new checklist items and reformulated several existing ones.
This research step resulted in the creation of a preliminary SPI checklist that includes items indicating the properties (attributes) of healthy SPI efforts. During this study, we confirmed the importance and usefulness of the preliminary SPI checklist within the organizations that had been involved in SPI. The results of this step are published in Paper 7 (Nikitina & Kajko-Mattsson, 2012c).

Creation of the initial SMA model

The second sub-stage, Creation of the initial SMA model, consisted of just one research step. During this step, we aimed at creating a process model to be used for adopting one software development method or transitioning from one method to another. The process model is called a model of Software Method Adoption (SMA).

In order to create the initial SMA model, we studied and compared two real-life software method adoptions that we had been studying using action research during the Explanatory empirical study sub-stage. The research method used during this study was a comparative study. We compared the process models of two adoptions using the descriptive comparison method (Sapsford & Jupp, 2006). Here, we used the following comparison criteria (1) Purpose focusing on the general overview and purpose of the method adoptions, (2) Culture comparing the cultural contexts of the two adoptions, (3) Roles focusing on the roles driving the method adoptions, (4) Attitude presenting the stakeholders’ attitude towards the method adoptions, (5) Strategy and design concentrating on the change strategy, and (6) Lessons learned listing the experiences gained during the method adoptions.

After we compared the two software method adoption processes, we looked for properties such as similarities, dissimilarities, inconsistencies, duplications and redundancies which we then analyzed and attended to. We first analyzed the process activities in their specific contexts, explained them from the perspective of their domestic processes, and, based on that, we then determined what process activities and in what order should be included within the software method adoption process model. After it, we compiled the collected and analyzed data and created an initial SMA process model. We also analyzed the data on the specific contexts of the domestic processes and outlined an initial list of contextual factors that impact the design of a software process change. This list could be used together with SMA process model in order to support its implementation in different organizational contexts.

This sub-stage resulted in the initial SMA model, which consisted of taxonomies of process transition activities and a process frame including containers to be filled in with the taxonomy activities relevant for the process context at hand. The initial SMA model is published in Paper 9 (Nikitina & Kajko-Mattsson, 2012a).

3.3.4 Evaluation

The final stage of this research, Evaluation, concerned adaptation and evaluation of the preliminary results created during the Creation of tentative hypothesis stage, to the industrial context. The Evaluation stage started with the SPI checklist evaluation and SMA model evaluation sub-stages, and it rounded up by generalizing research findings in final sub-stage, called Creation of SPIF.
SPI checklist evaluation

The SPI checklist evaluation sub-stage aimed at enhancing the preliminary SPI checklist and at evaluating the usefulness of this checklist in the industrial context. The sub-stage consisted of two consecutive steps: SPI checklist adaptation and SPI checklist evaluation.

The SPI checklist adaptation step started by examining the relevancy of the checklist items and by searching for all types of deficiencies such as duplications, omissions and redundancies. After it, we concluded that the initial checklist, created during the Creation of the SPI checklist sub-stage, was not complete and exhaustive enough. We decided to improve the checklist and adapt it to the industrial context by conducting a literature study and an industrial checklist review.

During the literature study, we studied books, standards, scientific articles and experience reports during which we focused on the issues that were not covered by the initial checklist, such as activities and best practices whose presence was imperative for making SPI projects successful and sustainable. During the industrial checklist review, we conducted a thorough examination of the checklist together with an SPI manager at Rolls Royce, in order to: 1) identify missing checklist items, 2) identify checklist items of low criticality and importance, and 3) clarify the formulation of each item. We graded all the checklist items according to their impact on the potential SPI success, as Low, Medium or High. The lowly graded items were discussed again, and were either upgraded to medium or were removed from the checklist. After the lowly graded checklist items were removed, we went through the checklist again in order to: 1) identify any other missing items, and 2) correct the formulation of any other item.

The second step of this sub-stage, called SPI checklist evaluation, concerned the evaluation of the usefulness and validity of the checklist on ten finalized SPI projects in Rolls Royce. After selecting the SPI projects for the evaluation, we interviewed the SPI manager responsible for those projects, in order to gather full understanding of the selected projects. The interviews were based on the questionnaire presented in Appendix A.7. Then, we evaluated the status of those projects by using the checklist, following the survey presented in Appendix A.8. The project evaluation implied matching each checklist item against each SPI project status and ticking it off, if it got fulfilled.

When analyzing the qualitative data, we identified missing checklist items and reformulated some of the existing ones. During the analysis of the quantitative data, we first determined the significance of the checklist items to the SPI projects’ success. We did it by identifying the correlation between the fulfillment of a checklist item by all the evaluated SPI projects and the success rates of those projects. We also defined weights for small and large SPI projects, limiting the sample to the small or large SPI projects, respectively.

Last, we defined SPI project scores for each evaluated SPI project. Here, we summed the weights of the checklist items that were fulfilled by the SPI project. Then, we compared the SPI project scores to the success rates of those projects, in order to determine the strength of the relationship between the two variables. The relationship was measured using the coefficient of correlation and the coefficient of determination.
As a result of the SPI checklist evaluation sub-stage, we improved the previously outlined SPI checklist and evaluated its usefulness and applicability in the industrial context. The study showed that the success of SPI projects was no accident but a foreseeable outcome of assessable characteristics that had been put into the SPI checklist. The checklist evaluated in this step constitutes one of the outcomes of this thesis, as presented in Section 5.6. The results of this study are published in Paper 8 (Nikitina, et al., 2014).

SMA model evaluation

The SMA model evaluation sub-stage aimed at improving and evaluating the initial SMA model and the initial list of contextual factors by means of literature review, and by mapping them to the industrial method adoption projects. The sub-stage consisted of two consecutive steps: SMA model enhancement and industrial evaluation of the SMA model.

The SMA model enhancement step started by reviewing the results of the Empirical study stage and the initial SMA model. This review pointed out that the model was very abstract in its original form. It only outlined the practices and their inherent activities that were critical for adopting methods. It did not suggest any process phases or any placements of the activities within the phases. For this reason, we structured the SMA model into stages, phases, placed activities within them.

The initial SMA model and the initial list of contextual factors were first evaluated against current state of the art, which was conducted in form of a literature study. Here, we studied various literary works such as books, standards, experience reports and other related articles describing, guiding and reporting on process adoptions and their successes and failures. As a result, we identified some new adoption activities, best practices, lessons learned and guidelines which we then used for enhancing the initial SMA model. We also identified additional contextual factors that impacted the process improvement projects. When enhancing the SMA model and the list of contextual factors with our new findings, we strongly considered their applicability and potential benefits for optimizing method adoptions. Finally, we revised the SMA model’s structure to make sure that it was still coherent, sound and reasonable.

The second step of this sub-stage, called industrial evaluation of the SMA model, concerned the evaluation of the SMA model against six industrial method adoption projects from three organizations. The SMA model was evaluated by studying the conformance of the SMA model to the industrial method adoption projects. The list of contextual factors was evaluated by conducting a review of the factors by the industrial representatives. We interviewed three representatives altogether, one representative per company. The representatives were leading or were actively involved in the selected method adoption projects. The evaluation was based on the questionnaire that is presented in Appendix A.9. Using the results of the interviews, we first evaluated whether the SMA model conformed to the industrial method adoption processes. We then attended to the deficiencies in the SMA model and somewhat enhanced it. With help of our interviewees, we also confirmed the impact of the contextual factors on the process change.

As a result of the SMA model evaluation sub-stage, we confirmed that: 1) the SMA model correctly reflected the state of the current software method adoption practice, and 2) the listed
contextual factors had significant impact on the process change processes in the studied method adoption projects. The SMA model and the contextual factors impacting process change constitute two of the main outcomes of this thesis, as presented in Section 5.4 and Section 5.5. The results of this study are published in Paper 10 (Nikitina & Kajko-Mattsson, 2014).

Creation of SPIF
The final sub-stage of this research, Creation of SPIF, aimed at rounding up the research findings and at creating the Software Process Improvement Framework (SPIF), in order to show how the results of this research contributed to the existing body of knowledge on SPI.

The sub-stage started by studying all the papers written during the research process, reviewing the results of the research and the preliminary conclusions, as well as comparing the research results to the existing state of the art within SPI. During this step, we also reviewed and adjusted the research goals and objectives of this study, focusing them on the impact of organizational, social and managerial aspects of SPI.

The second step of this sub-stage concerned creation the SPIF framework. The outlined framework provided a general overview of the complexity and diversity of SPI, presented existing SPI approaches and models, and incorporated the theories of organizational change. The framework was based the existing framework for SPI environment proposed by Sami Zahran (1998). Since SPIF was created in the final stage of this study in order to position the other research results in relation to the existing theories in SPI, it was not published in the papers attached to this thesis. The framework is however the main contribution of this thesis, and is presented in detail in Section 5.2.

Last, during this sub-stage, we generalized the results of this research and positioned them in the SPIF framework. Here, we also reviewed the thirty three SPI success and sustainability factors that were published in Paper 6, and selected the factors that relate to organizational, social and managerial aspects of SPI. As a result, we selected thirty one factors, called organizational, social and managerial factors facilitating SPI effort. Those factors constitute one of the results of this thesis, and are presented in detail in Section 5.3.

3.4 Research method
The major stage of this research, Empirical study, lasted for the period of three and a half years. During this time, we studied software processes and SPI in its real settings. The Empirical study aimed at investigating contexts and factors around SPI from the participants’ perspective. Therefore, qualitative research methods, such as action research and case study research, were used as the main research methods.

Action research and case study balance practical and academic applicability of the potential research results, since they aim at creating a theory or a method based on practice and purely empirical findings and solutions. The results of the action research and case study should be able to benefit largely both (1) practitioners since real life problems are addressed and (2) researchers since the detailed access to the results from the industrial cases is provided. Therefore, the use of those research methods would fit to both the academic and industrial target audiences of this research.
Case study allowed us to gain initial understanding of software process changes and improvements in the industry, based on the studies of process evolution over time. Action research, on the other hand, allowed us to observe the software process change and improvements process when actively participating in it in real time. This provided us a deep understanding of the SPI process, its context and factors that impacted it.

3.4.1 Case study

Case study is a qualitative research method that uses empirical evidence and data from people in real life context (Myers, 2009). Unlike action research, during the case study the information can be gathered concerning either real time or retroactive events, since the researcher does not participate in the event but instead only collects information to describe it (Yin, 2003). Therefore, we used case study method to study retroactive events of process change, since those were not possible to study using action research.

Case studies are common to use for exploring the phenomena of study and for discovering relevant factors and characteristics that might apply in similar conditions/situations (Myers, 2009). Therefore, case studies are the most appropriate in the early stages of the research for exploratory purposes (Myers, 2009). In line with this thought, we used case study research at the beginning of the Empirical study stage, when gathering explorative knowledge about the software process, its evolution and improvements.

3.4.2 Action research

Action research is a qualitative research method that focuses on research in action, and according to its definition, it is applied to “solve real organizational problems” (Coghlan & Brannik, 2005, p. 3). Action researchers do not only observe the phenomena in its natural context, but also participate actively on solving the problems related to the phenomena in real time (Coghlan & Brannik, 2005). This means that a real life case study is being written as a change in the process is occurring (Alexandre, et al., 2006). The direct participation in the SPI process that was enabled by action research, directly contributed to the research goal and objectives of this study, since it aided to the understanding of organizational, social and managerial aspects of SPI. Due to the nature and depth of the action research method, we used it at the end of the Empirical study stage, when gathering explanatory knowledge on the software process and its improvements.

3.5 Data collection and analysis

Different instruments and approaches for data collection and analysis were used during the course of this research. Data collection instruments and approaches are presented in Section 3.5.1, followed by the description of the data analysis approaches in Section 3.5.2. Finally, sampling methods used during this research are explained in Section 3.5.3.
3.5.1 Data collection approaches

This research consisted of literature studies and empirical studies. The literature studies implied structured studies/reviews of the published books, journals, articles, and experience reports on the topic of interest. The empirical studies, on the other hand, involved a number of qualitative data collection approaches, such as interviews and participatory observations, and incorporated few quantitative data collection instruments, such as surveys. The qualitative data collection approaches supported the research goal and problem of this study, which required having deep understanding of the phenomena, which was SPI process. Characteristics of the SPI process, its benefits and challenges in most cases cannot be counted or measured in numbers but instead should be expressed in words. Therefore, this research was mostly based on qualitative data.

In order to ensure the reliability of the collected empirical data, we implemented triangulation, by using a variety of data collection sources and approaches, including quantitative data gathering approaches, when appropriate. The use of different data gathering approaches, such as 1) interviews, 2) participatory observations, 3) studies of the company documentation, and 4) surveys, facilitated to build up a complete picture of the situation, activities or context. Moreover, in order to ensure reliability of the collected data, we tried to incorporate opinions and knowledge of people that held different roles, encompassed different expertise, and were involved in or were affected by SPI to different extent.

Structured, semi-structured and un-structured interviews were used in different stages of this research, in order to gain understanding of the SPI process, of the people involved in SPI, and of the factors that conditioned SPI success. The interviews would help to understand the insider view of the situation, environment and personal positions. Structured and semi-structured interviews were based on the carefully designed questionnaires, as presented in Appendix A. Unstructured interviews, on the other hand, complemented an overall understanding of the phenomena or its context.

Observations were used in order to collect rich qualitative and quantitative data about the SPI process and people involved in it from the insider perspective. Participatory observations often implied that the researcher joined software development team and participated in the process improvements and process changes processes. Participatory observation is defined as “the immersion of the researcher into the social setting for an extended period in order to observe, question, listen and experience the situation in order to gain an understanding of processes and meanings”. (Walliman, 2006).

To support the data collected during the participatory observation and interviews, the researchers also studied various company documentations. Studying documentation such as change management tool, software development process documentation, requirements and test specifications, and dynamic information boards contributed to gaining knowledge about the company, its processes and problems.

Another data collection method used during this research was survey. Surveys were used in order to collect structured data that aimed at evaluating employee satisfaction with the software process. The survey presented in Appendix A.3 were filled up twice by
approximately forty company employees including the software developers, testers, development managers and product owners.

3.5.2 Data analysis approaches

The qualitative data gathered during the research process were analyzed using hermeneutics approach. Hermeneutics approach focuses on the meaning of the qualitative textual data, analyzing what people say and do, and why (Myers, 2009). Therefore, using this approach allowed us to analyze and understand the actual meaning of the qualitative data and to analyze texts, considering the context and cultural issues.

The quantitative data gathered during this research were analyzed using statistical analysis technics. Statistical data was represented using variety of presenting formats, such as bar graph, pie chart, line graph and structured data in a table format. During one of the final research process steps, SPI checklist evaluation, we introduced a number of statistical variables for analyzing the data. The results of this research process step are presented in Appendix D.

3.5.3 Sampling

During the different stages of the research process, we used different sampling methods for selecting the subjects of study, such as software development organizations or software engineers.

First, when selecting the three organizations for case study research during the Explorative empirical study sub-stage, we followed the following sampling criteria: 1) the organization should have experienced software process improvement and/or process change; 2) the organization should be willing to take part of the study and share the information about the process change; and 3) the data on the software process should be accessible, and the personnel should be available for the interviews. Due to low availability of the organizations, we could not use any other method but purposive sampling for selecting the organizations to take part of this study. However, when selecting the sample of employees to be interviewed, we followed the conditional sampling method. This method was used because we needed to select employees that satisfied the required criteria. The criteria were the following: 1) employees that have worked in the company the longest time and therefore have historical information, 2) employees that are involved in the software process and therefore have information about it, 3) employees that represent both management and development levels.

Second, during the Explanatory empirical study sub-stage, we selected the organization for the action research study using purposive sampling method. This method was chosen due to lack of availability of the organizations that could satisfy the sampling criteria. The sampling criteria, for choosing the organization for the action research study was the following: 1) the organization should conduct software process improvement during the time when it was being studied; 2) the organization should be willing to take part in research and share the information about the process change; 3) the organization should allow the researcher to participate and observe the cause of events in the organization, and study the company documentation; and 4) the organization should allow the researcher to interview the personnel. On the other hand, when selecting the employees to be interviewed, we selected all
the employees that had influenced or had been affected by the SPI process. Therefore, we followed conditional sampling method to select the interviewees.

Third, during the Creation of the SPI checklist sub-stage, we selected forty five software engineers who had been involved in or who had been affected by SPI projects. Those engineers were selected using purposive sampling method. Among the interviewees, there were twenty seven software developers, ten testers, seven development managers and one SPI manager. They were interviewed in order to identify the factors that contributed or subtracted from the success of the SPI project. Later, during the Creation of the SPI checklist sub-stage, we selected and interviewed another seventeen software engineers that were involved in SPI or were affected by it. Here, we again used the purposive sampling method. The selected interviewees came from fifteen software companies involving six engineers, six technical managers, three SPI managers and two consultants. The interviews concerned evaluation of the significance of the identified earlier SPI checklist’s items.

Fourth, during the SPI checklist evaluation sub-stage, we selected an organization that had a large number of historical and ongoing SPI projects. Therefore, we used the purposive sampling method for selecting the organization. Then, we defined the sampling criteria for selecting the SPI projects, which was as follows: 1) the SPI projects should be finalized, 2) the SPI projects should be of different sizes, 3) the SPI project data should be accessible, 4) the people involved in the SPI projects should be available, 5) the success rate of the SPI projects should be known, and finally, 6) the success rate of SPI projects should be of varying degree. In order to satisfy the selection criteria, we could not choose any other method than the non-probability quota sampling method. The required sampling quota was as follows: 1) 50% of the selected SPI projects should be large SPI projects and other 50% should be small SPI projects, and 2) the selected SPI projects should represent unsuccessful, partly successful and successful SPI projects.

Last, during the SMA model evaluation sub-stage, we selected the method adoption projects for the evaluation using conditional sampling method. The sampling criteria for selecting the projects were the following: 1) the projects should have performed adoptions of new development methods, 2) the projects, if not finalized, should be at least in the final stage of the adoption process, and 3) the company representatives to be interviewed should be knowledgeable about the method adoption. Due to the difficulty to find the projects that would meet the defined sampling criteria, we could not use any other sampling method than purposive sampling. The interviewees were also selected based on the purposive sampling method, since we needed to interview people who were leading or were actively involved in the selected method adoption projects, and were willing to take part in the interviews.

3.6 Organizations involved

Thirty software development organizations were involved in this research. Their level of involvement differed depending on the stage in which they were involved and the purpose of the study. The following three sub-sections present the organizations involved in the Empirical study stage, the Creation of tentative hypothesis stage and the Evaluation stage.
3.6.1 Organizations involved in the Empirical study stage

In order to gather empirical data on the software process, its evolution and change, we studied three software development organizations in the Empirical study stage. Due to the sensitivity of the results presented here-in, all the three organizations chose to stay incognito. Therefore, we use fictitious names of the companies instead. We refer to the organizations involved in this stage as Mobile Navigation, VCP group and CMSiPro. All the three organizations were involved in the Explorative empirical study sub-stage as case studies. During the Explanatory empirical study sub-stage, however, we studied only CMSiPro using action research. The short descriptions of the organizations are presented above.

- **Mobile Navigation** was a small Swedish start-up company. At the time of the study, it had twelve employees and it evolved and maintained a software product for mobile phones. This product was a freeware available via Internet for unlimited amount of users.

- **Virtual Computing Product group (VCP group)** was a small product development group that underwent two major acquisitions. Prior to the acquisitions, the VCP group was the only development team of a small Swedish company called Virtual Software Group (VSG). The company developed a virtual platform product. In 2002 VSG was first acquired by an organization called Software Infrastructure Group (SIG), an application infrastructure software company. After it, the VCP group became one of SIG’s development teams and continued to develop the same software product. In 2008 SIG was acquired by Enterprise Software Provider (ESP), a world leading multinational software corporation. As a result, the VCP group became a development group within ESP where it is still working on the same product line. At the time of the study, VCP group had seventy five software engineers.

- **CMSiPro** is a Swedish software company that develops and maintains a content management system. Its headquarters are in Stockholm, Sweden. At the time of the study, the company had sales and marketing offices in ten countries all over the world, and company’s software development was geographically distributed between two development offices, one in Stockholm, Sweden and another one in Hanoi, Vietnam. The company’s technical staff included one team of developers located in Sweden, one team of developers and one team of testers located in Vietnam. Each team consisted of fifteen to twenty engineers.

3.6.2 Organizations involved in the Creation of tentative hypothesis stage

During the Creation of tentative hypothesis stage, in order to elicit initial SPI success factors we interviewed forty five software engineers. Among the interviewees, there were twenty seven software developers, ten testers, seven development managers and one SPI manager. They came from twelve different software organizations that were located in five different countries around the world. The profile of the organizations involved is presented in Table 3.1.
Table 3.1  Profile of twelve organizations involved in elicitation of SPI success factors

<table>
<thead>
<tr>
<th>No</th>
<th>No of employees</th>
<th>Domain</th>
<th>Headquarters</th>
<th>Roles of the interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;100</td>
<td>Software development</td>
<td>China</td>
<td>Project manager</td>
</tr>
<tr>
<td>2</td>
<td>&lt;50</td>
<td>Software service provider</td>
<td>Sweden</td>
<td>Software developer</td>
</tr>
<tr>
<td>3</td>
<td>&lt;500</td>
<td>Banking Software, ERP software</td>
<td>Bangladesh</td>
<td>Process manager</td>
</tr>
<tr>
<td>4</td>
<td>&lt;500</td>
<td>Online games</td>
<td>Iceland</td>
<td>Development manager</td>
</tr>
<tr>
<td>5</td>
<td>&gt;10 000</td>
<td>Information Technology and Services</td>
<td>Sweden</td>
<td>Development manager</td>
</tr>
<tr>
<td>6</td>
<td>&gt;10 000</td>
<td>Telecommunications</td>
<td>Sweden</td>
<td>15 developers, SPI manager</td>
</tr>
<tr>
<td>7</td>
<td>&gt;10 000</td>
<td>Electronic Manufacturing</td>
<td>Bangladesh</td>
<td>Test engineer</td>
</tr>
<tr>
<td>8</td>
<td>&lt;200</td>
<td>Content management system</td>
<td>Vietnam</td>
<td>Process manager, 9 testers, 9 developers</td>
</tr>
<tr>
<td>9</td>
<td>&lt;200</td>
<td>Web portal</td>
<td>Sweden</td>
<td>Software developer</td>
</tr>
<tr>
<td>10</td>
<td>&lt;100</td>
<td>Media systems</td>
<td>Sweden</td>
<td>IT manager</td>
</tr>
<tr>
<td>11</td>
<td>&lt;100</td>
<td>Social services</td>
<td>Sweden</td>
<td>IT manager</td>
</tr>
<tr>
<td>12</td>
<td>&lt;100</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>IT architect</td>
</tr>
</tbody>
</table>

Table 3.2  Profile of fifteen organizations involved in evaluation of preliminary SPI checklist

<table>
<thead>
<tr>
<th>No</th>
<th>No of employees</th>
<th>Domain</th>
<th>Headquarters</th>
<th>Roles of the interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;5000</td>
<td>Telecommunications</td>
<td>Sweden</td>
<td>Technical manager</td>
</tr>
<tr>
<td>2</td>
<td>&lt;500</td>
<td>Telecommunications</td>
<td>Sweden</td>
<td>Configuration manager</td>
</tr>
<tr>
<td>3</td>
<td>&lt;50</td>
<td>Business Solutions</td>
<td>US</td>
<td>Service operation</td>
</tr>
<tr>
<td>4</td>
<td>&gt;10 000</td>
<td>Oil and energy</td>
<td>Sweden</td>
<td>Technical manager, Process manager</td>
</tr>
<tr>
<td>5</td>
<td>&lt;1 000</td>
<td>Transportation</td>
<td>Sweden</td>
<td>System engineer</td>
</tr>
<tr>
<td>6</td>
<td>&lt;50</td>
<td>Software service provider</td>
<td>Sweden</td>
<td>Developer</td>
</tr>
<tr>
<td>7</td>
<td>&gt;10 000</td>
<td>Aviation and aerospace</td>
<td>England</td>
<td>SPI manager</td>
</tr>
<tr>
<td>8</td>
<td>&lt;50</td>
<td>Software development</td>
<td>Sweden</td>
<td>Technical manager</td>
</tr>
<tr>
<td>9</td>
<td>&lt;100</td>
<td>Online gaming</td>
<td>Sweden</td>
<td>Software engineer</td>
</tr>
<tr>
<td>10</td>
<td>&lt;100</td>
<td>IT consulting, Management consulting</td>
<td>Sweden</td>
<td>SPI manager, IT consultant</td>
</tr>
<tr>
<td>11</td>
<td>&gt;10 000</td>
<td>IT consulting</td>
<td>Portugal</td>
<td>IT consultant</td>
</tr>
</tbody>
</table>

The significance of the preliminary SPI checklist was investigated by seventeen software engineers that were involved in SPI or were affected by it. Altogether, those interviewees came from sixteen software companies involving six engineers, six technical managers, three SPI managers and two consultants. The profile of the organizations involved in this step is presented in Table 3.2.

3.6.3 Organizations involved in the Evaluation stage

The evaluation of the SPI checklist was done in cooperation with Rolls Royce, during which we applied and evaluated the checklist within ten finalized SPI projects at Rolls Royce.

Rolls Royce is a world-leading provider of power systems and services. The company has a century-long history and reputation, and a strong position in global markets of civil aerospace, defense aerospace, marine and energy. The company’s headquarters are located in...
London, England, and its operations are spread around North America, Europe, Asia and Middle East. Right now, Rolls Royce employs more than 40,000 people. Rolls Royce manages a wide variety of system development projects on a daily basis, and invests over £900m in research and development. Since their systems are mainly safety critical, their system development processes are well-defined and strictly followed. Their projects are always in compliance with a range of process and quality standards, such as RTCA DO-178B for Aerospace projects (RTCA DO-178B, 1992). (Rolls Royce, 2013)

The evaluation of the SMA model was done in collaboration with three large organizations. Due to the sensitivity of the results presented herein, two of the organizations chose to stay incognito. Therefore, we use fictitious names of those companies instead, referring to them as Engineering group and Telecommunication group, respectively. All three organizations had a solid experience in conducting process changes related to software method adoptions, with respect to which the SMA model was evaluated. The descriptions of the organizations are below.

- **Engineering group** is a leading engineering organization, based in London, England. The company manages a wide variety of system development projects on a daily basis, and its operations are spread world-wide.
- **Telecommunication group** is a world-leading provider of telecommunications equipment and services. The company’s headquarters are located in Stockholm, Sweden, and its operations are spread world-wide.
- **CCP Games** is one of the leading companies in the multiplayer games industry. The company has grown and changed tremendously since the year 2000. At the moment of writing this thesis, the products of this organization had more than 300,000 subscribers, and were well known all over the world. The company’s headquarters are located in Reykjavik, Iceland, and its operations are also distributed to USA, UK and China. CCP Games employs more than 600 people. (CCP Games, 2013)

### 3.7 Validity

All the qualitative research methods, including case studies and action research, encounter validity threats (Myers, 2009) (Coghlan & Brannik, 2005). The validity threats to our study concern risk of using wrong measures, risk of subjective judgment, risk of misinterpreting the data, and risk of context dependency of final results. The validity threats relevant to this study concern construct validity, internal validity, conclusion validity, external validity and reliability.

**Construct validity** refers to the degree to which inference can be made from the operational definition of a variable to the theoretical constructs (Trochim, 2005). The main threat to construct validity is to guarantee that the right measures have been chosen for the study.

Regarding the construct validity, the risk was that the researchers might use wrong measures, and as a result, they might misinterpret the phenomena of study, its impact and its results. The quantitative measurements were used only during the Creation of preliminary SPI checklist step and SPI checklist evaluation sub-stage, when the significance of checklist items was measured, as well as the SPI checklist was tested on the SPI projects. In order to avoid
misunderstanding of the measured data and the concepts used in the SPI checklist, we created
detailed definitions of the grading criteria and definitions of the terminology, which were
communicated to the interviewees. This enabled common understanding of the concepts used
in the SPI checklist and their measurements. Regarding the qualitative measurements that were
used in all the remaining research steps, during them we also provided detailed definitions of
the terminology used to the interviewees or stakeholders involved, in order to enable
common understanding of the concepts.

Internal validity refers to the degree of inferences of the cause-effect or causal relationships in
the study (Trochim, 2005). Even though internal validity is not relevant in most observational
or descriptive studies (Trochim, 2005), it was still highly relevant to this study. Main threats to
internal validity here concerned the social intervention, for instance, some of the observed
individuals could behave differently when being observed. During the case and action
research studies, those threads were minimized by means of triangulations of data, use of
different methods and involvement of various roles. In addition, the stakeholders and the
organizations studied had opportunity to review the results of the interviews and
observations and provide feedback on our understanding of the described events and
opinions. However, due to the high involvement of the researchers in the case and action
research studies, we cannot claim full objectivity of the results and conclusions of those
studies.

Conclusion validity refers to the degree to which the conclusions are based on the correct
interpretation of the relationships of the data (Trochim, 2005). The conclusion validity threat
to our study was that the conclusions and the outcomes of this study would not be related to
the data collected during the research. Regarding the organizational, social and managerial factors
facilitating SPI effort and the SPI checklist, its content was entirely based on the data collected
from identifying the SPI success factors, mandatory SPI activities, and possible causes of the
problems or failures with the SPI, both in literature and via interviews. The names of the
factors/checklist items, their content and clustering however could have been influenced by
our own interpretation and experience. Therefore, in order to minimize the conclusion
validity threats, we presented the results of this study to the interviewees in order to confirm
correct understanding of their feedback.

Regarding the SMA model, its process activities and their order had a direct relationship to the
mirrored transition processes that were developed during the action research study, and the
industrial method adoption projects used during the Evaluation stage. Finally, regarding the
contextual factors impacting process change, they were collected during the case and action
research studies, when we were analyzing the influence of the individual context.

External validity refers to the degree of whether the sample findings can be generalized
(Trochim, 2005). The main external validity threat to our study was the fact that the outcomes
of this thesis would be not applicable in other organizations. Regarding the organizational,
social and managerial factors facilitating SPI effort and the SPI checklist, they contained body of
knowledge that was gathered using both literature studies and empirical studies,
incorporating the existing and newly discovered knowledge in SPI. Moreover, the
applicability of the SPI checklist was evaluated in the industrial context showing that the
checklist items have positive impact on the success of the SPI. Therefore, the SPI checklist can
be used as a collection of best practices within any SPI project. It is however important to point out that the significance of certain checklist items may vary from organization to organization, and from SPI project to SPI project, thus for achieving its maximum potential the SPI checklist need to be tailored to each company/SPI project before it is being applied.

Regarding the SMA model, its phases and activities were based on data gathered during both literature studies and empirical studies. However, due to lack of previous research and the nature and limitations of the method used for developing the model, there is no formal foundation for data generalization. Still however, since the applicability of the SMA model was tested in the industrial context, we believe that the model can be found useful for software development companies, which plan to adopt software development methods to their software processes. Finally, regarding the contextual factors impacting process change, the importance of the factors depends on the individual organizational specifics and needs. However, since the factors provide just general recommendations, they can be useful for any organization that plan to change their processes.

Reliability refers to the degree of whether the sample findings can be repeatable and consistent (Trochim, 2005). Reliable results and conclusions are the ones that are achieved over and over when the same procedure is repeated. In order to contribute to the reliability of our findings, we thoroughly documented the research steps taken during this study as described in this chapter. Still however, due to the high context dependency of results gathered from case and action research studies, we cannot ensure their repeatability in other contexts. Nevertheless, we aimed to increase reliability of our final results by including data from wider contexts and by evaluating the results in industry. Positive results of the evaluation show that the results of this study are consistent with other similar cases and contexts.
Chapter 4

Research contribution

This chapter gives an overview of the research produced during the course of writing this thesis. Section 4.1 provides a summary of the research presented in the ten research papers attached to this thesis, and presents the papers. Section 4.2 lists three additional research papers that have been written during the course of writing this thesis.

4.1 Summary of research papers

The research results and contributions of this thesis have been presented in ten research papers. All the papers illustrate the research process and research work conducted during the course of writing this thesis. Paper 6, Paper 8 and Paper 10, however, present the final research results, which are described in Chapter 5. The research contributions of the ten papers are summarized in Table 4.1 and the summary of the papers is presented below.

**Paper 1. Impact of Growing Business on Software Processes**


Growing business may have serious consequences on the software process, and thereby, on the organizations’ future business prosperity. Therefore, it should be considered in process improvement and/or process changes strategies. By means of participatory observations and interviews, we conducted a case study by studying a medium-sized software company, referred in this thesis as CMSiPro. Our goal was to understand the company’s software development process and challenges brought into it by business growth.

**Table 4.1  Research contribution of the ten papers attached to this thesis**

<table>
<thead>
<tr>
<th>Number and title of the paper</th>
<th>Main contribution of the paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Impact of growing business on software processes</td>
<td>Studied three case studies of process change</td>
</tr>
<tr>
<td>2 Impact of corporate and organic growth on software development</td>
<td>Collected knowledge on the software process improvement process, and the impact of organizational context and culture on it</td>
</tr>
<tr>
<td>3 Historical perspective of two process transitions</td>
<td>Identified the importance of the organizational, social and managerial aspects on the results of the SPI effort</td>
</tr>
<tr>
<td>4 Developer-driven big-bang process transition from Scrum to Kanban</td>
<td>Studied two software process adoptions in real time</td>
</tr>
<tr>
<td>5 From Scrum to Scrumban: A case study of a process transition</td>
<td>Collected knowledge for guiding software method adoptions</td>
</tr>
<tr>
<td>6 Factors leading to the success and sustainability of software process improvement efforts</td>
<td>Confirmed the importance of context and the organizational, social and managerial aspects on the results of SPI effort</td>
</tr>
<tr>
<td>7 Software process improvement health checklist</td>
<td>Elicited and evaluated SPI success and sustainability factors</td>
</tr>
<tr>
<td>8 Successful process improvement projects are no accidents</td>
<td></td>
</tr>
<tr>
<td>9 Process model for software method transition</td>
<td></td>
</tr>
<tr>
<td>10 Process model for software method adoption</td>
<td></td>
</tr>
</tbody>
</table>
When doing our work, we mapped out the software company’s business growth on the course of its historical events and identified its impact on the company’s software production processes and capabilities. The impact concerned benefits brought, challenges encountered and problems met during the business growth. As a result, the paper lists eight lessons learned from the business growth. The lessons learned are the following: 1) build a product roadmap, 2) relieve developers from customer support tasks as soon as possible, 3) implement a process as early as possible, 4) reflect on and improve the process continuously, 5) keep system documentation consistent with the system state, 6) avoid uneven knowledge distribution, 7) implement a communication pattern, and 8) plan for growth. The most important lesson learned, however, is that although business growth has become a stimulus for starting thinking and improving software processes, the organization lacked guidelines aiding it in and aligning it to business growth. These lessons contributed to evolving the body of knowledge on how to change software processes and how to evolve them over time. This knowledge was incorporated into the process improvements and process change processes in the latter papers.

The paper illustrates the impact of business growth on the software processes, and highlights the importance of planning and preparing for process changes in advance, as well as the importance of continuously reviewing, changing and evolving software processes overtime.

**Paper 2. Impact of Corporate and Organic Growth on Software Development**


In this paper, we explored the effects of a corporate and organic business growth on the software development process over the course of its eleven-year history. Here, we continued the work in *Paper 1* by studying another software development organization that had experienced business growth. Our goal was to study the impact of organic and corporate growth on software development and identify issues that might contribute to software process improvement models.

In this paper we examined an evolution of a small start-up company, referred in this thesis as *VCP group* that with time had become part of a multinational software corporation. Initially, the company had a small product development group. It then underwent substantial expansion and two acquisitions. We studied this company by means of structured interviews, conducting a case study research.

While conducting our study, we mapped out the software company’s corporate and organic growth in the course of its historical events and identified its impact on the company’s software production processes and capabilities. We also listed benefits, challenges and problems brought by the business growth. The paper results in lessons learned as experienced by the company studied. These are the following: 1) create a support team early in the process, 2) implement a communication pattern, 3) focus on product quality from the beginning, 4) implement a uniform development process, 5) reflect and improve the process on the continuous basis, 6) plan for process changes and introduce them one by one, 7) educate developers on newly introduced processes, and 8) involve developers into process changes. The most important lessons are *Lessons 7 and 8* concerning the importance of
developers’ education and their involvement in the process changes. The paper rounds up with the suggestion for incorporating business growth elements into software process improvement models.

During this study, we completed our initial understanding of the impact of business growth on software development processes and identified the elements of business growth to be incorporated in software process improvement approaches.

**Paper 3. Historical Perspective of Two Process Transitions**


Transitions from one software development method to another or adoptions of new methods are a common routine in many software companies today. However, process changes do not always lead to changes for the better. They may lead for the worse or they may not imply any improvements at all. The companies need to have clear guidelines on how to change their processes.

In this paper, we report on the results of two process transitions and problems and benefits encountered before and after each transition. By means of participatory observations and interviews, we conducted a case study during which we studied a small software development company, referred to in this thesis as Mobile Navigation. Our goal was to identify challenges encountered during and potential benefits brought by process changes in order to gain solid understanding of how software processes should be improved or changed and how stakeholders involved should be prepared for it.

While conducting this study, we compared the two transitions and analyzed how the software processes evolved, by studying the evolution of the status of the process related problems. By looking at the problems and benefits as encountered before and after each process transition, we concluded that many of the problems were organization and people related. Unless the organization and people matured with respect to management, self-organization, and discipline, the newly introduced method could not help to improve the process.

During this research, we gained further understanding of how process transitions were conducted, what challenges they encountered, what benefits they could bring and how commitment of stakeholders might impact the results. This understanding contributed to building an in-depth knowledge of software processes improvement process and roles of stakeholders involved in it. It is this knowledge that was incorporated into success factors for software process improvement in the latter papers.

**Paper 4. Developer-driven Big-bang Process Transition from Scrum to Kanban**


---

1. At the moment of writing this paper, the company and the researchers believed that the newly introduced process was Kanban. However as acknowledged during the future research, the introduced process was a combination of Kanban and Scrum principles. It is referred to as Scrumban in Paper 5.
Even if most of the process transitions have been made with the goal of improving software processes, very few reports recognize process transition as an important element of software process improvement. Structured process transition can namely strongly contribute to the improvement of software processes.

While conducting our work, we studied process transition that was conducted in a medium-sized software development company, referred to in this thesis as CMSiPro. The transition was conducted in the main development office, situated in Stockholm, Sweden. The research method used was action research. While the company was undergoing process transition and improvements, we were actively observing the course of actions and took part in collective reflection and learning. Our goal was to gain full understanding of process transition and process improvement processes and gather knowledge on how it should be conducted.

This paper reports on a developer-driven big-bang process transition from Scrum to Kanban, and recognizes it as part of process improvement effort. The contribution of this paper are threefold: (1) a thorough account of the transition process and its results, (2) report on the lessons learned, and (3) an initial model of a process transition. The lessons learned during this study are the following: 1) training and education of all the stakeholders are crucial, 2) once the new method is adapted the process should be continuously tuned and adjusted, 3) continuous process reviews are the driving wheel of process improvements, 4) high participation and involvement of stakeholders in process change has a positive effect on the acceptance of change, and 5) dedicated responsibility for and ownership of software process ensures sustainability of the process improvement results.

During this research, we gathered in depth understanding of how big-bang process transition was conducted, what challenges and obstacles were met during the process transition and what conditions might have contributed to a successful process transition. Moreover, we elicited some knowledge that contributed to an initial design of a process transition process model. This model was further evolved in Paper 9, and improved and evaluated in Paper 10.

**Paper 5. From Scrum to Scrumban: A Case Study of a Process Transition**


Transitioning to another development method is a common routine for many companies. It is however not a risk-free adventure. Process transitions may be challenging and may not always bring the desired and long-lasting results. Awareness of the scale, complexity and impact of transitioning from one method to another may increase the chances to succeed with transitions. In order to contribute to building such awareness, we continued studying process transitions in empirical settings.

In this paper, we studied another process transition in CMSiPro. This time however, the transition was done in the company’s second development office situated in Hanoi, Vietnam. It was initiated as software process improvement effort and realized in form of a process transition from Scrum to Scrumban. Here, we conducted action research during which we actively participated in and observed the process transition and took part in collective
reflection and learning. Our goal was to gain understanding of process transition and process improvement processes, and gather in-deep knowledge on how it should be conducted.

The contribution of this paper is threefold: (1) a thorough account of the iterative transition process as executed in the company, (2) results achieved by the process transition, and (3) the lessons learned from its executing. The lessons learned during this study are the following: 1) prior to initiating the process change the organization's readiness to it needs to be determined, 2) training in software process is an important ingredient of process change, 3) coaching and mentoring in software process aids in achieving an in-depth understanding of process change and reasons behind it, 4) continuous process reviews contribute to the lasting results of process change, 5) stakeholder participation in process change has a positive effect on the acceptance of change, and 6) continuous process improvement and tuning activities should be incorporated into the process transition.

The paper concludes that the positive results achieved by the process transition were not so much thanks to the introduction of Kanban principles, but rather, thanks to introducing an appropriate forum for continuous process reviews and improvements. Introducing the mechanisms for continuous process improvements was, however, not enough. All the stakeholders involved in the process had to be adequately trained, mentored and coached, as well as they had to commit to continuous process improvement. Establishing mechanisms for continuous process improvement and having well-trained and committed personnel could improve the process and bring long lasting and sustainable results, with or without transitioning to or adopting new development methods.

During this research we gained an in-depth understanding of how iterative process transition was conducted, what challenges and obstacles were met during this process transition and what conditions contributed to successful and smooth process transition. This knowledge was used to develop a process transition model in Paper 9 that was further improved and evaluated in Paper 10.

**Paper 6. Factors Leading to the Success and Sustainability of Software Process Improvement Efforts**


Although software process improvement (SPI) may bring immediate positive results, this does not imply that the results will sustain in the long run. In order to succeed with continuous process improvement and sustain its results, organizations need to be aware of what makes their SPI efforts successful or unsuccessful. In this paper, we elicited thirty three factors that primarily contributed to the success and sustainability of SPI efforts. We elicited them via two independently conducted studies: literature study and empirical study. Our goal was to create a basis for identifying factors that contributed to successful and sustainable SPI which, in turn, would aid software companies in defining, planning, monitoring and improving their SPI efforts, and in sustaining their results.

The literature study resulted in twenty seven SPI success factors, out of which twenty four factors overlapped with the factors that had been identified during the empirical study. The
Interviews resulted in thirty SPI sustainability factors, out of which twenty-four overlapped with the factors as identified in the literature study, and six constituted new SPI success factors that had not been identified in the literature. The factors were organized into three categories: (1) **organizational factors** related to the organizational structure, politics and culture, (2) **implementation factors** related to the planning, preparation, execution and management of the SPI projects and (3) **social factors** dealing with human behavior and reactions in the SPI context.

Despite the high overlap of the identified factors (24 out of 33), the focus on the factors differed greatly among the sources used in this study. The SPI factors that were recognized by the majority of the literature sources dealt with continuous management commitment and support, provision of resources and involvement of the technical staff in SPI. The factors that were mentioned by most of the interviewees focused on the social factors and the effects of SPI on the daily routines. Their concern mainly dealt with down-to-earth issues such as training in software process undergoing improvement, availability of the information on the SPI activities and results, and acceptance of the SPI activities by the technical staff. This was due to the fact that our interviewees represented the developers’ view on how SPI affected their daily routines.

As a result of this research we outlined thirty-three SPI sustainability success factors. The factors may already be used by software development organizations for defining, planning, monitoring and improving their SPI efforts, for diagnosing the reasons of SPI decay or for confirming the prerequisites that are necessary for carrying out SPI. The factors presented in this paper were used as a base to identify organizational, social and managerial factors that facilitate SPI, as presented in Section 5.3. The factors were also further incorporated into the SPI checklist created in Paper 7 and further enhanced and evaluated in Paper 8.

**Paper 7. Software Process Improvement Health Checklist**


To prevent degradation of the results of software process improvement (SPI) efforts, 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 suggested an SPI Health Checklist. The checklist is intended to be used for evaluating the quality of continuous SPI efforts. 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.

During this study, we first conducted a literature review and created a list of initial checklist items. Then, we interviewed seventeen software engineers that were involved in SPI or were affected by it. Altogether, our interviewees came from fifteen software companies. During the interviews, we scored the significance of the initial checklist items and identified the missing items. The research resulted in the SPI Health Checklist.
The forty two items of the SPI Health Checklist were grouped into seven clusters: Software process, SPI process, Process tools, Organization, Stakeholders, Process literacy and Social factors. The checklist clusters and their inherent items were presented in the paper on the high level of detail. During this study, we confirmed the importance and usefulness of the SPI Health Checklist by the engineers interviewed. 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. Almost 70% of the evaluated items scored over 4 out of 5. The most significant checklist items as recognized by the interviewees were the following: 1) 4.3 Resources are assigned for SPI activities, 2) 4.2 Management commits to support and sponsor the SPI effort, 3) 2.1 SPI goals and vision are clear and realistic, and 4) 6.4 SPI leaders possess knowledge and competence to lead SPI.

The interviewees stated that the presented SPI checklist could already be used for evaluating SPI efforts. They also pointed out that the importance of the checklist items varied with organizational contexts, such as 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 SPI Health Checklist was further enhanced and its potential use within the industry was evaluated in Paper 8, whereas the organizational contexts that impact process change were further evolved in Paper 10.

**Paper 8. Successful Process Improvement Projects are no Accidents**

Natalja Nikitina, Mira Kajko-Mattsson, Andrew James Nolan. Successful Process Improvement Projects are no Accidents, (Submitted to Journal on Software: Evolution and Process)

Software companies have a tendency to overestimate the benefits and underestimate the effort of process improvement. This may be, many times, due to lack of awareness of all the challenges and problems that SPI projects may meet. To succeed with their SPI projects, organizations need to have guidelines listing the properties that need to be fulfilled when improving their software processes.

This paper suggests an SPI checklist to be used for evaluating the status of SPI projects. The SPI checklist focus on continuous SPI and includes items communicating the properties that are essential for ensuring the necessary conditions of successful SPI projects. These properties should guide organizations in locating potential SPI problems that might lead to SPI project decline or failure. The SPI checklist presented in this paper was built upon a preliminary checklist that had been created in Paper 7. The checklist presented in this paper was enhanced to better reflect the industrial needs. It was applied on ten real-life finalized SPI projects at Rolls Royce. Our main goal was to provide a proof of concept of potential use of the SPI checklist within the industry, and to demonstrate its usefulness when evaluating the status of SPI projects.

The paper gives an account of the SPI checklist and the SPI projects, which were evaluated with the checklist. The evaluated SPI projects were of different success rates and sizes. The success rates were measured by comparing project’s delivered project values to their planned values. Moreover, each checklist item was evaluated with respect to how much it contributed to the success of the evaluated SPI projects. This relation was measured by identifying the
correlation between the fulfillment of a checklist item by the evaluated SPI projects and success rates of those projects, referred to as weights. The weights were used in three contexts (1) weights for all projects, using the data from all ten evaluated SPI projects, (2) weights for small projects, using the data from five small SPI projects, and (3) weights for large projects, using the data from five large SPI projects. 

The study shows a clear correlation between the fulfillment of the checklist items and SPI projects’ success. All the checklist items that could be weighed achieved their weights above zero, meaning that they had a positive impact on the projects’ success rates. As many as one third of the items achieved their weights between 0.6 and 1. These items had a strong influence on the success of the evaluated SPI projects. When analyzing the SPI projects by their size, we achieved different weights for large versus small SPI projects. The checklist items that had the highest weights for small SPI projects, mostly concerned the stakeholders, their expectations, understanding and preparations for SPI. The checklist items that had the highest weights for large SPI projects, on the other hand, were more related to higher degree of risk, complexity and impact of large SPI projects. They also concerned organization-wide factors, such as measurement and monitoring of the SPI project, and the ability to maintain the SPI process and sustain its results.

Based on the results of the checklist application, each evaluated SPI project was given a project score. The project scores were then compared to the actual success rates of those projects. The correlation coefficient between those two variables showed to be positive and as high as 0.9267, representing a strong positive straight-line pattern between the project scores and success rates. The coefficient of determination between the two variables studied was as high as 0.8587 representing the accuracy of 85.87% of predicting one variables via another.

The conclusions of our study indicate that the success of SPI projects is no accident but a foreseeable outcome of assessable characteristics such as effective management, dedicated and committed personal, and continuous sponsorship to mention a few, and, it is these characteristics that have been put into the SPI checklist. Therefore, this study shows that the SPI checklist can already be used by the software organizations for collecting information about possible problems and weaknesses of their SPI projects. The checklist presented in this paper constitutes one of the results of this thesis, as presented in Section 5.6.


Despite the fact that transitioning from one development method to another has become a common routine for many companies, very few reports give guidelines on how to conduct a process transition and there are no published process transition models whatsoever. As a result, the transitions often are costly, ineffective or unsuccessful.

This paper suggests a process model of Software Method Transition (SoMeT) to be used for transitioning from one method to another. Using a comparative research method, we studied and compared two transition process models mirroring the two real-life process transitions that had been conducted in CMSiPro. The two transition process models included a body of
knowledge encompassing process phases, activities, lessons learned and roles involved. They had similar overall design; however, they differed due to different conditions in which they were performed.

The paper presents the two real-life process transitions and analyzes them based on the following comparison criteria: (1) Purpose focusing on the general overview and purpose of the transitions, (2) Culture comparing the cultural contexts of the two transitions, (3) Roles focusing on the roles driving the transitions, (4) Attitude presenting the engineers' attitude towards the transitions, (5) Strategy and design concentrating on the strategy of the transitions, and (6) Lessons learned listing the experiences gained during the two transitions.

The main contribution of this paper is the SoMeT model. Bearing in mind the differences between the two compared real-life process transitions, the SoMeT model has a flexible structure and allows a free order of activities. The choice of activities depends on the process context at hand. The SoMeT model provides an opportunity to tailor the process on the fly, and allows flexibility when designing the executed process transition models.

The SoMeT model, as defined in this paper, could already be used by the organizations (1) to properly plan and prepare for process transition, (2) to choose appropriate transition strategy, and (3) to establish appropriate mechanisms for continuous SPI. The model, however, was further enhanced and its potential use within the industry was evaluated in Paper 10.


Adoption of new software methods has recently become a routinized practice for many software companies. Still however, there are no published process transition models except for the SoMeT model, presented in Paper 9 (Nikitina & Kajko-Mattsson, 2012a). This paper continued the previously started research (Nikitina & Kajko-Mattsson, 2012a). It suggested a process model of a software method adoption and elicited contextual factors of a software process change. The Software Method Adoption model (SMA model), suggested in the paper, provides guidelines for software development organizations on how to adopt new software methods in a structured and disciplined way. The model was based on the previously created the SoMeT model, which was reviewed, improved and evaluated in an industrial context during the process of writing this paper.

First, the SoMeT model was reviewed, which resulted in a conclusion that the model was very abstract in its original form and did not provide any direct guidelines for how to conduct method adaptations. Neither did it suggest process phases and activities or their placements. For this reason, the model was enhanced by adding stages and phases, and by adding contextual factors that impact the process change. Then, the model was evaluated against the current state of the art and state of practice. The evaluation against the current state of the art was conducted in form of a literature study, and resulted in an initial version of the SMA model. The evaluation against the current state of practice was conducted in form of interviews, by simply mapping the activities of the industrial method adoption projects to the activities of the SMA model. The model was mapped against six projects within three software and
engineering organizations. The selected projects concerned adoptions of new development methods and were in at least the final stage of the adoption process, if not finalized.

The study confirmed that: 1) the SMA model correctly reflected the state of the current software method adoption practice, and 2) the listed contextual factors impacted the process change process in studied projects. As many as 38 out of 52 method adoption activities were implemented in all the six evaluated method adoption projects, and all the fifty two activities were implemented in at least one of the projects. The evaluation had also led to the creation of a new activity called Track performance of the changed process.

The conclusions of our study indicate that the SMA model correctly reflects the state of the current software method adoption practice. Therefore, it can already be used for guiding software method adoption efforts. Nevertheless, due to different organizational contexts, the formality levels of the method adoption activities may vary strongly. Therefore, the activities of the model need to be adapted to the organizational needs and contexts. For this reason, the contextual factors and organizational needs must be considered when designing specific the software method adoption processes. The SMA model and the contextual factors impacting process change constitute two of the results of this thesis, as presented in Section 5.4 and Section 5.5.

4.2 Other contribution

During the course of writing this thesis, three additional research papers were written and presented on the research conferences. The papers add to the research contribution presented herein and are listed above.

Consequences of Business Growth on Software Processes

From Knowing Nothing to Knowing a Little: Experiences Gained from Process Improvement in a Start-Up Company

Success Factors Leading to the Sustainability of Software Process Improvement Efforts
Chapter 5

Research results

This chapter presents the results of this research and shows how they contribute to the existing body of knowledge on SPI. First, Section 5.1 lists the terminology used when describing the results. Section 5.2 outlines the software process improvement framework that positions the other research results in relation to the existing theories in SPI. Section 5.3 provides a thorough description of the organizational, social and managerial factors that lead to the efficient and successful SPI effort. Section 5.4 lists the contextual factors impacting the design of a software process change, and Section 5.5 presents the SMA model, for conducting and guiding software method adoption. Finally, Section 5.6 presents the SPI checklist that can be used for evaluation of the SPI efforts.

5.1 Terminology

Different organizations use different terminologies for the same activities, documents or roles. To ensure that the results of this thesis are understood and applied in a correct manner, below we provide the definitions of some basic terms. The definitions of the roles involved into the process change are, however, presented separately, in Section 5.2.2.

- **SPI project**: an endeavor realized by a set of organized software process improvement activities. It normally aims at assessing software process, planning for process improvements, deploying the improvements to the software process, and finally, at reviewing and evaluating the results of the improvements.
- **Method adoption project**: an endeavor realized by a set of organized activities that aims at introducing/adopting a new software method, set of practices, or processes.
- **Process undergoing change**: a software engineering process to be changed or affected by process improvement.
- **SPI project scope**: size, extent and scope of the changes and transformations brought to the software process by the SPI project, e.g. a pilot project, or a full process transformation along the whole organization.
- **SPI activity**: an activity corresponding to a process change that is deployed (implemented) as part of an SPI project.
- **Change strategy**: a strategy used for introducing changes and improvements into an organization, its culture, processes, roles and the like.
- **Organizational culture**: a collective behavior of people who are part of the same organization. It includes organizational values, norms, visions, beliefs, habits, assumptions, and ways of interacting and ways of perceiving information.
- **Action plan**: a sequence of process changes or activities that must be performed for changing the process towards its desired state. An action plan includes three major elements: (1) activities to be performed, as well as roles responsible for those activities, (2) schedule listing what tasks need to be done and when, and (3) resource plan determining allocation of resources and funds to specific activities. (Business Dictionary, u.d.)
• **Communication approach:** a structured approach defining what information is to be disseminated, to whom, how often, and by means of what communication channels.

• **Level of stakeholders’ involvement:** an extent to which stakeholders are involved in the software change process by, for instance, participating in the decision making, assessing software process, planning process change, and by preparing for and implementing the change.

• **Process control mechanism:** a mechanism that controls that the recommended process is performed and properly followed.

### 5.2 Software process improvement framework

The software process improvement is a highly complex process that is affected by great variety of technical, instrumental, procedural, organizational, social and managerial aspects. In order to exemplify the combined knowledge of the SPI discovered during the course of writing this thesis, we outlined the Software Process Improvement Framework (SPIF). The framework was not published in the papers attached to this thesis, since it was created in the final stage of this study in order to position the other research results in relation to the existing theories in SPI. The framework presented in this section aims to give a general overview of the complexity and diversity of SPI, present the existing SPI approaches and models, incorporate theories of the organizational change, and position the outcomes of this thesis in relation to the existing theories in SPI.

The SPIF framework is based on the existing framework for SPI environment proposed by Sami Zahran (1998), as presented in Section 2.3. Building on Zahran’s core ideas, we expanded the original model by changing its focus from only the software process to both the software process and organizational culture. This was done by incorporating the theories of organizational change, which are presented in Section 2.4. We kept the structure of the framework proposed by Zahran unchanged, but modified the names of the components as it is seen in the Figure 5.1.

Right side of Figure 5.1 represents the SPIF framework and its components graphically. In the core of the framework is the SPI environment. The SPI environment includes appropriate conditions for initiating and conducting the SPI effort as well as for sustaining its results in the long term. It incorporates the necessary organizational, social and managerial aspects to the SPI. It is further briefly presented in Section 5.2.1.

The four components of SPIF represent the technical, instrumental, procedural, organizational, social and managerial guidelines for conducting and guiding SPI effort. The components are the following:

1) **Infrastructure for SPI:** includes organizational and technical infrastructure needed for conducting the SPI effort. It involves human resources needed for leading, managing, coordinating and monitoring the SPI, and physical resources and technical infrastructure to support the changes to the process. This component is described in more detail in Section 5.2.2.
Software process infrastructure
Software process improvement action plan
Software process assessment
Software process improvement roadmap

SPI roadmap
Infrastructure for SPI
Change implementation approach
Assessment method
SPI environment

Figure 5.1  Left: Framework for SPI environment, (Zahran, 1998), Right: Software Process Improvement Framework (SPIF), based on (Zahran, 1998)

2) SPI roadmap: outlines a desired state of the process and organizational culture, and represents high level SPI vision and goals. In order to guide their SPI efforts organizations can use either software process maturity models or SPI vision and goals as their SPI roadmap. The description of the SPI roadmap is provided in Section 5.2.3.

3) Assessment method: outlines practices, techniques and measurement criteria for assessing the organizations current state. The method is strongly linked to the chosen SPI roadmap. This component is described in more detail in Section 5.2.4.

4) Change implementation approach: defines process model, strategy and overall guidelines for introducing changes to the organization, as well as for managing and monitoring the introduced process changes. Similar to the Assessment method, the Change implementation approach is linked to the chosen SPI roadmap. The description of the Change implementation approach is provided in Section 5.2.5.

5.2.1 SPI environment

Even if the initial SPI efforts show immediate gains, it is not a guarantee that the gains will be long lasting and sustainable. Processes undergoing improvement may demonstrate temporary gains as a result of initial organizational enthusiasm, eagerness and/or desire to do SPI. These gains, however, may not survive in the long run. The organizations may quickly reverse to the pre SPI process state, and thereby, make the SPI efforts a waste of time and resources, if they do not have SPI environment (Nikitina & Kajko-Mattsson, 2009).

The SPI environment includes mechanisms for establishing and sustaining process culture within the organization. It covers process documentation, process training and coaching, as well as continuous planning, monitoring and controlling the SPI effort and progress. The organizations should be aware of the environment for SPI continuously review it and take appropriate measures. In this way, they would become more conscious of their SPI efforts which, in turn, would contribute to reaching long-lasting, successful SPI results.

5.2.2 Infrastructure for SPI

Process improvement is a complex process that demands continuous supply of necessary infrastructure. It includes both human resources for leading, managing, coordinating and monitoring the SPI, and physical resources and technical infrastructure to support the changes to the process. They are presented in Figure 5.2.
1) Roles responsible for sponsoring SPI effort:
   - SPI sponsor
   - Company management
   - ...

2) Roles responsible for leading SPI effort:
   - SPI leader
   - SPI manager
   - SPI team
   - ...

3) Roles impacted by SPI:
   - Stakeholder
   - Technical staff
   - ...

Figure 5.2  Infrastructure for SPI (The figure lists possible examples of the resources and infrastructure. The symbol “…” implies that other roles and resources can be incorporated)

Human resources required for SPI are SPI sponsor, SPI manager, SPI leader, SPI team, and Process owner, who hold responsibilities for SPI sponsorship, management, coordination and process change. Those roles may be shared by several people or several roles may be assigned to one person, depending on the size of the organization and scale of the SPI effort. Other roles impacted by the SPI are Stakeholder and Technical staff. The naming of the roles and responsibilities vary in different academic and industrial contexts. For this reason, below we define the roles involved in SPI, (Nikitina & Kajko-Mattsson, 2012b), (Nikitina, et al., 2014).

- **SPI sponsor:** a senior executive or a group of executives that is responsible for leading, governing and sponsoring an SPI project.
- **SPI manager:** a person that manages and holds responsibility for an SPI project.
- **External SPI leader:** a person that is in charge of the overall SPI process. He/she initiates the improvement projects, requests resources, encourages local improvement efforts and establishes communication channels between different groups. External SPI leader is not the doer in the process to be improved. For this reason, he/she is seen as an external and independent role.
- **Internal SPI leader:** a person or a group within the technical staff who is responsible for supporting and following the SPI strategy on a local process level.
- **SPI team:** a group of people that coordinates an SPI project and introduces process changes.
- **Process owner:** a person or a group of people that manages and holds responsibility for the software process undergoing change.
- **Stakeholder:** a person or a group that is involved in or affected by SPI. Stakeholders include all the internal and external roles that are influenced by SPI.
- **Key stakeholder:** a person or a group of people who has interest in and has influence on an SPI project and its success.
- **Technical staff:** a group consisting of developers, testers, development managers, support personnel and other roles involved in executing the process undergoing the improvement.
They are the “doers” within the process undergoing improvement, and therefore, they get affected by the process change the most.

Physical resources and technical infrastructure are needed not only for supporting the changed software process, but also for sustaining the successful SPI effort. They should include physical resources, technical platforms, computing facilities and software tools, and should cover the purposes of:

- documenting and storing SPI related information
- distributing SPI related information to stakeholders
- managing SPI roles and responsibilities
- implementing process control mechanisms
- creating, estimating and monitoring SPI budget
- establishing training capabilities
- measuring and monitoring the results of SPI effort
- measuring and monitoring the software process
- documenting, storing and distributing the lessons learned from SPI

5.2.3 SPI roadmap

When starting the process improvement, organizations should have clear goals and vision for the improvement, defined by the SPI roadmap. In other words, the organizations should specify the desired state to be achieved by their SPI efforts. As shown in Figure 5.3, within SPI roadmap, we outline desired process state, and desired organizational culture. Those two are the main two elements of the desired future state within an organization.

The SPI roadmap for the desired software process state may imply that the organization aims either to improve the existing software process or adopt another software development method. The improvement of the existing software process is normally done by either: 1) reaching a certain maturity level of a maturity/reference model, such as CMM, CMMI, or SPICE, 2) reaching customized SPI vision and goals, or 3) implementing all the practices of the desired software development method. Examples of the software development methods are Waterfall, Scrum and Kanban, as presented in Section 2.1.3.

In order to support successful transition to the desired software process state, the organization needs to investigate if it has to change its organizational culture or not. In cases when the organization needs to change its culture in order to support the desired process, the SPI roadmap should also define the desired organizational culture. As shown in Figure 5.3, different models can be used for defining the desired organizational culture, such as Schneider’s or Hofstede’s models (Schneider, 1994), (Hofstede, 2001). Schneider’s culture model is presented in Section 2.4.2.

5.2.4 Assessment method

In order to have controlled, monitored and structured improvement effort, the organizations should first assess the initial state of the software process and the organizational culture. The assessment of the initial (pre-improvement) state is necessary for establishing the
1) Reaching a certain maturity level of a software process reference model - CMM - CMMI - SPICE - People CMM - ...

2) Reaching customized improvement goals

3) Adoption of another software development method - Waterfall model - V-model - Scrum - Kanban - ...

Desired process state

1) Reaching a certain maturity level of a software process reference model - CMM - CMMI - SPICE - People CMM - ...

2) Reaching customized improvement goals

3) Adoption of another software development method - Waterfall model - V-model - Scrum - Kanban - ...

Desired organizational culture

1) Schneider's culture model
   1.1) Control oriented culture
   1.2) Collaboration oriented culture
   1.3) Competence oriented culture
   1.4) Cultivation oriented culture

2) Hofstede's culture model
   2.1) ...

Figure 5.3  SPI roadmap (The figure lists possible examples of the process reference models, software development methods, and models of organizational culture. The symbol “…” implies that other models and methods can be incorporated)

improvement suggestions and recommendations. Moreover, one needs to have measured pre-improvement state in order to evaluate the results of the SPI effort.

As shown in Figure 5.4, there are three main types of the assessment. Those are the following: 1) self-assessment that is conducted by SPI sponsor, SPI manager, SPI leader, SPI team, Process owner, or Technical staff, 2) independent assessment that is conducted by the employees of the company, whose role is to provide an independent view of the SPI effort, or 3) external assessment that is conducted by external bodies, such as for instance, inspection companies or another independent third party organization (Nikitina, et al., 2014).

The scope of the assessment may differ depending on the size and influence of the SPI project, or needs of the organization. Three following assessment scopes are presented in Figure 5.4: 1) organization-wide assessment, 2) project or team specific assessment, and 3) individual or personal assessment.

As mentioned above, the process assessment method should be strongly linked to the chosen SPI roadmap. The process assessment method may be guided by a reference model, such as CMM, CMMI, or SPICE, in cases when the main SPI goal is achieving a certain maturity level of the reference model. In cases when the organization is not using a reference model for guiding its improvement effort, the assessments can be guided by an SPI goal or vision using light-weighted customized assessment methods, such as for instance Goal-Question-Metric method (Basili, 1992).
The methods and tools for assessing the organizational culture should also be linked to the SPI roadmap. This implies that the assessment of the organizational culture is done only when the organization aims to change it. The assessment can be done using organizational culture assessment instruments (Cameron & Quinn, 2000), assessment questionnaires or surveys, among other methods.

Despite the choice of the process assessment method, the organizations should normally define both quantitative metrics and qualitative benefits. Quantitative metrics are used in order to estimate the measurable results of the SPI effort. They include metrics related to software product, software process, software project and customer satisfaction. Qualitative benefits normally cover added business value, such as improved maintainability, process simplification or increased productivity.

### Change implementation approach

Introducing the recommended process changes can be a challenging task that should be performed in a structured and organized manner. To establish the effective SPI implementation, the organization should follow a structured change implementation approach.
Based on the choice of the SPI roadmap, the organization should choose the SPI implementation approach. As seen in Figure 5.5, we propose two types of the SPI implementation approaches: 1) SPI approach or process model, or 2) software method adoption model. The SPI approaches or process models can provide guidelines on how to improve the existing software process. They can be used for both reference model driven SPI and customized goal driven SPI efforts. The best known SPI approaches are the IDEAL model (McFeeley, 1996), ISO/IEC 15504-4 (ISO/IEC, 2004), Quality Improvement Paradigm (QIP) (Basili, 1985), and Six Sigma (Bhote, 1989), as presented in Sections 2.2.1 and 2.2.2. Regarding the software method adoption model, there are no process models and very few guidelines that organizations can use for guiding the adoption of new software methods. Therefore, as a result of this research we developed a Software Method Adoption process model (SMA model), which is presented in Section 5.5 of this thesis.

Process change impacts organization and business, as well as people and their behaviors, resulting in organizational change. In order to bring structured organizational change, one should use organizational change management approaches, such as Kotter’s change management approach (Kotter, 1995), or Kandt’s change management program (Kandt, 2003). The organizational change management is described in Section 2.4.

How fast or slow the change should be implemented? Who should drive the change? What should be the size and scale of the changes? The answers to these among other questions should depend on the selected change strategy and change process. Change strategy is used for introducing changes and improvements into an organization, its culture, processes, roles and the like. The common change strategies that can be used for SPI are the following:
• **Top-down or commanding** process change strategy is normally dictated by top management of the organization (Pries-Heje & Johansen, 2007). This strategy has limited stakeholder involvement in the SPI process.

• **Bottom up or employee-driven** process change strategy is normally driven from the bottom of the organization, by its employees (developers or testers). This strategy includes high stakeholders’ involvement in the SPI process. Often, the SPI project is even led by development team.

• **Iterative** process change strategy implies that the changes to the process are introduced in the slow manner, divided in the serious of iterations.

• **Big-Bang** process change strategy implies that the changes to the process are introduced all at ones within a short time period (Nikitina & Kajko-Mattsson, 2011a).

### 5.2.6 Placement of the research results within SPIF

The four additional research results, presented in Sections 5.3, 5.4, 5.5 and 5.6, aim to compliment the SPIF framework. Figure 5.6 shows the relationship between the research results and SPIF. The four research results are represented by four yellow boxes in the figure, whereas the framework is presented in its original view, as shown in the right side of Figure 5.1.

The first research result, the organizational, social and managerial factors facilitating SPI effort, is a part of the SPI environment. Therefore, it contributes to its description and understanding. The factors represent SPI success factors that are related to organizational, social and managerial aspects of SPI. The factors are described in detail in Section 5.3. Another research result is a list of contextual factors that impact the process change, as presented in Section 5.4. Those factors need to be considered when defining the elements of the SPIF framework that are suitable for the organization, its needs and context. The factors strongly impact the choice of the SPI roadmap and change strategy. The third research result presented in this thesis is a model for Software Method Adoption, (SMA model). The SMA model can be used as a change implementation approach for guiding the adoption of new software development methods. It is presented in...
Section 5.5. Finally, the SPI checklist, presented in Section 5.6, lists the properties of successful and sustainable SPI projects. It can be used in order to evaluate SPI process, and review whether all the necessary elements, and conditions defined by the SPIF framework are established within the organization.

5.3 Organizational, social and managerial factors facilitating SPI effort

The factors, presented in this section, represent SPI success factors or conditions for successful and sustainable SPI efforts that relate to the organizational, social and managerial aspects of SPI. The factors also contribute to the description and understanding of the SPI environment, presented in Section 5.2.1.

The organizational, social and managerial factors facilitating SPI effort, presented in this section, were taken from the list of SPI success and sustainability factors, which was reported in Paper 6 of this thesis (Nikitina & Kajko-Mattsson, 2012b). The SPI success and sustainability factors resulted from both literature and empirical studies, as presented in Appendix B. From the thirty three SPI success and sustainability factors we selected thirty one factors that directly relate to organizational, social and managerial aspects of SPI. The other two factors were discarded because they concern only procedural and technical aspects of SPI. The thirty one selected factors are presented in this section. Out of them, twenty five factors were found during the literature study, and twenty nine were found during the empirical study. Twenty three overlapping factors were found in both studies. The sources of the factors are presented in Appendix B. The factors are the following.

1) Management continuously supports and commits to the SPI process: To provide long-term sustainable results, SPI requires continuous investment in time, resources and effort. This, in turn, requires strong management commitment and continuous support (Stelzer & Mellis, 1998), (Rainer & Hall, 2002), (Hall, et al., 2002), (Paulish & Carleton, 1994), (Abrahamsson & Iivari, 2002). Strong management commitment helps retain high priority of the SPI projects and the continuous management support helps assure continuous supply of the required resources. It is especially important in the initial SPI phases during which the cost of the SPI activities is higher than the initially expected and planned cost (Goldenson & Herbsleb, 1995).

2) Resources are dedicated to SPI: As many as 72% of SPI improvement projects have suffered from lack of resources and constant time pressure (Goldenson & Herbsleb, 1995), (Beecham, et al., 2003), (Niazi, et al., 2010). SPI projects cannot run on their own. Investment in resources has been recognized not only for starting and implementing the SPI projects but also for sustaining the achieved results (Goldenson & Herbsleb, 1995), (El Emam, et al., 1999), (Niazi, et al., 2006b), (Curtis & Paulk, 1993), (Hall, et al., 2002). Without dedicated resources, SPI can only rely on the engagement of individuals, which tends to decrease with time.

3) SPI responsibilities are clearly specified and compensated: People involved in SPI should have clear responsibilities and compensation for their effort (Goldenson & Herbsleb, 1995), (El Emam, et al., 1999). Time dedicated to the SPI activities should be compensated in the same manner as other work. Otherwise, the SPI activities may be done in a rush, they may be neglected, they may be delayed or they may even be forgotten.
4) **Competent external SPI leaders are designated:** The level of competency, experience, commitment and engagement of the external SPI leaders can greatly determine the success of the SPI projects (Goldenson & Herbsleb, 1995) (El Emam, et al., 1999) (Stelzer & Mellis, 1998), (Hall, et al., 2002). However, this may not always be enough. Authority and respect paid to the external SPI leaders is just as important. When lacking the authority, trust and respect, then the ideas of SPI leaders may not be supported and successfully transmitted to the process change (Goldenson & Herbsleb, 1995) (El Emam, et al., 1999) (Stelzer & Mellis, 1998), (Hall, et al., 2002).

5) **Internal SPI leaders are designated:** The internal SPI leaders are recognized as important SPI actors since they take on the immediate responsibility for leading and supporting continuous process improvement (Stelzer & Mellis, 1998), (Rainer & Hall, 2002), (Paulish & Carleton, 1994), (Nazir, et al., 2008). By possessing knowledge of the process, they are able to adapt the improvement suggestions to the different needs of the development teams, projects and cultures. They help SPI activities get started and their engagement aids in winning support of their team members towards SPI.

6) **The level of technical staff turnover is low:** High people turnover can become a significant barrier to the sustainability of the SPI efforts. When the key employees leave the company, so does the knowledge of the process and SPI. With a high technical staff turnover, more effort needs to be spent on the education and training of the new hires.

7) **SPI is aligned with business goals:** The goals of SPI projects should not only go in line with the standardization of the process and quality standards, but also with the business goals of the company. Alignment of SPI goals with the organizational business goals contributes to the better management of, commitment to and support of the SPI projects (Dyba, 2000), (Dyba, 2005), (Hall, et al., 2002), (Nazir, et al., 2008), (Dangle, et al., 2005).

8) **SPI is aligned with organizational policies and strategies:** Improvement projects often conflict with the existing organizational policies by requiring changes to the routines and processes that are common to the whole organization. Therefore, organizational policies have to be aligned with the SPI goals and vice versa. In cases when organizations do not have any policies, they have to establish ones and make the process standardization and improvement coherent with them (Goldenson & Herbsleb, 1995), (Beecham, et al., 2003).

9) **SPI methods are tailored to specific organizational contexts and needs:** Each organization is different with respect to its structure, culture and policies. For this reason, SPI initiatives should consider the contextual specifics of the organizational culture, product characteristics, customer availability and people influenced by the process. The adaptation of process improvement methods to the specific organizational contexts and needs helps address individual problems and contributes to sustainable SPI efforts (Stelzer & Mellis, 1998), (Hall, et al., 2002).

10) **Stakeholders are trained in software process:** The technical staff needs to be trained in the process and its techniques and practices in order to fully understand their role in the process change. They need to be prepared for the process improvement and understand the reasons behind each suggested change. Other stakeholders that are affected by SPI should also receive necessary training. Otherwise, they will less likely follow the new process (Beecham, et al., 2003). In organizations or cultures where knowledge of the process is low, the training in the
process is even more important (Niazi, et al., 2010). The levels of training may differ from stakeholder to stakeholder with respect to the stakeholders’ training needs and their level of involvement in SPI.

11) Stakeholders are continuously mentored and coached: Training in the new process contributes to its understanding and allows stakeholders to follow it dedicatedly. This however may not always be enough. Some stakeholders may misunderstand the process or continue following old techniques and practices. Therefore, the stakeholders should be continuously mentored and coached in SPI and the process changes. (Rainer & Hall, 2002). Continuous mentoring and coaching increases the credibility of the strategic SPI decisions and contributes to building trust both in those decisions and in the new process.

12) SPI leaders possess experience and expertise in SPI: SPI implies changes to the deeply ingrained organizational culture, habits, working patterns and manners that have been developed throughout a long time. It is easier to change them if the SPI team possesses enough knowledge and experience in implementing software process improvement changes. If there is lack of such knowledge and experience, then there is a risk of using unsuitable SPI strategy and of having poor SPI execution, which could potentially fail the SPI projects (Rainer & Hall, 2002) (Niazi, et al., 2006b), (Beecham, et al., 2003), (Niazi, et al., 2006a), (Nikitina & Kajko-Mattsson, 2009).

13) SPI goals and objectives are clear and realistic: Clear, realistic and well communicated SPI goals and objectives contribute to good understanding of the SPI process and assurance that they are well understood across all the organizational levels (El Emam, et al., 1999). Realistic SPI goals and objectives lead to realistic expectations and aid in maintaining high motivation for and support of the SPI activities. Unrealistic, too ambitious or unreachable goals and objectives, on the other hand, may jeopardize the SPI projects, by decreasing employees’ engagement and motivation even in projects with positive results (Goldenson & Herbsleb, 1995), (Nazir, et al., 2008).

14) SPI project is effectively managed: Management of the SPI project involves a wide range of activities such as planning for change, identifying actors involved, ensuring the level of understanding the process changes, monitoring the status of SPI, evaluating the progress, and the like. It needs to be performed in an effective and professional manner (Niazi, et al., 2010). Without project management, the SPI project is doomed to fail or it may lead to chaos (Stelzer & Mellis, 1998).

15) Process improvement effort is flexible: Software process should continuously change and adapt to the organizational needs and situation. SPI activities can be risky and may not always lead to the expected results. Therefore, SPI should be flexible and allow for experimenting with the process. In cases when the process change is proven to be unsuccessful or unsuitable, the organization should be able to quickly rollback the process to the pre-change status.

16) Information about SPI activities and its results is disseminated: SPI projects bring many changes to the process and daily routines. These changes have to be communicated to all the stakeholders that can be directly or indirectly impacted. Insufficient communication of the SPI changes and its results may lead to lack of transparency of the SPI projects, confused personnel and poor quality process. Team collaboration and communication, on the other
hand, may help the staff members to exchange knowledge and experience during the improvement projects and contribute to a more coherent organizational culture (Stelzer & Mellis, 1998).

17) **Process standards are defined and enforced:** When the stakeholders lack dedication and commitment to the new process, people are tempted not to follow the process standards, unless there is a strong control mechanism in place (Zahran, 1998). Even when properly trained, the staff may not follow the newly introduced process. Therefore, in order to guarantee that the process is dedicatedly followed by all the stakeholders, the process standards should be defined, enforced and controlled by the SPI managers (Zahran, 1998).

18) **SPI effort brings positive results:** The results achieved by the early SPI effort should be positive and should speak for themselves. Early gains of SPI effort can encourage and motivate stakeholders to continue with the SPI activities and change the opinions of those who did not support SPI from the very beginning.

19) **Software process is monitored and measured:** Continuous process monitoring and measurement indicates whether the SPI activities are effective or not, and allows providing early feedback on the sustainability of the SPI efforts. Hence, the process should be monitored and measured on a continuous basis in order to reassure its purpose and to increase the engagement of the SPI supporters. Measured and acknowledged SPI will positively affect team morale and motivation (Rainer & Hall, 2002), (Dyba, 2000), (Dyba, 2005), (Paulish & Carleton, 1994).

20) **Software process and its efficiency are continuously reviewed:** Continuous software process reviews allow learning from previous experience and from experimenting with the process, which, in turn, contributes to a self-driven continuous SPI and thereby, to long lasting SPI results (Dyba, 2000), (Dyba, 2005). Process reviews also help to identify problems in the current process and to acknowledge benefits achieved by SPI. Without frequent reviews and changes to the process, gains of SPI would soon outdate.

21) **SPI effort is continuous:** To sustain the gains of the SPI efforts, the organization should view the SPI as a continuous activity, that cannot be achieved without mechanisms for continuous process review and tuning (Zahran, 1998), and comprehensive support of those responsible for the process (Stelzer & Mellis, 1998). In addition, all the roles responsible for the SPI project should continuously reaffirm commitment to change, communicate progress of improvement, and provide continuous feedback and motivation (Stelzer & Mellis, 1998).

22) **Stakeholders have a common understanding of the process undergoing change:** The process cannot be efficiently improved unless it is properly understood. The technical staff and management have to reach a common understanding on the status of the current process, its problems and possible solutions, as well as the organization’s vision and improvement goals (Stelzer & Mellis, 1998). Common understanding of the current and new process, suggested changes and its potential benefits is important to increase support for SPI among all the stakeholders.

23) **Stakeholders are aware of complexity, challenges and benefits of SPI:** Since SPI requires continuous effort and often brings mainly long-term results, organizations must make sure that all the stakeholders involved are aware of complexity, challenges and potential benefits of SPI. This can be realized via education, training and effective communication. Raising
awareness of SPI and effective communication of its complexity, challenges and benefits strongly affects the success of the SPI projects (Niazi, et al., 2006b), (Varkoi, 2002), (Basri & O’Connor, 2010), (Niazi, et al., 2010).

24) **Stakeholders have realistic expectations:** In order to be satisfied with the SPI and its results, the stakeholders should have realistic expectations. Otherwise, they would get disappointed with SPI and would not continue with it, even though SPI brought positive results.

25) **Technical staff accepts SPI activities:** Changes to the process may affect daily work of many employees. Therefore, it is important that all the members of the technical staff agree and accept future process changes (Hall, et al., 2002), (Sweeney & Bustard, 1997). This can substantially decrease inertia to change. Acceptance of process changes can be encouraged by high involvement of the technical staff in the SPI activities.

26) **Technical staff is committed to the SPI process:** Together with the increased motivation and engagement, the commitment of the technical staff can become a driving wheel of process improvement. Committed staff takes on the responsibility and ownership of the process and keeps process in a healthy state (Nikitina & Kajko-Mattsson, 2011a).

27) **Stakeholders are being encouraged to support SPI:** Commitment to and support of SPI by all the stakeholders is a great asset to help successful SPI implementation and to decrease inertia towards change. However, it is not easy to reach everybody’s support for SPI. Therefore, already during the early stages of the SPI project, the management should start encouraging the stakeholders towards supporting SPI. Encouraging the technical staff in SPI from the very beginning would increase support, motivation and engagement in future SPI activities.

28) **Technical staff is rewarded for contribution to SPI success:** Moral appreciation and financial rewarding acknowledge individual contributions to SPI. Recognized contribution engages and motivates people to continue with the SPI effort (Rainer & Hall, 2002), (Zahran, 1998).

29) **SPI leaders encourage initiative and openness of stakeholders:** SPI leaders should focus on process weaknesses and problems and should encourage initiative, innovation, creativity and openness of the stakeholders involved. Without it, employees cannot share valuable ideas, and thereby, contribute to process improvement (Curtis & Paulk, 1993).

30) **Technical staff participates in SPI:** Technical staff constitutes an important process knowledge and experience asset (Stelzer & Mellis, 1998), and may provide useful feedback on the suggested SPI changes (Sweeney & Bustard, 1997). The involvement and participation of the technical staff reduce resistance to change, and thereby, strongly impact the SPI success (Stelzer & Mellis, 1998), (Hall, et al., 2002), (Nazir, et al., 2008). By being involved in the SPI activities, the technical staff members feel more motivated to adhere to the process changes, and therefore, they are more likely to accept them (Stelzer & Mellis, 1998), (Hall, et al., 2002).

31) **Technical staff owns the software process:** It is important that not only external and internal SPI leaders but also all the technical staff members take on the ownership of the process to be improved. The members should take the responsibility for tailoring the process and for continuously improving it. It is only in this way they will feel more affiliated with the process and more responsible for future process improvement. This, in turn, will lead to a built-in, self-driven continuous process improvement process, which, in turn, will strongly contribute to the sustainability of the SPI results (Rainer & Hall, 2002).
5.4 Contextual factors impacting process change

The appropriateness and suitability of the chosen change process and strategy to the context of the organization has a significant impact on the overall results, sustainability and success of any change project, including process improvement projects. The contextual factors that impact the process change process, listed in this section, need to be considered when defining the SPI roadmap suitable for the organization, its needs and context, and when defining a change implementation approach and strategy.

Below we list seven contextual factors that strongly impact the process change, which were reported on in Paper 10, (Nikitina & Kajko-Mattsson, 2014). These factors have been evaluated by individuals involved in six industrial projects. The evaluation had confirmed that the listed contextual factors have significant impact on the process change process and should be considered when defining SPI roadmap, choosing a change strategy and designing a change implementation process.

1) Size and scope of the process change

The size and scope of an improvement project may vary greatly from project to project. A project of a small scope normally concerns only an introduction of a few process changes or an adoption of a new set of practices. Usually, it does not require any change to the formal organizational structure, social behaviors, team structure or composition. The main process workflow and organizational culture would normally remain the same. The company would only change a few of their development practices or techniques. This would be usually done via a continuous SPI by means of iterative process changes and reviews.

A large process improvement project, on the other hand, concerns a transformation of an existing software development process and organizational culture. This requires changes to the process workflow, to the formal structure of the organization, to the organizational culture as well as to social behaviors, team structure and/or composition. Due to a large amount of changes required and impact on the stakeholders involved, a large project is time consuming, costly, and sometimes very risky. It demands a clear vision, strong leadership, sufficient expertise, continuous sponsorship and management commitment to the process change. It should be supported by well-defined methods for process change, such as SPI models, and well-established approaches for managing organizational changes.

2) Existing organizational culture

Every organization, including software organizations, has its own culture, character and identity, referred to as organizational culture (Schneider, 2000). Organizational culture is stronger in larger, older and more successful organizations (Schneider, 2000). It usually consists of organization’s core values, norms, principles, beliefs and stories (Anderson & Ackerman Anderson, 2010).

In control-driven organizations with hierarchical organizational structure, the process change is often implemented top-down using commanding change strategy (Pries-Heje & Johansen, 2007). It is led by management with little to no participation of the developers and testers in the process change. In people-oriented organizations with collaboration-oriented
organizational cultures, on the other hand, the process change is often implemented bottom-up using the employee-oriented change strategy (Pries-Heje & Johansen, 2007). The process change is often enforced by high levels of developers' engagement, involvement and participation in the change process. This implies that most of the developers and testers take active part in the process change by providing their feedback, sharing their ideas and volunteering to introduce process changes.

3) Attitude towards the process change

The amount of training and preparation activities depends, among many things, on the attitude towards the process change in the organization. The positive attitude implies that the stakeholders are engaged and prepared for the process change, whereas the negative attitude implies that the stakeholders are reluctant and skeptical towards the process change.

Stakeholders’ attitude to and readiness for process change impacts how fast the process change can be introduced, and how many additional preparation and training activities are required. In case of a negative attitude and low interest towards process change, the stakeholders should be trained, coached, informed, engaged, and thoroughly prepared for the process change in advance. The amount of training activities depends also on the stakeholders’ knowledge of the new practices and processes.

4) Urgency of the process change

How fast or slow the process changes need to be implemented depends, among many things, on the actual urgency of it. The urgency implies how fast the organization needs and/or is able to change and improve its processes.

In case a process change needs to be introduced expeditiously, the organization can use Big-Bang change strategy. This strategy implies that the process is transformed in one fell swoop within a short time period (Nikitina & Kajko-Mattsson, 2011a). The reasons for urgency may vary. For instance, the process change may seriously obstruct the development work and/or its stakeholders are highly motivated towards the process change.

If, for one reason or another, the organization is not able to conduct a process change in a swoop, it is recommended that the organization uses an iterative process change strategy instead. This strategy implies that the process changes are introduced slowly in a portion-wise manner until the desired process state is achieved. This choice of the change strategy also depends on the size of the process change, readiness of the organization to future changes, and on the need to change the organizational culture.

5) Readiness to the process change

Positive attitude towards process change is not enough. The readiness of the organization to do the changes is just as important. Usually, it depends on the employees’ knowledge of the new software development practices and processes. In cases when the stakeholders involved do not have sufficient knowledge of the processes or methods to be introduced, and tend to avoid the change, a substantial amount of training, coaching and preparation activities need to be conducted. The process changes should not start before assuring that the stakeholders involved are ready for it.
6) Gap between the existing and desired organizational cultures

Software process improvements or new software method adoptions may require changes to the existing organizational culture. In cases when the gap between the existing and desired organizational culture is big, a sufficient amount of change management activities need to be conducted, which requires dedicated resources, long time, continuous dedication, strong leadership and employees’ engagement.

7) Piloting the changes on the small scale project first

Before introducing significant changes to the software process, the organizations are recommended to pilot the process changes on small-scale projects. It allows the organizations to gain the experience on how to conduct changes, shows early results and costs of the process changes, identifies risks, and serves as a success-story for gaining acceptance and motivation of other stakeholders. Based on the results of the pilot projects, the organizations may then decide whether to introduce the same process changes on a larger scale. In this way, the organizations decrease the risks of failing large scale process changes.

Piloting the process changes are only suitable for the organizations that have two or more separate projects or products, as well as two or more development teams that work independently. The development project or team, selected for piloting the process changes should not deal with critical functions of the organization. It should be independent of other development projects and it should involve the motivated and engaged employees. Piloting process changes is not recommended in small organizations having only one or few development teams, working on one project or product.

5.5 SMA process model

When the organization aims to improve its process by adopting new software development method, they should use a process model to guide the process change. Unfortunately, despite a large number of software method adoptions taking place in the industry, there are very few guidelines that the organizations can use on how to adopt new software methods. There is no widely accepted and clearly defined method for software method adoption. Therefore, during this research we developed the Software Method Adoption process model (the SMA model), presented in this section.

The SMA process model or the SMA model has been evaluated in six industrial method adoption projects, as reported in Paper 10, (Nikitina & Kajko-Mattsson, 2014). The summary of the evaluation results are presented in Table C.1 in Appendix C. The evaluation confirmed that the SMA model correctly reflected the state of the current software method adoption practice. As many as 38 out of 52 process activities of the SMA model were implemented in all the six projects, and all the fifty two activities were implemented in at least one of the projects.

The SMA model can be used as an SPI implementation approach for the organizations that aim to adopt new software development methods. The incorporation of the SMA model to the SPIF framework is illustrated in Figure 5.5 in Section 5.2.5, as part of the change implementation approach component. The SMA model is a process model to be used for guiding adoption of a new software method, of a set of new practices, or an adoption of significant process changes...
or transformations. It is applicable in small pilot projects or in large process transformations. In both cases, it should be applied in the same manner.

**Figure 5.7** General representation of the SMA model

**Figure 5.8** Detailed representation of the SMA model
The SMA model is represented on two levels: phase and activity levels. Figures 5.7 and 5.8 show the two levels. As can be seen there, the model consists of two main stages, where each stage consists of a set of phases, which, in turn, consist of a set of activities. The stages represent the major states of the software method adoption process. They are 1) Method adoption, during which a new method or a set of practices are implemented, and 2) Continuous improvement during which the implemented process is being stabilized, tailored to the organizational needs and continuously improved.

The phases of each stage are individual parts of the process that aim to fulfill a particular process adoption mission. As shown in Figure 5.7, the Method adoption stage consists of five consecutive phases: 1) Initiation during which a process change project is initiated, 2) Assessment during which an existing software process is assessed and a desired process or method is being chosen, 3) Planning during which a process change is planned for, 4) Preparation during which preparations for the process change are made, and finally, 5) Deployment, during which the process is changed and the new method is adopted. The Continuous improvement stage, on the other hand, consists of two iterative phases: 1) Process review during which the process is being reviewed and the possible improvements are identified, and 2) Deployment during which the identified improvements to the process are implemented.

The Deployment phase appears twice in the SMA model both in the Method adoption stage and in the Continuous improvement stage, see Figure 5.7. It however contains the same set of general activities. Therefore, it is presented only once in the detailed illustration of the SMA model, see Figure 5.8. There is, however, one difference between the Deployment phases in the different stages. The Deployment phase in the Method Adoption stage may be conducted either in a big-bang manner implying that the changes are introduced all at once or in an iterative manner implying that the changes are conducted in a portion-wise manner until the desired method or process gets fully adopted. In the Continuous improvement stage, the Deployment phase is always conducted in an iterative manner, and followed by the Process review phase.

The activities of each phase are process units that fulfill a specific, well-limited and scoped function. In the SMA model, they all contribute to a well-structured, systematic and optimal method adoption process. We have put much care into identifying the genuine activities and finding their most optimal placements in the SMA model. We believe that, together, they strongly contribute to a well-structured, systematic and optimal method adoption process. Nevertheless, we understand that all processes strongly vary from case to case. Therefore, not all the adoption activities are mandatory and neither are their placements within the phases compulsory. Their applicability and placements can be adjusted to the organizations’ specific cultures and method adoption contexts. We, still however, strongly recommend to use the model as it is, unless some of its parts go against organizational policies or change strategies.

Due to different organizational contexts and needs, the formality levels of the method adoption activities may vary strongly. Therefore, we recommend that the formality levels are to be adapted to the needs of the adoption effort at hand.
5.6 SPI checklist

The SPI checklist represents the properties of successful and sustainable SPI projects. By evaluating the states of those properties, the organizations can identify problems within their SPI projects, locate their causes and initiate relevant process improvements.

The SPI checklist has been reported on in Paper 8 of this thesis (Nikitina, et al., 2014). The SPI checklist was evaluated on ten industrial SPI projects at Rolls Royce. The results of the evaluation are presented in Table D.1 of Appendix D. All the checklist items that could be weighed achieved their weights above zero, meaning that they had a positive impact on the projects’ success rates. The evaluation results also indicated that the success of SPI projects was no accident but a foreseeable outcome of assessable characteristics that had been put into the SPI checklist.

The SPI projects can be evaluated with the SPI checklist on a continuous basis either internally (in-house) or externally. The internal evaluations may be conducted by either (1) an SPI team, whose role is to lead, manage, guide and/or be accountable for the SPI projects, or (2) independent assessors, who are still the employees of the company and whose role is to provide an independent view of the SPI projects. The external assessors, on the other hand, are external bodies, such as for instance, inspection companies who have a high degree of economical and organizational independency. Their role is to assess SPI projects, evaluate their efficiency and provide recommendations for their improvement.

Whether SPI projects are evaluated internally or externally, the accuracy of the evaluation results strongly depends on the level of honesty and objectivity of the people involved. Moreover, different stakeholders may have different views on whether an SPI project achieves certain properties. Therefore, it is recommended that the stakeholders involved in SPI project evaluation be of different roles such as SPI manager, high and low level managers, developers and testers.

Since the SPI process is cyclical, the SPI projects can be assessed with the SPI checklist at any phase of the process. However, we strongly recommend that the projects are regularly assessed at the end of the software process improvement cycle. The SPI project assessment should then be followed not only by planning and implementing the improvements to the software process undergoing change but also by planning and implementing the improvements to the way the SPI is introduced, managed and communicated.

The SPI checklist consists of sixty four checklist items representing the properties of successful SPI projects. As presented in Figure 5.9, the items are organized into eleven clusters, each focusing on a specific aspect of an SPI project. These are the following: (1) SPI goals and scope, (2) Sponsorship and resources, (3) SPI roles and responsibilities, (4) Stakeholder management, (5) Competence and knowledge, (6) Preparation for SPI, (7) Deployment management, (8) SPI project governance and support, (9), SPI project monitoring and measurement, (10) Management commitment, and (11) Stakeholders’ attitude.

Each checklist item is placed in one cluster only. The choice of its parental cluster is based on the items’ degree of impact on the cluster. For instance, the item 1.4 Key stakeholders reach
<table>
<thead>
<tr>
<th>1. SPI goals and scope</th>
<th>6. Preparation for SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1: SPI project has a business case that shows its ROI</td>
<td>6.1: Organizational culture and competence are analyzed and considered in SPI method and change strategy</td>
</tr>
<tr>
<td>1.2: SPI vision and main goals are well-defined and disseminated to key stakeholders</td>
<td>6.2: Processes undergoing change and SPI goals are correctly understood and considered in SPI method and change strategy</td>
</tr>
<tr>
<td>1.3: SPI vision and main goals are stable and realistic</td>
<td>6.3: Similar process improvements were previously implemented on another SPI project(s)</td>
</tr>
<tr>
<td>1.4: Key stakeholders reach consensus on SPI vision and main goals</td>
<td>7. Deployment management</td>
</tr>
<tr>
<td>1.5: Scope of SPI project is clearly bounded</td>
<td>7.1: SPI method and change strategy are defined</td>
</tr>
<tr>
<td>1.6: Scope of SPI project is continuously evaluated and modified when needed</td>
<td>7.2: SPI method and change strategy are tailored to fit organization, its processes and stakeholders</td>
</tr>
<tr>
<td>2. Sponsorship and resources</td>
<td>7.3: Defined SPI method and change strategy are tailored to fit organization, its processes and stakeholders</td>
</tr>
<tr>
<td>2.1: Sponsorship and resources for SPI are continuously provided</td>
<td>7.4: SPI action plan is created</td>
</tr>
<tr>
<td>2.1.1: Budget &amp; resource for initiating, planning, and preparing SPI activities are assigned and provided</td>
<td>7.5: SPI schedule is created based on sensible estimates and planning</td>
</tr>
<tr>
<td>2.1.2: Budget &amp; resource for deploying SPI activities are assigned and provided</td>
<td>7.6: Impact of each SPI activity is carefully analyzed before it is being deployed</td>
</tr>
<tr>
<td>2.1.3: Budget &amp; resource for sustaining gains achieved by SPI are assigned and provided</td>
<td>7.7: SPI activities are deployed according to SPI action plan and schedule</td>
</tr>
<tr>
<td>2.2: Amount of provided resources is aligned with scope, complexity, and novelty of SPI project</td>
<td>7.8: Impediments blocking SPI activities are continuously identified and removed</td>
</tr>
<tr>
<td>3. SPI roles and responsibilities</td>
<td>7.9: SPI action plan and schedule are continually reviewed and modified, if needed</td>
</tr>
<tr>
<td>3.1: Accountability for processes undergoing change is assigned</td>
<td>7.10: Necessary tools to support SPI project are adequate and available</td>
</tr>
<tr>
<td>3.2: Accountability for SPI project is assigned</td>
<td>8. SPI project governance and support</td>
</tr>
<tr>
<td>3.3: Responsibilities for leading local improvements are defined and assigned</td>
<td>8.1: Governance of SPI project is consistent with its value, size, criticality, and risk</td>
</tr>
<tr>
<td>3.4: Responsibilities for deploying SPI activities are defined and assigned</td>
<td>8.2: SPI risks are identified, analyzed, and planned for</td>
</tr>
<tr>
<td>3.5: Responsibilities for maintenance and support of SPI activities are defined and assigned</td>
<td>8.3: SPI risks are continuously monitored</td>
</tr>
<tr>
<td>3.6: SPI responsibilities are assigned, understood and agreed upon by the stakeholders</td>
<td>8.4: Deployed SPI activities are institutionalized and supported by organizational policies</td>
</tr>
<tr>
<td>4. Stakeholder management</td>
<td>8.5: Essential mechanisms to control adherence to the deployed process are established and continuously improved</td>
</tr>
<tr>
<td>4.1: Stakeholders and key stakeholders are clearly identified</td>
<td>9. SPI project monitoring and measurement</td>
</tr>
<tr>
<td>4.2: Stakeholders’ support and readiness to SPI are correctly understood and considered in SPI method and change strategy</td>
<td>9.1: SPI objectives are measurable by carefully designed SPI measures (metrics and quality indicators)</td>
</tr>
<tr>
<td>4.3: Stakeholders are continuously trained, coached, and mentored in newly deployed processes, according to their training needs</td>
<td>9.2: General SPI project milestones and reporting mechanisms are defined and established</td>
</tr>
<tr>
<td>4.4: Stakeholders are encouraged to support and commit to SPI project</td>
<td>9.3: SPI metrics are consistently collected</td>
</tr>
<tr>
<td>4.5: Communication approach is established and it is agreed upon who receives communications, when, how, and to what level of detail</td>
<td>9.4: Progress of SPI project is continuously monitored, analyzed, and communicated</td>
</tr>
<tr>
<td>4.6: Stakeholders, who get affected by an SPI activity, are informed about its purpose and impact</td>
<td>9.5: Results of SPI project are continuously reviewed, analyzed and reflected on in SPI action plan</td>
</tr>
<tr>
<td>4.7: Stakeholders, who get affected by an SPI activity, are prepared for its deployment</td>
<td>9.6: SPI measures (metrics and quality indicators) are regularly reviewed and modified, if needed</td>
</tr>
<tr>
<td>4.8: Stakeholders are informed about complexity, challenges, and benefits of SPI project</td>
<td>10. Management commitment</td>
</tr>
<tr>
<td>4.9: Stakeholders have realistic expectations from SPI project</td>
<td>10.1: High level management (including project sponsor(s)) is engaged and committed to SPI project</td>
</tr>
<tr>
<td>5. Competence and knowledge</td>
<td>10.2: Middle level management is engaged and committed to SPI project, if any</td>
</tr>
<tr>
<td>5.1: Stakeholders possess necessary technical skills and domain knowledge to run the deployed process</td>
<td>10.3: Low level management is engaged and committed to SPI project, if any</td>
</tr>
<tr>
<td>5.2: Stakeholders possess knowledge about processes undergoing change and process to be deployed</td>
<td>11. Stakeholders’ attitude</td>
</tr>
<tr>
<td>5.3: SPI manager has knowledge, leadership competence and personal profile to lead SPI</td>
<td>11.1: Key stakeholders are highly committed to SPI project</td>
</tr>
<tr>
<td>5.4: SPI manager is engaged in and passionate about the SPI project</td>
<td>11.2: Stakeholders accept planned SPI activities</td>
</tr>
<tr>
<td>5.5: Competence of SPI manager is consistent with size, complexity, criticality, and novelty of SPI project</td>
<td>11.3: Stakeholders follow defined software process</td>
</tr>
<tr>
<td>11.1: Key stakeholders are highly committed to SPI project</td>
<td>11.4: Stakeholders, who run a software process, have partial ownership of it</td>
</tr>
<tr>
<td>11.2: Stakeholders accept planned SPI activities</td>
<td>11.5: Stakeholders trust and respect SPI managers</td>
</tr>
<tr>
<td>11.3: Stakeholders follow defined software process</td>
<td>11.6: Stakeholders participate in SPI project</td>
</tr>
</tbody>
</table>

Figure 5.9 SPI checklist
consensus on SPI vision and main goals may be relevant to two clusters, such as SPI goals and scope and Stakeholders’ attitude. However, it has the highest impact on the stability of the SPI vision and goals. Hence, we place it in the SPI goals and scope cluster.

The SPI checklist can already be used by software organizations for collecting information about possible problems and weaknesses of their SPI projects. Irrespective of when and how the checklist is used, it is important that the individuals using it are honest and objective about the status of the SPI projects. They also have to be aware of the fact that the fulfillment of the SPI checklist items only illustrates the compliance with the recommended activities; it does not evaluate the efficiency of those activities.
Chapter 6

Conclusions

Despite significant efforts, most of the software process improvement (SPI) initiatives fail (Ferreira & Wazlawick, 2011). This is due to the fact that SPI is a highly complex process and organizations often lack expertise or knowledge in SPI, underestimate their magnitude and required effort and/or, most importantly lack explicit guidelines on how to choose a suitable SPI approach and follow it (Kuvaja, et al., 1999). Nowadays, the majority of existing SPI approaches focus only on technical, instrumental, procedural and partly organizational aspects of SPI (McFeeley, 1996), (ISO/IEC, 2004), (Basili, 1985). They present prescriptive processes for guiding the implementation of process changes. However, studies show that the success of the SPI initiatives is highly dependent on the organizational, social and managerial aspects, such as senior management commitment, staff involvement, training and mentoring, just to mention a few (Niazi, et al., 2006b). Those aspects are not sufficiently covered by the majority of the SPI approaches or models (McFeeley, 1996), (ISO/IEC, 2004), (Basili, 1985). The little knowledge on organizational, social and managerial aspects of SPI that is covered by a few SPI approaches is mostly scattered across the domain. There is just one model that provides a general overview of the SPI field and, to some extent, considers organizational, social and managerial aspects of SPI (Zahran, 1998). The model however is outdated. Therefore the research problem discussed in this thesis concerns lack of a holistic overview of the current SPI domain that provides sufficient coverage of organizational, social and managerial aspects of SPI. Incorporating the organizational, social and managerial aspects of SPI is argued to be essential for software development organizations to succeed with SPI initiatives and sustain their results.

Because of the significant relevance of the research problem to the software industry, we based this research on a strong industrial cooperation and involved thirty software development organizations throughout our study. We conducted the research in line with the inductive reasoning, which guided the research into building the knowledge from empirical studies. However, since the field of software processes and SPI contained some already defined knowledge, we incorporated literature studies to the inductive reasoning approach in some stages of this research. The main research methods of this study are action research and case studies, whereas data collection methods are primarily structured interviews, participatory observations and surveys.

Based on the above announced research design and methods, this thesis explored the organizational, social and managerial aspects of SPI and placed them in the context of the SPI domain. The main research result of this thesis is the Software Process Improvement Framework (SPIF). The framework provides an overview of the SPI domain and positions theories representing organizational, social and managerial aspects of SPI in the context of existing SPI approaches, models, methods and practices, as presented in Section 5.2. Four additional research results have been obtained from this research, which have complemented SPIF, and fulfilled four secondary research objectives of this study. These additional research results are: 1) the list of organizational, social and managerial factors facilitating SPI effort, as presented in Section 5.3; 2) the
Table 6.1  Results of this thesis

<table>
<thead>
<tr>
<th>Main research result</th>
<th>Corresponding research objective</th>
<th>Papers or thesis in which the research work is presented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Process Improvement Framework</td>
<td>Main research objective: to provide an overview of the SPI domain and position theories representing organizational, social and managerial aspects of SPI in the context of existing SPI approaches, models, methods and practices.</td>
<td>PhD thesis</td>
</tr>
<tr>
<td>Additional research results</td>
<td>Corresponding research objective</td>
<td>Papers or thesis in which the research work is presented</td>
</tr>
<tr>
<td>Organizational, social and managerial aspects of SPI</td>
<td>Research objective 1: to elicit SPI success factors that relate to organizational, social and managerial aspects of SPI</td>
<td>Paper 6: Outlined and evaluated list of SPI success and sustainability factors</td>
</tr>
<tr>
<td>Contextual factors that impact software process change</td>
<td>Research objective 2: to elicit contextual factors that impact the design of a software process change</td>
<td>Paper 1, Paper 2 and Paper 3: Studied three organizations that experienced process change</td>
</tr>
<tr>
<td>SMA process model</td>
<td>Research objective 3: to explore how to conduct software method adoption as part of software process improvement and organizational change processes.</td>
<td>Paper 4 and Paper 5: Studied two real-life software method adoptions</td>
</tr>
<tr>
<td>SPI checklist</td>
<td>Research objective 4: to explore how to assess the status of the SPI effort and provide recommendations for its improvements, considering organizational, social and managerial aspects of SPI.</td>
<td>Paper 6: Outlined and evaluated list of SPI success and sustainability factors</td>
</tr>
</tbody>
</table>

The research results are based on the findings of the ten research papers that are presented in Part II of this thesis. Table 6.1 shows the research results, their corresponding research objectives and the papers in which the research work was described. Each of the outcomes is described in detail in the following paragraphs.

SPIF was based on the existing framework for SPI environment proposed by Sami Zahran (1998). Building on Zahran’s core ideas, we expanded the original framework by changing its focus from only the software process to both the software process and organizational culture. This was done by incorporating the theories of organizational change. The SPIF framework gives a general overview of the complexity and diversity of process improvement process, and provides a base for representing a variety of SPI approaches, models, methods, and practices that are important to understand and acknowledge before and when conducting process improvement. The framework incorporates organizational, social and managerial aspects of SPI, such as adaptation of the organizational culture and process training.

In the core of the SPIF framework is the SPI environment that includes appropriate conditions for initiating and conducting the SPI effort as well as for sustaining its results in the long term. SPIF includes the four following components: 1) Infrastructure for SPI, that involves human
resources, physical resources and technical infrastructure to support the changes to the process, 2) SPI roadmap that outlines a desired state of the process and organizational culture, and represents high level SPI vision and goals, 3) Assessment method that outlines practices, techniques and measurement criteria for assessing the organizations current state, and 4) Change implementation approach that defines process model, strategy and overall guidelines for introducing changes to the organization, as well as for managing and monitoring the introduced process changes. SPIF was used in order to show the placement of four additional results of this study within SPI domain.

The organizational, social and managerial factors facilitating SPI effort are SPI success factors or conditions for successful and sustainable SPI efforts that relate to the organizational, social and managerial aspects of SPI. The factors also contribute to the description and understanding of the SPI environment of SPIF. The organizational, social and managerial factors facilitating SPI effort were taken from the list of SPI success and sustainability factors, which were reported on in Paper 6 of this thesis (Nikitina & Kajko-Mattsson, 2012b). The factors may already be used by software development organizations for defining, planning, monitoring and improving their SPI efforts, for diagnosing the reasons of SPI decay or for confirming the prerequisites that are necessary for carrying out SPI.

The contextual factors impacting process change are the factors that need to be thoroughly analyzed and considered before choosing a change strategy and designing a change process. Those factors need to be considered when defining the elements of SPIF that are suitable for the organization, its needs and context. The factors strongly impact the choice of SPI roadmap and Change implementation approach of the SPIF framework.

The Software Method Adoption (SMA) model is a process model to be used for guiding adoption of a new software method or a set of new practices. The SMA model can be used as a change implementation approach, as defined in the SPIF framework. The SMA model was evaluated in six industrial method adoption initiatives within three organizations. The evaluation confirmed that the model correctly reflected the state of the current software method adoption practice. Therefore, it is argued that the SMA model can already be used for guiding software method adoption efforts. Nevertheless, the activities of the model need to be adapted to the organizational needs and contexts.

The SPI checklist is a list of the properties of successful and sustainable SPI projects. By evaluating the states of those properties, the organizations can identify problems within their SPI projects, locate their causes and initiate relevant process improvements. The checklist can be used in order to evaluate SPI process, and review whether all the necessary elements, and conditions defined by the SPIF framework are established within the organization. The checklist was evaluated on ten real-life SPI projects at Rolls Royce. The evaluation results showed a clear correlation between the fulfillment of the checklist items and SPI projects' success. This indicated that the success of SPI projects was no accident but a foreseeable outcome of assessable characteristics that had been put into the SPI checklist. Therefore, the SPI checklist can already be used by the software organizations for collecting information about possible problems and weaknesses of their SPI projects.

SPIF and the four additional results of this thesis may significantly benefit software development organizations that plan to conduct software process change, or have already
done it. These organizations may use SPIF for getting an overview of the process improvement process and the theories, methods and tools that should support it, as well as for structuring the process improvement effort. The other results of this thesis can be used 1) for incorporating organizational, social and managerial aspects in the process change, 2) for choosing suitable change strategy and SPI roadmap based on the organizational context, 3) for guiding the adoption of new software methods, and finally 4) for evaluating and improving the process improvement effort.

Positioning the results of this research in relation to the above-stated research problem, we would like to point out that SPIF and the four additional results of this thesis do not provide a comprehensive or holistic view on the SPI effort. Software process improvement is a highly complex and diverse process which embraces a myriad of various technical, instrumental, procedural, organizational, social and managerial aspects. The structured and holistic overview of the SPI process covers several scientific fields; among them are software engineering, organizational science, and psychology. Hence, the results of this thesis neither provide a comprehensive nor holistic view on the SPI effort. They, however, contribute to such comprehensive and holistic view and to the understanding of the complexity of the SPI effort by acknowledging the importance of organizational context, positioning the organizational, social and managerial aspects of SPI within the SPI domain and highlighting the importance of those aspects.

Based on the empirical and theoretical knowledge that was developed during the course of writing this thesis, we stress the importance of organizational, social and managerial aspects of SPI on the results of improvement initiatives. Our findings allow us to argue that implementing recommended software development processes or practices using well defined SPI approaches is not enough. When implementing the recommended software development processes, organizations need to consider organizational, social and managerial aspects of SPI. Organizational aspects of SPI, such as long term management commitment and support, sufficient resources and budget, as well as clear goals and directions are necessary for creating effective environment for continuous process improvements. Social aspects of SPI, such as expertise, competence, attitude, and expectations of the people are the key for the successful and lasting process improvement. People leading the change should be engaged in it and should take ownership and accountability for it, whereas people who are impacted by process change need to agree with and commit to the process changes. Finally, managerial aspects of SPI ensure smooth and effective SPI implementation, by thoroughly planning and preparing for it and carefully introducing the process changes. Therefore, successful and sustainable process improvement efforts can be achieved only by considering organizational, social and managerial aspects of SPI, along with technical, instrumental and procedural aspects.

6.1 Future work

Because of the magnitude and complexity of the SPI field, a large amount of potential research questions can emerge from the results presented herein. Therefore, it would be difficult to enumerate all possible research that can develop from or use the content and results of this thesis. Nonetheless, this thesis and its SPIF framework can contribute
significantly to research aiming at further development of the comprehensive overview of SPI. For that, the framework and its elements should be carefully explored and expanded in order to include other relevant approaches for SPI that were not mentioned in the framework, and to update it in a continuous manner as new approaches are developed. The framework could also be complimented with a set of general guidelines that organizations would use for choosing appropriate SPI methods and approaches from the SPIF framework based on the context and their needs at hand. In this way, the results of this thesis and the broader understanding of the software process change that is presented herein can be seen as a significant contribution to future studies aiming at further developing the comprehensive view of SPI, or aiming at providing informative guidelines for managing process improvement process.
References


Alexandre, S., Renault, A. & Habra, N., 2006. OWPL: A Gradual Approach for Software Process Improvement in SMEs, s.l., IEEE.


Available at: http://www.businessdictionary.com/definition/action-plan.html
[Accessed 15 April 2013].

Cameron, K. & Quinn, R., 2000. Diagnosing and Changing Organizational Culture. Upper Saddle River,
New Jersey: Pearson Education, Inc..

Available at: http://www.ccpgames.com
[Accessed 15 April 2013].


Kniberg, H. & Skarin, M., 2010. Kanban and Scrum: making the most of both. USA: C4Media Inc.


Appendix A

Data collection instruments

Appendix A.1

Questionnaire on historical process evolution

Introductory questions
1. What is your name?
2. What company do you work for?
3. What is your role in the company?

Main interview questions

Part 1. Information about the company
1.1. What is the size of the company (in terms of people)?
1.2. What does the company work with?
1.3. Which software development method does the company use?

Part 2. Business growth
2.1. Has the company had any business growth?
2.2. Does the company encounter the continuous growth or is it coming in intervals?
   2.2.1 In case of the continuous growth.
      i. Since when has the company had a business growth?
   2.2.2 In case of growth in intervals.
      i. How many business growths has the company had?
      ii. How frequently does the company encounter business growth?
      iii. What is a business growth in the context of your company?
      iv. For how long do the intervals of business growth last?
         (1) The last one
         (2) The former ones

Part 3. People relation
3.1. Did the amount of employees change as a result of business growth(s)? If yes, how?
   3.1.1. Have the roles of the employees changed as a result of business growth(s)? If yes, how and why?
   3.1.2. Were new employees hired as a result of business growth(s)? If yes, how many and to which positions?
   3.1.3. Were the company employees fired as a result of business growth(s)? If yes, how many and why?
3.2. How has the hiring of new employees affected the software development process?
   3.2.1. What training techniques were introduced for newly employed?

Part 4. Influence of business growth on the software development process
4.1. If the business growth was coming in intervals (Question 2.2). For each business growth under discussion.

4.1.1. What was the software development process like before the business growth? Did you use any software development method? Which one? Did you use it fully?

4.1.2. What was the software development process like after the business growth? Did you use any software development method? Which one? Did you use it fully?

4.2. Was the software development process affected by the business growth?

4.2.1. If yes, how was the process changed and adapted to the growing team? What was done? Please list the changes to the process. For each change/adaptation.
   i. Please name the change.
   ii. Please describe the change. Was this change for better or for worse? Please comment on it.
   iii. How did this change impact the development process?
   iv. Was the change intentional or accidental? Was a change reactive or proactive?
   v. Why did each change occur? What caused the change?
   vi. Who initiated the change? Who supported the initiation?
   vii. What other challenges were met in order to implement the change?

4.2.2. If not, how come that it was not changed?

4.3. What challenges has the business growth created? Please list the challenges.

4.4. Please list the problems you have met due to the business growth. For each problem.

4.4.1. What was the problem?

4.4.2. How did you address each problem?

Part 5. The cause of business growth

5.1. What was/were the reason(s) of business growth(s)?

5.2. Was the development process scalable before business growth? Is it scalable now?

5.3. Were the development teams prepared for business growth when it happened? If yes, how?

5.4. Does the company have any business growth plan? If yes, please briefly describe what such a plan covers?

5.5. Do you think business growth will happen again/continue in this company?

5.6. How is the company prepared for future business growth?

5.7. Does the company plan for business growth? In what way?

Part 6. Lessons learned from business growth

6.1. What lessons did you learn from business growth?

6.2. What is your suggestion to avoid negative impact of future business growth?

6.3. What can be done to adapt the process for smooth future business growth?

6.4. How can we keep or even improve the maturity of the process during the business growth? Is it possible?
Appendix A.2

Questionnaire for assessing the results of SPI

Introductory questions
1. What is your name?
2. What is your role in the company?

Main interview questions

Part 1. Solved problems
1.1. Describe the original problem and its solution(s).
   *Comment: to be conducted by the interviewer.*
2.1. Please comment on the problem.
   *Comment: The interviewee is expected to confirm the problem description, and provide his opinion, if need arises.*
   
   1.2.1. Has this problem reappeared?
       If yes, then
       i. In what form does it show itself?
       ii. To what extent?
       iii. Why has this problem come back?
       iv. What is/are the reason(s) for it?
       v. Could you please rank this problem as
           - very serious (5),
           - serious (4),
           - medium (3),
           - low (2),
           - minor nuisance (1).
       vi. Has the reappearance of this problem created new problems?
           (1) If yes, then what is the new problem?
           (2) If not, then what has helped to sustain the improvement results?

1.3. Remind the problem solution again.
   *Comment: to be conducted by the interviewer.*
1.4. Has the introduction of Scrum contributed to the resolution of the problem?
    1.4.1. What was the Scrum solution?
    1.4.2. How has the Scrum solution affected the problem?
    1.4.3. Has the Scrum solution solved the problem partially or totally?
    1.4.4. Is the original solution still used?
1.5. What is your opinion on the current solution to this problem?
1.6. Has the current solution to this problem created new problems?
    1.6.1. What problems have been created?
1.7. Do you have any other suggestions for solving this problem?
    1.7.1. What is the solution?
Part 2. Partially solved problems

2.1. Describe the original problem and its solution(s).
*Comment: to be conducted by the interviewer.*

2.2. Please comment on the problem.
*Comment: The interviewee is expected to confirm the problem description, and provide his opinion, if need arises.*

2.3. Has the status of this problem been changed?

2.3.1. If yes, then
   i. To what extent?
   ii. How has the problem evolved?
   iii. What has triggered the changes in the status of the problem? What is/are the reason(s) for it?
   iv. Could you please rank this problem as
      - very serious (5),
      - serious (4),
      - medium (3),
      - low (2),
      - minor nuisance (1).

2.3.2. If not, then why has the status of this problem not been changed?

2.4. Has this problem created new problems?

2.4.1. What problems have been created?

2.5. Remind the problem solution again.
*Comment: to be conducted by the interviewer.*

2.6. Has the introduction of Scrum contributed to the resolution of the problem?

2.6.1. What was the Scrum solution?

2.6.2. How has the Scrum solution affected the problem?

2.6.3. Has the Scrum solution solved the problem partially or totally?

2.6.4. Is the original solution still used?

2.7. What is your opinion on the current solution to this problem?

2.8. Has the current solution to this problem created new problems?

2.8.1. What problems have been created?

2.9. Do you have any other suggestions for solving this problem?

2.9.1. What is the solution?

2.9.2. Please motivate the solution.

Part 3. Unsolved problems

3.1. Describe the original problem and its solution(s).
*Comment: to be conducted by the interviewer.*

3.2. Please comment on the problem.
Comment: The interviewee is expected to confirm the problem description, and provide his opinion, if need arises.

3.3. Has the status of this problem been changed?
   3.3.1. If yes, then
      i. To what extent?
      ii. How has the problem evolved?
      iii. What has triggered the changes in the status of the problem? What is/are the reason(s) for it?
      iv. Could you please rank this problem as
          • very serious (5),
          • serious (4),
          • medium (3),
          • low (2),
          • minor nuisance (1).

3.3.2. If not, then why has the status of this problem not been changed?

3.4. Has this problem created new problems?
   3.4.1. What problems have been created?

3.5. Has the introduction of Scrum contributed to the resolution of the problem?
   3.5.1. What was the Scrum solution?
   3.5.2. How has the Scrum solution affected the problem?
   3.5.3. Has the Scrum solution solved the problem partially or totally?

3.6. Has the current solution to this problem created new problems?
   3.6.1. What problems have been created?

3.7. Do you have any other suggestions for solving this problem?
   3.7.1. What is the solution?
   3.7.2. Please motivate the solution.

Part 4. New problems

4.1. Have any new problems been encountered of which we have not been aware of so far?
   4.1.1. What is the new problem?

4.2. Let us now go through all new problems that we have been listed before.

Comment: to be conducted by the interviewer.
   4.2.1. Name the problem
   Comment: to be conducted by the interviewer.
   4.2.2. Please describe the problem
   4.2.3. What are the consequences of it?
   4.2.4. What causes the problem?

4.3. Could you please rank this problem as
   • very serious (5),
   • serious (4),
4.3.1. Please motivate your rating.

4.4. Do you have any other suggestions for solving this problem?
   4.4.1. What is the solution?
   4.4.2. Please motivate the solution.

Part 5. Ranking problems
5.1. Could you please rank all the mentioned problems once again as
   - very serious (5),
   - serious (4),
   - medium (3),
   - low (2),
   - minor nuisance (1).

Part 6. Evaluation of Scrum
6.1. How has the implementation of Scrum affected following
   6.1.1. organization
   6.1.2. software development productivity
   6.1.3. products' quality
   6.1.4. well-being of employees
6.2. Has the implementation of Scrum affected anything else?
6.3. How much are you committed to the Scrum methodology?
   6.3.1. Could you rank your level of commitment as
           - high
           - medium
           - low
   6.3.2. Please motivate your answer.

Part 7. Preparations for SPI activities
7.1. What do you expect from the process improvement?
7.2. How much are you committed to the process improvement?
   7.2.1. Could you rank your level of commitment as
           - high
           - medium
           - low
   7.2.2. Please motivate your answer.
7.3. How can we make you to be more committed?
7.4. Have you been involved in an SPI process before?
7.5. What do you think is important when improving a process?
7.6. Do you have any suggestions for how to make a process assessment?
7.7. What values should we bring out when making process improvement?
7.8. What pitfalls do you think we should avoid, when making process improvements?
<table>
<thead>
<tr>
<th>Question</th>
<th>Rating</th>
<th>Commentary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are the development tasks assigned to you clear enough?</td>
<td></td>
<td>“No, I always have to clarify the them and the feature requirements”</td>
</tr>
<tr>
<td>Are the Product Owners always available to clarify the requirements?</td>
<td></td>
<td>“We never can reach them”</td>
</tr>
<tr>
<td>Are the company vision and the goals of the project clear to you?</td>
<td></td>
<td>“I have no idea what the goals are”</td>
</tr>
<tr>
<td>To what extent do the meetings and other non-development work affect your productivity?</td>
<td></td>
<td>“Meetings take majority of my time. I cannot make anything done”</td>
</tr>
<tr>
<td>Are responsibilities of the teams and the product owners clear to both parties?</td>
<td></td>
<td>“I still have no idea, who is responsible for what”</td>
</tr>
<tr>
<td>Are you satisfied with the software development process that is currently used?</td>
<td></td>
<td>“This process does not work for us”</td>
</tr>
</tbody>
</table>
8. How interesting and creative is your work?

- "It is very boring"
- "I enjoy my work a lot"

9. Are you continuously improving your skills and knowledge?

- "I have the same knowledge and skills as when I started here"
- "I learn new things almost every day"

10. Are you satisfied with the company you are working for?

- "It is depressing here. I want to leave"
- "I'm very happy with this company"

11. Does the company provide social activities to a satisfactory extend?

- "No, we have no opportunity to socialize with the colleagues"
- "Yes, it does"

12. Are you sufficiently rewarded for your work?

- "No, my work is not appreciated or rewarded enough"
- "Yes, my work is fully appreciated and rewarded"

13. How strong is the team-cohesion in your team?

- "It does not even feel like a team"
- "It is very strong"

14. How good is your relationship with your colleagues?

- "Very bad"
- "Very good"

15. Which of the mentioned questions are specifically important for you? Write question numbers _________

16. Why has your motivation increased? What changes/factors have facilitated it?

____________________________________________________________________________
____________________________________________________________________________

17. What changes/factors have decreased your motivation?

____________________________________________________________________________
____________________________________________________________________________

18. What are the key success factors to improve and sustain high developers' motivation in the company?
19. Is there something missing in this questionnaire that affects developers' motivation? If yes, please specify what it is?
Appendix A.4

Questionnaire for process assessment

Introductory questions
1. What is your name?
2. What is your role in the company?

Main interview questions
1. What are the benefits of current software process?
2. What are the problems of current software process?
3. What do you think is the reason of the process transition?
4. What are your expectations of the process transition?
5. What are your worries about the process transition?
6. Are you satisfied with the current software process? To what extent?
7. Do you support the process transition? To what extent?
Appendix A.5

Questionnaire for evaluation of SPI success factors

Introductory questions
4. What is your name?
5. What company do you work for?
6. What is your role in the company?

Main interview questions
1. Are you aware that the information you provide will be kept confidential?
2. Have you been involved in process improvement or process transition before? To what extent?
   2.1 If yes, have the results of the process improvement been lasting?
      i. If yes, why do you think the results have been lasting?
      ii. If no, why do you think the results have not been lasting?
3. What factors contribute to the process improvement sustainability? Please list them and motivate your answers.
4. What factors prevent the process improvement sustainability? Please list them and motivate your answers.
5. What are your suggestions for keeping the process improvement results lasting/sustainable? Please list them and motivate your answers.
Appendix A.6

Questionnaire for evaluation of the SPI checklist

Introductory questions
1. What is your name?
2. What company do you work for?
3. What is your role in the company?

Main interview questions
1. Present the core clusters of the SPI checklist.
2. For each cluster do the following.
   2.1 Present the cluster and give an overview of its checklist items.
   i. For each checklist item ask the following:
      (1) Is this checklist item significant for SPI? On the scale 1-5, please grade the significance of this checklist item for the SPI success?
         (Grading scale: This checklist item..
           - does not contribute to the success of SPI (1)
           - has a small impact on the success of SPI (2)
           - has a medium impact on the success of SPI (3)
           - has a high impact (4)
           - has may fail the SPI project (5)
      (2) Please motivate your grading.
   2.2 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 defining SPI checklist and SPI project health attributes can be useful for the SPI process? Please motivate your answer.
5. Do you have any suggestions for how one could evaluate the SPI checklist with the use of the checklist items (SPI project health attributes)? If yes, please describe it.
Appendix A.7

Questionnaire for assessing SPI projects

Introductory questions
1. What is your name?
2. What is your role in the company?

Main interview questions
1. How many people worked full-time on the SPI team?
2. How many stakeholders were affected by the introduced process changes?
3. How long was the SPI project?
4. How were the SPI team and the stakeholders distributed?
5. What was the degree of novelty of the SPI project?
6. What was the level of complexity of the introduced process changes?
7. To what extend was the SPI project dependent on other projects?
8. What were causes of failures or successes of this SPI project?
9. What was the success rate of this project in percentage?

Comment: The success rate of the SPI project was determined by evaluating how much of its planned measurable goals were achieved.
Appendix A.8

Assessment survey for evaluation of SPI projects against SPI checklist

1. SPI goals and scope

☐ 1.1: SPI project has a business case that shows its ROI (Return On Investment)
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 1.2: SPI vision and main goals are well-defined and disseminated to key stakeholders
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 1.3: SPI vision and main goals are stable and realistic
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 1.4: Key stakeholders reach consensus on SPI vision and main goals
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 1.5: Scope of SPI project is clearly bounded
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 1.6: Scope of SPI project is continuously evaluated and modified when needed
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

2. Sponsorship and resources

☐ 2.1: Sponsorship and resources for SPI are continuously provided
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 2.1.1: Budget & Resource for initiating, planning and preparing SPI activities is assigned and provided
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 2.1.2: Budget & Resource for deploying SPI activities is assigned and provided
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 2.1.3: Budget & Resource for sustaining gains achieved by SPI is assigned and provided
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 2.2: Amount of provided resources is aligned with scope, complexity and novelty of SPI project
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

3. SPI roles and responsibilities

☐ 3.1: Accountability for processes undergoing change is assigned
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 3.2: Accountability for SPI project is assigned
   How significant is this item for the success of this project?  □ High □ Medium □ Low □

☐ 3.3: SPI responsibilities are understood, assigned and agreed upon by stakeholders
3.3.1: Roles responsible for leading overall SPI (SPI manager) are defined and assigned
How significant is this item for the success of this project?  High □  Medium □  Low □

3.3.2: Roles responsible for leading local improvements are defined and assigned
How significant is this item for the success of this project?  High □  Medium □  Low □

3.3.3: Roles responsible for deploying SPI activities are defined and assigned
How significant is this item for the success of this project?  High □  Medium □  Low □

3.3.4: Roles responsible for maintenance and support of SPI activities are defined and assigned
How significant is this item for the success of this project?  High □  Medium □  Low □

4. Stakeholder management

4.1: Stakeholders and key stakeholders are clearly identified
How significant is this item for the success of this project?  High □  Medium □  Low □

4.2: Stakeholders’ support and readiness to SPI are correctly understood and considered in SPI method and change strategy
How significant is this item for the success of this project?  High □  Medium □  Low □

4.3: Stakeholders are continuously trained, coached and mentored in newly deployed processes, according to their training needs
How significant is this item for the success of this project?  High □  Medium □  Low □

4.4: Stakeholders are encouraged to support and commit to SPI project
How significant is this item for the success of this project?  High □  Medium □  Low □

4.5: Communication approach is established and it is agreed upon who receives communications, when, how and to what level of detail
How significant is this item for the success of this project?  High □  Medium □  Low □

4.6: Stakeholders, who get affected by an SPI activity, are informed about its purpose and impact
How significant is this item for the success of this project?  High □  Medium □  Low □

4.7: Stakeholders, who get affected by an SPI activity, are prepared for its deployment
How significant is this item for the success of this project?  High □  Medium □  Low □

4.8: Stakeholders are informed about complexity, challenges and benefits of SPI project
How significant is this item for the success of this project?  High □  Medium □  Low □

4.9: Stakeholders have realistic expectations from SPI project
How significant is this item for the success of this project?  High □  Medium □  Low □

5. Competence and knowledge

5.1: Stakeholders possess necessary technical skills and domain knowledge to run the deployed process
How significant is this item for the success of this project? High □ Medium □ Low □

☐ 5.2: Stakeholders possess knowledge about processes undergoing change and process to be deployed

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 5.3: SPI manager has knowledge, leadership competence and personal profile to lead SPI

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 5.4: SPI manager is engaged in and passionate about the SPI project

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 5.5: Competence of SPI manager is consistent with size, complexity, criticality and novelty of SPI project

How significant is this item for the success of this project? High □ Medium □ Low □

6. Preparation for SPI

☐ 6.1: Organizational culture and competence are analyzed and considered in SPI method and change strategy

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 6.2: Processes undergoing change and SPI goals are correctly understood and considered in SPI method and change strategy

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 6.3: Similar process improvements were previously implemented on another SPI project

How significant is this item for the success of this project? High □ Medium □ Low □

7. Deployment management

☐ 7.1: SPI method and change strategy are defined

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.2: SPI method and change strategy are tailored to fit organization, its process and stakeholders

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.3: Defined SPI method and change strategy is properly followed

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.4: SPI action plan is created

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.5: SPI schedule is created based on sensible estimates and planning

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.6: Impact of each SPI activity is carefully analyzed before it is being deployed

How significant is this item for the success of this project? High □ Medium □ Low □

☐ 7.7: SPI activities are deployed according to SPI action plan and schedule

How significant is this item for the success of this project? High □ Medium □ Low □
7.8: Impediments blocking SPI activities are continuously identified and removed
How significant is this item for the success of this project?  High □  Medium □  Low □

7.9: SPI action plan and schedule are continuously reviewed and modified if needed
How significant is this item for the success of this project?  High □  Medium □  Low □

7.10: Necessary tools to support SPI project are adequate and available
How significant is this item for the success of this project?  High □  Medium □  Low □

8. SPI project governance and support

8.1: Governance of SPI project is consistent with its value, size, criticality and risks
How significant is this item for the success of this project?  High □  Medium □  Low □

8.2: SPI risks are identified, analyzed and planned for
How significant is this item for the success of this project?  High □  Medium □  Low □

8.3: SPI risks are continuously monitored
How significant is this item for the success of this project?  High □  Medium □  Low □

8.4: Deployed SPI activities are institutionalized and supported by organizational policies
How significant is this item for the success of this project?  High □  Medium □  Low □

8.5: Essential mechanisms to control adherence to the deployed process are established and are continuously improved
How significant is this item for the success of this project?  High □  Medium □  Low □

9. SPI project monitoring and measurement

9.1: SPI objectives are measurable by carefully designed SPI measures (metrics and quality indicators)
How significant is this item for the success of this project?  High □  Medium □  Low □

9.2: General SPI project milestones and reporting mechanisms are defined and established
How significant is this item for the success of this project?  High □  Medium □  Low □

9.3: SPI metrics are consistently collected
How significant is this item for the success of this project?  High □  Medium □  Low □

9.4: Progress of SPI project are continuously monitored, analyzed, and communicated
How significant is this item for the success of this project?  High □  Medium □  Low □

9.5: Results of SPI project are continuously reviewed, analyzed and reflected on in SPI action plan
How significant is this item for the success of this project?  High □  Medium □  Low □

9.6: SPI measures (metrics and quality indicators) are regularly reviewed and modified if needed
How significant is this item for the success of this project?  High □  Medium □  Low □
10. Management commitment

☐ 10.1: Management is engaged and committed to SPI project
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 10.1.1: High level management (including project sponsor(s)) is engaged and committed to SPI project
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 10.1.2: Middle level management is engaged and committed to SPI project, if any
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 10.1.3: Low level management is engaged and committed to SPI project, if any
   How significant is this item for the success of this project?  High □  Medium □  Low □

11. Stakeholders’ attitude

☐ 11.1: Key stakeholders are highly committed to SPI project
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 11.2: Stakeholders accept planned SPI activities
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 11.3: Stakeholders follow defined software process
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 11.4: Stakeholders, who run a software process, have partial ownership over it
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 11.5: Stakeholders trust and respect SPI managers
   How significant is this item for the success of this project?  High □  Medium □  Low □

☐ 11.6: Stakeholders participate in SPI project
   How significant is this item for the success of this project?  High □  Medium □  Low □
Appendix A.9

Questionnaire for evaluation of the SMA model

Introductory questions
1. What is your name?
2. What company do you work for?
3. What is your role in the company?

Main interview questions
1. Receive general information about the selected method adoption project
2. Present the overview of the SMA model.
3. For each stage of the model:
   3.1 Present the stage of the SMA model.
   3.2 For each phase of the presented stage:
      i. Present the phase of the SMA model.
      ii. For each activity of the presented phase:
         (1) Present the activity.
         (2) Did you perform this activity during the adoption of new method in your organization?
            • If no, would your organization benefit from performing this activity?
              a. If no, why? Please motivate your answer.
            • If yes, was this activity necessary or useful?
              a. If no, why? Please motivate your answer.
      iii. Are there any activities missing in this phase?
      iv. Are there any unnecessary or duplicated activities in this phase?
      v. Are there any activities in this phase that have problems with formulation?
3.3 Are there any phases missing in this stage?
3.4 Are there any unnecessary or duplicated phases?
4. What do you think about the structure of the model? Do you have any suggestions for improvements?
5. Present the overview of the contextual factors. For each contextual factor:
   5.1 Do you think this factor is useful for guiding organizations that plan to adopt new software method?
6. Do you think this model can be useful for other organizations that plan to adopt new software method?
### Appendix B

#### Table B.1  Organizational SPI success and sustainability factors and their sources

<table>
<thead>
<tr>
<th>Cluster</th>
<th>SPI success factor</th>
<th>Literature</th>
<th>Sources</th>
<th>No of interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support of SPI</td>
<td>1. Management continuously supports and commits to SPI process</td>
<td>(Goldenson &amp; Herbsleb, 1995), (El Emam et al., 1999), (Stelzer &amp; Mellis, 1998), (Rainer &amp; Hall, 2002), (Niazi et al., 2006b), (Varkoi, 2002), (Curtis &amp; Paulk, 1993), (Dyba, 2000), (Dyba, 2005), (Sulayman et al., 2012), (Hall et al., 2002), (Paulish &amp; Carleton, 1994), (Nazir et al., 2008), (Dangle et al., 2005), (Basri &amp; O’Connor, 2010), (Niazi et al., 2006a), (Niazi et al., 2010), (Sweeney &amp; Bustard, 1997)</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>2. Resources are dedicated to SPI</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a) (Goldenson &amp; Herbsleb, 1995), (El Emam et al., 1999), (Curtis &amp; Paulk, 1993), (Sulayman et al., 2012), (Beecham et al., 2003), (Paulish &amp; Carleton, 1994), (Nazir et al., 2008), (Dangle et al., 2005), (Niazi et al., 2006a), (Niazi et al., 2010), (Sweeney &amp; Bustard, 1997)</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>3. SPI responsibilities are clearly specified and compensated</td>
<td>(Goldenson &amp; Herbsleb, 1995), (El Emam et al., 1999), (Varkoi, 2002), (Curtis &amp; Paulk, 1993), (Sulayman et al., 2012), (Beecham et al., 2003), (Sweeney &amp; Bustard, 1997)</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>4. Competent external SPI leaders are designated</td>
<td>(Goldenson &amp; Herbsleb, 1995), (El Emam et al., 1999), (Stelzer &amp; Mellis, 1998), (Niazi et al., 2006b), (Varkoi, 2002), (Curtis &amp; Paulk, 1993), (Dyba, 2000), (Dyba, 2005), (Hall et al., 2002), (Paulish &amp; Carleton, 1994), (Sweeney &amp; Bustard, 1997) (Ekdahl &amp; Larsson, 2006)</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>5. Internal SPI leaders are designated</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Stelzer &amp; Mellis, 1998), (Rainer &amp; Hall, 2002) (Rainer &amp; Hall, 2002), (Varkoi, 2002), (Dyba, 2000), (Dyba, 2005), (Hall et al., 2002), (Paulish &amp; Carleton, 1994), (Nazir et al., 2008), (Basri &amp; O’Connor, 2010), (Sweeney &amp; Bustard, 1997)</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Resources</td>
<td>6. The level of technical staff turnover is low</td>
<td>-</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Alignment</td>
<td>7. SPI is aligned with business goals</td>
<td>(Dyba, 2000), (Dyba, 2005), (Hall et al., 2002), (Nazir et al., 2008), (Dangle et al., 2005)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Alignment</td>
<td>8. SPI is aligned with organizational policies and strategies</td>
<td>(Goldenson &amp; Herbsleb, 1995), (Sulayman et al., 2012), (Beecham et al., 2003), (Nazir et al., 2008), (Niazi et al., 2010)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Alignment</td>
<td>9. SPI methods are tailored to specific organizational contexts and needs</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Stelzer &amp; Mellis, 1998), (Sulayman et al., 2012), (Hall et al., 2002), (Dangle et al., 2005)</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Cluster</td>
<td>SPI sustainability success factor</td>
<td>Sources</td>
<td>No of interviews</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------</td>
<td>---------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>Education and knowledge</td>
<td>1. Stakeholders are trained in software process</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Goldenson &amp; Hersbleb, 1995), (Rainer &amp; Hall, 2002), (Niazi, et al., 2006b), (Curtis &amp; Paulk, 1993), (Hall, et al., 2002), (Beecham, et al., 2003), (Paulish &amp; Carleton, 1994), (Nazir, et al., 2008), (Dangle, et al., 2005), (Basri &amp; O'Connor, 2010), (Niazi, et al., 2006a), (Ekdahl &amp; Larsson, 2006)</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Stakeholders are continuously mentored and coached</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Goldenson &amp; Hersbleb, 1995), (Rainer &amp; Hall, 2002), (Niazi, et al., 2006b), (Niazi, et al., 2006a)</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. SPI leaders possess experience and expertise in SPI</td>
<td>(Rainer &amp; Hall, 2002), (Niazi, et al., 2006b), (Hall, et al., 2002), (Beecham, et al., 2003), (Dangle, et al., 2005), (Niazi, et al., 2006a), (Niazi, et al., 2010), (Sweeney &amp; Bustard, 1997), (Ekdahl &amp; Larsson, 2006)</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>SPI strategy</td>
<td>4. SPI goals and objectives are clear and realistic</td>
<td>(Goldenson &amp; Hersbleb, 1995), (El Emam, et al., 1999), (Steizer &amp; Mellis, 1998), (Dyba, 2000), (Dyba, 2005), (Sulayman, et al., 2012), (Hall, et al., 2002), (Beecham, et al., 2003), (Nazir, et al., 2008), (Dangle, et al., 2005)</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. SPI method is well defined</td>
<td>(Rainer &amp; Hall, 2002), (Niazi, et al., 2006b), (Sulayman, et al., 2012), (Hall, et al., 2002), (Paulish &amp; Carleton, 1994), (Basri &amp; O'Connor, 2010), (Niazi, et al., 2006a), (Niazi, et al., 2010)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SPI management and execution</td>
<td>6. SPI project is effectively managed</td>
<td>(Steizer &amp; Mellis, 1998), (Niazi, et al., 2006b), (Beecham, et al., 2003), (Niazi, et al., 2006a), (Niazi, et al., 2010), (Ekdahl &amp; Larsson, 2006), (Sweeney &amp; Bustard, 1997)</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Process improvements are focused on specific areas</td>
<td>(Paulish &amp; Carleton, 1994), (Nazir, et al., 2008), (Dangle, et al., 2005), (Sweeney &amp; Bustard, 1997)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. Process improvement effort is flexible</td>
<td>-</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. Information about SPI activities and its results is disseminated</td>
<td>(Steizer &amp; Mellis, 1998), (Niazi, et al., 2006b), (Varkoi, 2002), (Dyba, 2000), (Dyba, 2005), (Sulayman, et al., 2012), (Beecham, et al., 2003), (Dangle, et al., 2005), (Niazi, et al., 2010)</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. Process standards are defined and enforced</td>
<td>-</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11. SPI effort brings positive results</td>
<td>-</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Continuity of SPI effort</td>
<td>12. Software process is monitored and measured</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Steizer &amp; Mellis, 1998), (Rainer &amp; Hall, 2002), (Dyba, 2000), (Dyba, 2005), (Sulayman, et al., 2012), (Paulish &amp; Carleton, 1994), (Nazir, et al., 2008), (Sweeney &amp; Bustard, 1997)</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13. Software process and its efficiency are continuously reviewed</td>
<td>(Chakravorty, 2010), (Nikitina &amp; Kajko-Mattsson, 2011a), (Rainer &amp; Hall, 2002), (Varkoi, 2002), (Dyba, 2000), (Dyba, 2005), (Dangle, et al., 2005), (Basri &amp; O'Connor, 2010), (Niazi, et al., 2006a), (Aaen, et al., 2001)</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14. SPI effort is continuous</td>
<td>(Chakravorty, 2010), (Curtis &amp; Paulk, 1993), (Aaen, et al., 2001), (Sweeney &amp; Bustard, 1997)</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>
### Table B.3  Social SPI success and sustainability factors and their sources

<table>
<thead>
<tr>
<th>Cluster</th>
<th>SPI sustainability success factor</th>
<th>Literature</th>
<th>Sources</th>
<th>No of interviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding and awareness of SPI</td>
<td>1. Stakeholders have a common understanding of the process undergoing change</td>
<td>(Stelzer &amp; Mellis, 1998), (Curtis &amp; Paulk, 1993)</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>2. Stakeholders are aware of complexity, challenges and benefits of SPI</td>
<td>(Niazi, et al., 2006b), (Varkoi, 2002), (Sulayman, et al., 2012), (Paulish &amp; Carleton, 1994), (Basri &amp; O’Connor, 2010), (Niazi, et al., 2006a), (Niazi, et al., 2010)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Stakeholders have realistic expectations</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Attitude to SPI</td>
<td>4. Technical staff accepts SPI activities</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Hall, et al., 2002), (Sweeney &amp; Bustard, 1997)</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>5. Technical staff is committed to the SPI process</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Nazir, et al., 2008), (Dangle, et al., 2005), (Basri &amp; O’Connor, 2010)</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Facilitation of SPI</td>
<td>6. Stakeholders are being encouraged to support SPI</td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>7. Technical staff is rewarded for contribution to SPI success</td>
<td>(Rainer &amp; Hall, 2002), (Niazi, et al., 2006b)</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>8. SPI leaders encourage initiative and openness of stakeholders</td>
<td>(Curtis &amp; Paulk, 1993), (Dyba, 2000), (Dyba, 2005), (Beecham, et al., 2003)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>9. Technical staff participates in SPI</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (Goldenson &amp; Herbsleb, 1995), (El Emam, et al., 1999), (Stelzer &amp; Mellis, 1998), (Niazi, et al., 2006b), (Varkoi, 2002), (Dyba, 2000), (Dyba, 2005), (Hall, et al., 2002), (Nazir, et al., 2008), (Basri &amp; O’Connor, 2010), (Niazi, et al., 2006a), (Sweeney &amp; Bustard, 1997), (Ek Dahl &amp; Larsson, 2006)</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. Technical staff owns the software process</td>
<td>(Nikitina &amp; Kajko-Mattsson, 2011a), (El Emam, et al., 1999), (Rainer &amp; Hall, 2002), (Hall, et al., 2002), (Sweeney &amp; Bustard, 1997)</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>
### The SMA model evaluation results

#### Table C.1  Results of the evaluation of the SMA model

(The meaning of the symbols used in the table: “+” stands for this activity was performed, “−” stands for this activity was not performed, “P” – stands for this activity was partly performed) (to be continued)

<table>
<thead>
<tr>
<th>Stage/Phase/Activity</th>
<th>Evaluated SMA projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method adoption stage</td>
<td></td>
</tr>
<tr>
<td>Initiation phase</td>
<td></td>
</tr>
<tr>
<td>I1: Identify needs for software process change</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>I2: Set goals for software process change</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>I3: Set the scope for method adoption project</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>I4: Get management support for method adoption project</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Assessment phase</td>
<td></td>
</tr>
<tr>
<td>A1: Assess and measure current software process</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>A2: Examine organizational needs</td>
<td>- + + - -</td>
</tr>
<tr>
<td>A3: Examine current organizational culture</td>
<td>- - - - P +</td>
</tr>
<tr>
<td>A4: Identify process related problems and challenges</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>A5: Design suitable process practices</td>
<td>- + + - -</td>
</tr>
<tr>
<td>A6: Define desired organizational culture</td>
<td>- - - - P +</td>
</tr>
<tr>
<td>A7: Design/proposal software method to be adopted</td>
<td>- - - - - +</td>
</tr>
<tr>
<td>A8: Evaluate benefits and cost of the proposed method</td>
<td>- - - - - +</td>
</tr>
<tr>
<td>A9: Suggest improvements to the proposed method</td>
<td>- - - - - +</td>
</tr>
<tr>
<td>Planning phase</td>
<td></td>
</tr>
<tr>
<td>Pl1: Identify stakeholders</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl2: Identify gaps between current and desired state</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl3: Assess stakeholders' knowledge of new method</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl4: Assess attitude towards new method adoption</td>
<td>P P P P P +</td>
</tr>
<tr>
<td>Pl5: Determine changes to the organizational culture</td>
<td>P P P P P -</td>
</tr>
<tr>
<td>Pl6: Determine and prioritize process changes</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl7: Design training program</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl8: Determine change strategy</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl9: Identify risks involved</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl10: Create action plan</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl11: Determine budget and resources required</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl12: Request sponsorship and resources</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pl13: Announce method adoption project and its purpose</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Preparation phase</td>
<td></td>
</tr>
<tr>
<td>Pr1: Assign manager/team for method adoption project</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pr2: Train/coach SMA manager/team</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pr3: Determine communication approach</td>
<td>- + P + + +</td>
</tr>
<tr>
<td>Pr4: Determine level of stakeholders’ involvement</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pr5: Announce the impact and strategy of method adoption project</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pr6: Train the stakeholders in the new method</td>
<td>+ + + + + +</td>
</tr>
<tr>
<td>Pr7: Encourage stakeholders in the new method</td>
<td>P P P P P +</td>
</tr>
</tbody>
</table>
Table C.1  Results of the evaluation of the SMA model (continued)

<table>
<thead>
<tr>
<th>Stage/Phase/Activity</th>
<th>Evaluated method adoption projects</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1: Identify process changes;</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>D2: Announce the process changes and their impact</td>
<td></td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D3: Assign roles responsible for process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>D4: Prepare/coach stakeholders in process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>D5: Prepare infrastructure for process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D6: Train stakeholders in the new domain knowledge</td>
<td></td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D7: Implement the process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>D8: Prepare new process control mechanisms</td>
<td></td>
<td>-</td>
<td>*</td>
<td>P</td>
<td>*</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D9: Adjust organizational culture</td>
<td></td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D10: Track performance of the changed process - <em>New</em></td>
<td></td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>D11: Encourage stakeholders to follow the process</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>D12: Update the action plan</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Continuous improvement stage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process review phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1: Review and measure current software process</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R2: Identify and prioritize impediments</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R3: Record and distribute lessons learned</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R4: Propose potential process improvements</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R5: Propose improvements of the Continuous improvement stage</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R6: Identify and prioritize next set of process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R7: Plan implementation of the process changes</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>R8: Update the action plan</td>
<td></td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Deployment phase (see Deployment phase in the Method adoption stage)
<table>
<thead>
<tr>
<th>Checklist items</th>
<th>Weights for SPI projects</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. SPI goals and scope</strong></td>
<td></td>
</tr>
<tr>
<td>1.1: SPI project has a business case that shows its ROI</td>
<td>0.2938 0.3203 -</td>
</tr>
<tr>
<td>1.2: SPI vision and main goals are well-defined and disseminated to key stakeholders</td>
<td>0.0417 0.3203 0.0000</td>
</tr>
<tr>
<td>1.3: SPI vision and main goals are stable and realistic</td>
<td>0.0000 0.3203 -</td>
</tr>
<tr>
<td>1.4: Key stakeholders reach consensus on SPI vision and main goals</td>
<td>0.9325 0.9784 0.8810</td>
</tr>
<tr>
<td>1.5: Scope of SPI project is clearly bounded</td>
<td>- - -</td>
</tr>
<tr>
<td>1.6: Scope of SPI project is continuously evaluated and modified when needed</td>
<td>- - -</td>
</tr>
<tr>
<td><strong>2. Sponsorship and resources</strong></td>
<td></td>
</tr>
<tr>
<td>2.1: Sponsorship and resources for SPI are continuously provided</td>
<td>0.5045 0.6200 -</td>
</tr>
<tr>
<td>2.1.1: Budget &amp; resource for initiating, planning and preparing SPI activities are assigned and provided</td>
<td>0.5045 0.6200 -</td>
</tr>
<tr>
<td>2.1.2: Budget &amp; resource for deploying SPI activities are assigned and provided</td>
<td>0.5045 0.6200 -</td>
</tr>
<tr>
<td>2.1.3: Budget &amp; resource for sustaining gains achieved by SPI are assigned and provided</td>
<td>0.6623 0.4509 0.8810</td>
</tr>
<tr>
<td>2.2: Amount of provided resources is aligned with scope, complexity and novelty of SPI project</td>
<td>0.6623 0.4509 0.8810</td>
</tr>
<tr>
<td><strong>3. SPI roles and responsibilities</strong></td>
<td></td>
</tr>
<tr>
<td>3.1: Accountability for processes undergoing change is assigned</td>
<td>0.9325 0.9784 0.8810</td>
</tr>
<tr>
<td>3.2: Accountability for SPI project is assigned</td>
<td>0.3789 0.4390 -</td>
</tr>
<tr>
<td>3.3: Responsibilities for leading local improvements are defined and assigned</td>
<td>0.3494 0.4509 0.1667</td>
</tr>
<tr>
<td>3.4: Responsibilities for deploying SPI activities are defined and assigned</td>
<td>0.5212 0.6200 0.3208</td>
</tr>
<tr>
<td>3.5: Responsibilities for maintenance and support of SPI activities are defined and assigned</td>
<td>0.6623 0.4509 0.8810</td>
</tr>
<tr>
<td>3.6: SPI responsibilities are assigned, understood and agreed upon by the stakeholders</td>
<td>0.2712 0.2519 0.3095</td>
</tr>
<tr>
<td><strong>4. Stakeholder management</strong></td>
<td></td>
</tr>
<tr>
<td>4.1: Stakeholders and key stakeholders are clearly identified</td>
<td>0.6885 0.9784 -</td>
</tr>
<tr>
<td>4.2: Stakeholders' support and readiness to SPI are correctly understood and considered in SPI method and change strategy</td>
<td>0.0167 0.3203 -</td>
</tr>
<tr>
<td>4.3: Stakeholders are continuously trained, coached and mentored in newly deployed processes, according to their training needs</td>
<td>0.6216 0.4509 0.7857</td>
</tr>
<tr>
<td>4.4: Stakeholders are encouraged to support and commit to SPI project</td>
<td>0.9325 0.9784 0.8810</td>
</tr>
<tr>
<td>4.5: Communication approach is established and it is agreed upon who receives communications, when, how and to what level of detail</td>
<td>0.3285 0.0000 0.8810</td>
</tr>
<tr>
<td>4.6: Stakeholders, who get affected by an SPI activity, are informed about its purpose and impact</td>
<td>0.3192 0.3203 0.3208</td>
</tr>
<tr>
<td>4.7: Stakeholders, who get affected by an SPI activity, are prepared for its deployment</td>
<td>0.7196 0.9784 0.3208</td>
</tr>
<tr>
<td>4.8: Stakeholders are informed about complexity, challenges and benefits of SPI project</td>
<td>0.6623 0.9784 0.3095</td>
</tr>
<tr>
<td>4.9: Stakeholders have realistic expectations from SPI project</td>
<td>0.6216 0.9784 0.2624</td>
</tr>
<tr>
<td><strong>5. Competence and knowledge</strong></td>
<td></td>
</tr>
<tr>
<td>5.1: Stakeholders possess necessary technical skills and domain knowledge to run the deployed process</td>
<td>0.6216 0.4509 0.7857</td>
</tr>
<tr>
<td>5.2: Stakeholders possess knowledge about processes undergoing change and process to be deployed</td>
<td>- - -</td>
</tr>
<tr>
<td>5.3: SPI manager has knowledge, leadership competence and personal profile to lead SPI project</td>
<td>- - -</td>
</tr>
<tr>
<td>5.4: SPI manager is engaged in and passionate about the SPI project</td>
<td>- - -</td>
</tr>
<tr>
<td>5.5: Competence of SPI manager is consistent with size, complexity, criticality and novelty of SPI project</td>
<td>- - -</td>
</tr>
<tr>
<td><strong>6. Preparation for SPI</strong></td>
<td></td>
</tr>
<tr>
<td>6.1: Organizational culture and competence are analyzed and considered in SPI method and change strategy</td>
<td>0.6885 - 0.7582</td>
</tr>
<tr>
<td>6.2: Processes undergoing change and SPI goals are correctly understood and considered in SPI method and change strategy</td>
<td>0.3789 - 0.7582</td>
</tr>
<tr>
<td>6.3: Similar process improvements were previously implemented on another SPI project(s)</td>
<td>0.4798 0.2519 0.7857</td>
</tr>
<tr>
<td>Checklist items</td>
<td>Weights for SPI projects</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>all</td>
</tr>
<tr>
<td>7: Deployment management</td>
<td></td>
</tr>
<tr>
<td>7.1: SPI method and change strategy are defined</td>
<td>0.1982</td>
</tr>
<tr>
<td>7.2: SPI method and change strategy are tailored to fit organization, its process and stakeholders</td>
<td>0.1982</td>
</tr>
<tr>
<td>7.3: Defined SPI method and change strategy are properly followed</td>
<td>0.1982</td>
</tr>
<tr>
<td>7.4: SPI action plan is created</td>
<td>0.2171</td>
</tr>
<tr>
<td>7.5: SPI schedule is created based on sensible estimates and planning</td>
<td>0.0167</td>
</tr>
<tr>
<td>7.6: Impact of each SPI activity is carefully analyzed before it is being deployed</td>
<td>0.2964</td>
</tr>
<tr>
<td>7.7: SPI activities are deployed according to SPI action plan and schedule</td>
<td>0.9325</td>
</tr>
<tr>
<td>7.8: Impediments blocking SPI activities are continuously identified and removed</td>
<td>0.9325</td>
</tr>
<tr>
<td>7.9: SPI action plan and schedule are continuously reviewed and modified, if needed</td>
<td>0.1477</td>
</tr>
<tr>
<td>7.10: Necessary tools to support SPI project are adequate and available</td>
<td>-</td>
</tr>
<tr>
<td>8: SPI project governance and support</td>
<td></td>
</tr>
<tr>
<td>8.1: Governance of SPI project is consistent with its value, size, criticality and risks</td>
<td>0.7196</td>
</tr>
<tr>
<td>8.2: SPI risks are identified, analyzed and planned for</td>
<td>0.5658</td>
</tr>
<tr>
<td>8.3: SPI risks are continuously monitored</td>
<td>0.9568</td>
</tr>
<tr>
<td>8.4: Deployed SPI activities are institutionalized and supported by organizational policies</td>
<td>0.8623</td>
</tr>
<tr>
<td>8.5: Essential mechanisms to control adherence to the deployed process are established and continuously improved</td>
<td>0.5212</td>
</tr>
<tr>
<td>9: SPI project monitoring and measurement</td>
<td></td>
</tr>
<tr>
<td>9.1: SPI objectives are measurable by carefully designed SPI measures (metrics and quality indicators)</td>
<td>0.3285</td>
</tr>
<tr>
<td>9.2: General SPI project milestones and reporting mechanisms are defined and established</td>
<td>0.5684</td>
</tr>
<tr>
<td>9.3: SPI metrics are consistently collected</td>
<td>0.4215</td>
</tr>
<tr>
<td>9.4: Progress of SPI project is continuously monitored, analyzed, and communicated</td>
<td>0.5770</td>
</tr>
<tr>
<td>9.5: Results of SPI project are continuously reviewed, analyzed and reflected on in SPI action plan</td>
<td>0.9325</td>
</tr>
<tr>
<td>9.6: SPI measures (metrics and quality indicators) are regularly reviewed and modified, if needed</td>
<td>-</td>
</tr>
<tr>
<td>10: Management commitment</td>
<td></td>
</tr>
<tr>
<td>10.1: High level management (including project sponsor(s)) is engaged and committed to SPI project</td>
<td>0.6885</td>
</tr>
<tr>
<td>10.2: Middle level management is engaged and committed to SPI project, if any</td>
<td>0.1082</td>
</tr>
<tr>
<td>10.3: Low level management is engaged and committed to SPI project, if any</td>
<td>0.0894</td>
</tr>
<tr>
<td>11: Stakeholders’ attitude</td>
<td></td>
</tr>
<tr>
<td>11.1: Key stakeholders are highly committed to SPI project</td>
<td>0.7196</td>
</tr>
<tr>
<td>11.2: Stakeholders accept planned SPI activities</td>
<td>0.9325</td>
</tr>
<tr>
<td>11.3: Stakeholders follow defined software process</td>
<td>-</td>
</tr>
<tr>
<td>11.4: Stakeholders, who run a software process, have partial ownership of it</td>
<td>0.5770</td>
</tr>
<tr>
<td>11.5: Stakeholders trust and respect SPI managers</td>
<td>0.5904</td>
</tr>
<tr>
<td>11.6: Stakeholders participate in SPI project</td>
<td>0.7396</td>
</tr>
</tbody>
</table>
Part II

Publications
List of publications

Paper 1. Impact of Growing Business on Software Processes

Paper 2. Impact of Corporate and Organic Growth on Software Development

Paper 3. Historical Perspective of Two Process Transitions

Paper 4. Developer-driven Big-bang Process Transition from Scrum to Kanban

Paper 5. From Scrum to Scrumban: A Case Study of a Process Transition

Paper 6. Factors Leading to the Success and Sustainability of Software Process Improvement Efforts

Paper 7. Software Process Improvement Health Checklist

Paper 8. Successful Process Improvement Projects are no Accidents


Natalja Nikitina, Mira Kajko-Mattsson. Submitted to ICSSP 2014