Experience driven software process assessment and improvement

MARTIN IVARSSON

Department of Computer Science & Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010
Experience driven software process assessment and improvement
MARTIN IVARSSON

© Copyright Martin Ivarsson, 2010.

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie 3079
ISSN 0346-718X

Technical Report No. 68D
Department of Computer Science and Engineering
Research group: Applied Software Engineering

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31-772 1000

Contact information:
Department of Computer Science and Engineering
Chalmers University of Technology
Hörsalsvägen 11
Telephone: +46 31 772 57 15
SE-412 96 Göteborg, Sweden
martin.ivarsson@chalmers.se

Chalmers Reproservice
Göteborg, Sweden 2010
Abstract

To maintain and increase competitive advantages, software organizations must continually strive to refine and improve their development processes. Software process assessment and improvement is the means by which development organizations assure and improve their processes, practices and tools to support their specific needs. Development practices are ultimately based on the knowledge and experiences of the organization’s staff. However, relying only on experiences inherent to the organization might, in the long run, mean becoming increasingly good at using obsolete practices. Thus, organizations also need to take advantage of industry best practice, innovations and, knowledge and experience inherent to the organization for improvements.

This thesis presents research aimed at supporting organizations to assess and improve their processes and practices with a focus on requirements engineering. First, the validation of a framework for assessment and improvement planning enables identification of improvement issues. The framework is characterized by relative accuracy and resource efficiency in using multiple data sources and involving practitioners in identifying what needs to be improved first. When challenges have been uncovered, the decision for how to improve involves: looking for innovation by introducing new practices, or improve practices currently used in the organization.

To support process innovation, an investigation of the level and quality of evidence in requirements engineering research was made. Technology evaluations are classified according to the level of industrial relevance and rigor. To support learning from experiences inherent to the organization, a framework for diffusing and improving practices was developed and evaluated. Practices are used to transfer knowledge between different parts of the organization. Experiences from practice usage are then collected in postmortems offering iteratively improved decisions support. As the validity of experiences collected can be questioned, an example of how to strengthen the evidence is presented. The results can be stored in relation to the practice to improve the decision support related to it.

The main contribution of the thesis is the development and evaluation of methods for process assessment and improvement based on experience and knowledge inherent to an organization. Another contribution is the investigation into state-of-research in requirements engineering. It is important to leverage experiences both internal and external to an organization when making decisions on what practices to use. Future work should investigate how to balance process and practice innovation and refinement.

Keywords: Software Engineering, Software Process Assessment, Software Process Improvement, Requirements Engineering
# Table of Contents

ABSTRACT .................................................................................................................................................... V

TABLE OF CONTENTS ............................................................................................................................... VII

ACKNOWLEDGMENTS ..................................................................................................................................... IX

LIST OF APPENDED PAPERS .................................................................................................................. XI

RELATED PUBLICATIONS ........................................................................................................................... XII

CHAPTER I: INTRODUCTION ...................................................................................................................... 1

1. INTRODUCTION ................................................................................................................................. 1
   1.1 GUIDE TO THE THESIS ............................................................................................................. 2

2. BACKGROUND AND RELATED WORK .............................................................................................. 3
   2.1 SOFTWARE PROCESS ASSESSMENT AND IMPROVEMENT .................................................... 3
   2.2 REQUIREMENTS ENGINEERING ............................................................................................. 11
   2.3 KNOWLEDGE MANAGEMENT IN SOFTWARE ENGINEERING .................................................. 14

3. RESEARCH APPROACH ...................................................................................................................... 17
   3.1 RESEARCH CONTEXT .................................................................................................................. 18
   3.2 RESEARCH PROCESS .................................................................................................................. 18
   3.3 RESEARCH QUESTIONS ............................................................................................................ 23
   3.4 RESEARCH METHODOLOGY ................................................................................................... 24

4. RESEARCH RESULTS ............................................................................................................................ 27
   4.1 CONTRIBUTIONS ....................................................................................................................... 28
   4.2 FUTURE WORK ............................................................................................................................ 31

5. LIST OF PAPERS .................................................................................................................................. 31
   5.1 INCLUDED PAPERS ................................................................................................................... 31
   5.2 STATEMENT OF CONTRIBUTION ............................................................................................ 34
   5.3 PAPERS NOT INCLUDED .......................................................................................................... 34

REFERENCES ............................................................................................................................................. 35

CHAPTER II: A PRACTITIONER’S GUIDE TO LIGHT WEIGHT SOFTWARE PROCESS
ASSESSMENT AND IMPROVEMENT ........................................................................................................ 45

CHAPTER III: TECHNOLOGY TRANSFER DECISION SUPPORT IN REQUIREMENTS
ENGINEERING RESEARCH: A SYSTEMATIC REVIEW OF REJ .......................................................... 89

CHAPTER IV: A METHOD FOR EVALUATING RIGOR AND INDUSTRIAL RELEVANCE
OF TECHNOLOGY EVALUATIONS ........................................................................................................ 123

CHAPTER V: PRACTICE SELECTION FRAMEWORK ............................................................................. 157

CHAPTER VI: TOOL SUPPORT FOR DISSEMINATING AND IMPROVING
DEVELOPMENT PRACTICES ..................................................................................................................... 197

CHAPTER VII: AN EMPIRICAL QUALITY ASSESSMENT OF AUTOMOTIVE USE CASES
................................................................................................................................................................... 227
Acknowledgements

This thesis and the included research have been achieved with the help and support of many people. I would like to take this opportunity to pay my gratitude.

I would like to thank my supervisor, Dr. Dr. (Master) Tony Gorschek, for having the absent mind to adopt me as a Padawan learner, and for the energy and stamina to see it through until the end. Most would have cut their losses and run.

“The force is kinda weak with you, my young Padawan”- Master Tony

My understanding of software engineering and research would have been very different if it was not for you (not worse – but different). On a more serious note, this thesis would not have been possible without your support. Thanks!

I would like to thank my examiner Prof. Johan Karlsson for his support and encouragement. I would also like to thank Dr. Peter Öhman for providing me with this opportunity, supervision and for approval of travel expenses. I would like to thank my co-authors, Fredrik Pettersson, Fredrik Törner for their contribution to the research presented in this thesis. The CEDES research project at Chalmers, and all the Ph.D. students who are or have been involved, for providing a enjoyable working environment and giving me a broader understanding of current research not only my own area but also in theirs.

I would also like to thank all persons that have been involved in the research studies. In particular, I would like to thank David Hellström, Fredrik Bernin, Lic. Ulrik Eklund, Johan Haraldsson, Anders Magnusson and Dr. Martin Hiller for providing me with the opportunity to perform research in industry and for your support.

Finally, I would like to thank my family for their continuous support.

To all of you, and the ones I forgot – Thank you!

This research has been funded by the Swedish industry and government joint research program IVSS – Intelligent Vehicle Safety Systems through the CEDES project.
Chapter I

Introduction

Experience driven software process assessment and improvement
1. Introduction

Software continues to become more important for a number of industries. Software is the predominant innovation driver in many domains, e.g. automotive and consumer electronics such as TVs, digital cameras and mobile phones [1]. However, the flexible and intangible nature of software has caused software projects to be overrepresented in project failures. In software product development the activity of eliciting, analyzing, specifying and validating requirements is called requirements engineering (RE). It is rather well understood that RE has a high impact on final product quality and constitutes a major determinant in software project failure [2-10].

To maintain and increase competitive advantages, software organizations must continually strive to refine and improve their development practices¹. Process assessment and improvement is the means by which development organizations assure and improve their processes and tools to support their specific needs. Process assessment assures that development processes meet quality requirements, and identifies improvement potential. Assessment can be performed in different ways, for instance by benchmarking against a set of best practices or by using measurements. The goal is to identify an organization’s strengths and weaknesses to establish a plan for what to improve. One potential issue in starting an improvement effort is that extensive software process improvement (SPI) frameworks and measurement programs are viewed by many as too large to implement [11-14]. This makes it difficult for organizations to initiate and perform assessment and improvement efforts, as cost and time are crucial considerations [11, 12, 15].

When weaknesses have been identified, it is important to establish how to alleviate them to improve the performance of the organization. The decision for how to improve involves: looking for innovation by introducing new processes and practices, or improve ones currently used in the organization. Aiming at innovation by e.g. introducing new practices into the organization is high-risk, and the fit can be hard to gauge [16, 17]. Software organizations’ development practices are ultimately based on knowledge and competence of their staff [18]. Managing and sharing existing knowledge is thus a promising approach to improving the organization. Different parts of the organization have different knowledge and experiences, and might thus use different practices. Practices already used in the organization offer a low-risk alternative for improvement, as experiences from using them already exist in the organization [19]. However, focusing only on experiences inherent to the organization might, in the long run, mean becoming increasingly good at using obsolete practices [19]. Thus, organizations need to take advantage of industry best practice, innovations and, knowledge and experience inherent to the organization for improvements.

Development projects are concerned with fulfilling predetermined requirements on time, on budget and with quality, and consequently improvement often only happens as a reaction to a problem [20]. Even though knowledge of how to improve may exist in the organization, product development often takes precedence over improvements - i.e., short-term gains in terms of implementing new features outweigh long-term goals in terms of realized improvements [21].

¹ In this thesis, the terms practice and technology are used interchangeably. The terms refers to any technique, model, procedure, method and tool used in software development and maintenance.
In addition, focusing only on weaknesses might mean missing important opportunities for improvement [22, 23]. Managing strengths, weaknesses and means to improve development practices on the organizational level can help avoid missing important improvement opportunities. In addition, managing experiences outside of project contexts can help the organization to avoid localized optimizations that might deteriorate the organization’s performance [20]. For instance, excessively adding features to a product in one project can be successful in that project but might render the architecture difficult to use in subsequent projects.

The main goal of the research presented in this thesis is to enable organizations to continuously improve to gain a competitive edge. By developing and applying lightweight methods for process assessment and improvement of the requirements engineering process, an organization’s performance is expected to improve. Four main contributions are made by the thesis. First, it provides a packaged framework for process assessment and improvement planning. Assessment is performed by eliciting improvement issues based on the organization’s experience and knowledge. The findings are validated through triangulation utilizing multiple data sources. In addition, practitioners are actively involved in prioritizing improvement issues and identifying dependences between them, in order to package improvements and thus establish an improvement plan that is realistic for the organization. Second, an evaluation and identification of evidence is given concerning the state-of-research on technology evaluations in requirements engineering. The evaluation considers both industrial relevance and rigor, and characterizes technology evaluations published in software engineering journals and how they have changed over time. The results can be used by practitioners to gauge the available evidence related to technologies, and by researchers as an empirical base on which to build on. Third, a framework is described for continuous process assessment and improvement through capturing experiences from development projects in postmortems. Experiences related to how practices have performed in the organization are captured and stored to make them available throughout the organization. Practices that have worked well are candidates for disseminating to other projects while practices that are estimated to be deficient can motivate closer investigation of the issues captured. Finally, as the validity of experiences collected in postmortems can be considered low, an example of how to strengthen the experiences is presented. Retrospective analysis is employed to evaluate a practice, Use case, used in industry. The results from the evaluation can be stored in relation to the practice in PSF to improve the available decision support for it.

1.1. Guide to the thesis

The thesis starts with an introduction to the research area and outlines how the research fit together. Following the introduction there are six thesis parts, each including one paper.

Chapter I provides a background to the research. Section 2 describes related work in the areas of software process improvement, requirements engineering and knowledge management to put the work in context. Section 3 outlines the research process and the main research questions addressed in this thesis, together with the methodology used to answer the research questions. The research results and contributions are given in Section 4 as well as an outline of future work. Finally, Section 5 lists the papers which are part of the thesis.
Chapter II presents a packaged framework for process assessment and improvement planning, iFLAP. The framework is applied at two departments at Volvo Technology to assess the requirements engineering process. The paper details these two cases of application and the lessons learned.

Chapter III describes an in-depth investigation into the available technology transfer decision support in the Requirements Engineering journal (REj). The investigation is performed as a comprehensive systematic literature review of all papers published in REj containing any type of technology evaluation.

Chapter IV presents a model for evaluating the relevance and rigor of technology evaluations in software engineering. The model is validated in a systematic literature review of requirements engineering technology evaluations published in software engineering journals.

Chapter V presents Practice Selection Framework (PSF), an Experience Factory approach, enabling lightweight experience capture and use by utilizing postmortem reviews. In addition, an initial evaluation of PSF in industry is presented.

Chapter VI introduces tool support for PSF. The tool is used to further evaluate PSF in both academia and industry.

Chapter VII presents a retrospective analysis of Use cases used in industry. The results from the analysis can be stored in PSF to improve decision support related to Use cases.

2. Background and related work

Software development is often viewed as comprising requirements engineering, design, implementation, testing and maintenance activities [24] as seen in Figure 1. The work presented in this thesis focuses on the requirements engineering process, and especially on process assessment and improvement of this process. Assessment and improvement are illustrated as the feedback cycle in Figure 1. Assessment identifies weaknesses and strength in projects, and changes are made to improve on the weaknesses. To understand the contents of this thesis, a short introduction is given to software process assessment and improvement in Section 2.1 and requirements engineering in Section 2.2. In addition, the work presented capitalizes on knowledge and experiences inherent to the organization for identifying and selecting improvements thus an introduction to knowledge management is given in Section 2.3.

![Figure 1: Overview of concepts used in this thesis.](image-url)

2.1. Software process improvement and assessment

To maintain and increase competitive advantages, software organizations must continually strive to refine and improve their development processes. A process, and in
particular a software development process, is defined by the IEEE Standard Glossary of Software Engineering Terminology [25] as:

“process. (1) A sequence of steps performed for a given purpose; for example, the software development process.”

“software development process. The 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.”

In this thesis, the definition of a process is assumed to include both the overall software development process and processes which are part of it, e.g. the requirement engineering process. Software process improvement and assessment are the means by which development organizations improve and assure their processes and tools to support their specific needs. Software process improvement (SPI) is often implicitly defined in the literature. Olson et al. [26] defines SPI as “the changes implemented to a software process that bring about improvements”. Other definitions often contain the stipulation that the changes should be improvements regarding process maturity or performance. Dybå [19] define two different improvement strategies that software organizations can adopt:

**Exploitation**
Exploitation entails learning from existing knowledge and experience and thereby improving existing technologies. This involves refining and elaborating on existing methods, processes and technologies, and is thus a low-risk approach to pursue improvements as no major changes are introduced into the organization.

**Exploration**
Exploration involves searching for new knowledge by either innovating new technologies or imitating existing ones. This involves experimentation with new methods, processes and technologies, and thus involves higher risks than exploitation as the results are more uncertain.

Process assessment, on the other hand, includes determining the current state of the organization and establishing what needs to be improved. Depending on what improvement strategy an organization has adopted, the assessment can take different forms. Several well-known SPI frameworks, presented below, used for process assessment and improvement have been defined. On the most fundamental level, all frameworks incorporate a continuous cycle consisting of two phases: analyze the process and change the process.

**Analyze phase**
The result of analyzing the process is an envisioned process that should be pursued. The process is then changed and the new, changed process becomes the current process. Analyzing the process consists of two steps: evaluation of the current situation and subsequent planning of what needs to be improved.
Change phase
Changing the process entails incorporating the needed changes that were identified in the earlier phase. This involves two steps: implementing the improvements identified and evaluating the effect of the improvements. If the effect is positive the new process becomes the current process and the cycle starts over.

The fundamental phases and the four steps presented can be visualized as in Figure 2, which was inspired by [27, 28].

![Diagram of the generic process for SPI.](image)

Figure 2: Generic process for SPI.

SPI frameworks are either bottom-up, inductive, or top-down, prescriptive. The next two sections describe the most well-known frameworks of both kinds.

2.1.1. Inductive frameworks
Inductive frameworks take their start in a bottom-up approach, basing the improvements on a thorough understanding of the current processes [29]. A well-known example of an inductive SPI framework is Basili’s Quality Improvement Paradigm (QIP) [30]. QIP proposes basing improvements on experiences from executing processes in projects rather than benchmarking against a pre-defined set of practices. QIP comprises two closed-loop cycles: a project cycle and an organizational cycle (see Figure 3). The project cycle provides feedback to the project executing the process, and the organizational cycle provides feedback to the organization when the project is completed. QIP begins with characterizing and understanding current processes. Based on this understanding, quantifiable goals for successful project performance are set, and changes to methods, techniques and tools are evaluated in pilot projects. The results of the pilot projects are then analyzed and if successful, are packaged and institutionalized in the organization.
2.1.2. Prescriptive frameworks

In contrast to inductive frameworks, prescriptive frameworks base improvements on the adherence of processes to a collection of best practices describing how software should be developed. The prescriptive nature of such frameworks means that there is one set of practices, a reference model, that is advocated. The reference model provides a description of the process entities that are to be evaluated. No special consideration is given to the specific organization’s needs, business goals or strategy other than that the process should adhere to the one prescribed by the framework.

2.1.2.1. Capability Maturity Model Integration

Capability Maturity Model Integration (CMMI) was developed to combine three models: Systems Engineering Capability Model (SECM), Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98, and the retired well-known Capability Maturity Model for Software (SW-CMM) [31]. CMMI replaced CMM when it was retired, and statements regarding CMM in the remainder of this introduction should also be considered valid for CMMI. CMM was originally developed to assist the US Department of Defense (DoD) in software acquisitions [32]. The rationale at the DoD was that it should be able to assess the development capability of contractors to minimize risks when acquiring software. As adherence to a specific maturity level became a business requirement for obtaining DoD contracts, contractors started using CMM as a guide to improve their development processes.

The CMMI reference model comprises 22 process areas, and enables organizations to approach SPI and appraisals by using two basic representations: staged and continuous. The staged representation provides a way to assess and improve the organization’s overall maturity level. CMMI has five maturity levels: initial, managed, defined, quantitatively managed, and optimizing. The level that an organization achieves is governed by which process areas are implemented in accordance with
CMMI (see Figure 4); each maturity level has its set of process areas that needs to be implemented. Thus, the staged representation of CMMI guides the order in which the process areas shall be implemented. The continuous representation comprises six capability levels, and offers organizations the option to focus on each individual process area. This provides the possibility to select an order in which to implement process areas that do not conform to the staged representation. However, CMMI still guides the order in which practices should be implemented in each process area.

The appraisal method used when assessing an organization with CMMI is called Standard CMMI Appraisal Method for Process Improvement (SCAMPI). The efforts required to assess an organization range from 20 – 60 person hours, for the least rigorous Class C appraisal, to 800 – 1600 person hours for the full and comprehensive Class A appraisal [33, 34]. It is only the full assessment that results in a maturity rating. It should also be noted that estimates stated pertain only to the time spent by the assessment team and not time spent by the staff being assessed.

CMMI v. 1.2

![CMMI Maturity Levels and Process Areas](image)

**Figure 4: CMMI maturity levels and process areas.**

### 2.1.2.2. ISO/IEC 15504
ISO/IEC 15504, Information technology – Process assessment [35], is an international standard for process assessment. During the development of the standard, the focus shifted from providing one reference model with which organizations are compared to including an effort to harmonize between different standards. Instead, ISO/IEC 15504 sets requirements on the reference models and process assessment models used. This has led to ISO/IEC 15504 relying on external sources for definitions of reference models. In particular, the life cycle models ISO 12207 [11], *Software Life Cycle Process*, and ISO 15288 [36], *Systems Life Cycle Process*, formally constitute the process reference model used in ISO/IEC 15504. This effectively leads to these...
standards being harmonized with ISO/IEC 15504. In addition, this has led to different communities developing reference models tailored for particular domains, with the development of such frameworks as Automotive SPICE [37] and SPICE for SPACE [38].

In contrast to CMMI, ISO/IEC 15504 provides only continuous representation, and the measurement framework used for capability determination of a process consists of six capability levels. The measure of capability is based upon a set of process attributes (PA) that describe a number of characteristics that should be achieved to obtain a particular capability level (see Figure 5, based on [39]).

ISO/IEC 15504 assessments are very similar to SCAMPI (CMMI Class A). A fundamental difference is that, while CMMI can use both internal (if trained) or external assessment group members, SPICE demands that an external assessor head the assessment [35]. The cost of performing SPICE assessments ranges from 33 to 824 person-hours (median of 110) [40].

ISO/IEC 15 504
Measurement framework

Process attributes

Figure 5: ISO/IEC 15504 capability levels and Process Attributes.

2.1.3. Concerns relating to SPI research

Even though there exist numerous examples of successful SPI in the literature, utilizing both inductive and prescriptive approaches, understanding how to implement SPI successfully is still one of the most challenging issues that the SPI field faces. Most of the case studies of SPI in the literature report on positive results, which is to be expected as organizations that have not shown any improvement or even regressed are unlikely to publish their results [40, 41]. A number of concerns relating to the application of current SPI practices have been identified in the literature, of which five related to this thesis are described and exemplified below together with their relation to
the frameworks presented in Sections 2.1.1 and 2.1.2. Most of the discussion will target prescriptive frameworks, as they are the most widely used in industry.

**Challenge 1: Focus of SPI**
The first challenge for software process improvement is the perspective taken in identifying, selecting and evaluating changes to the process. Process improvement often addresses the context of a development project. Process assessment is performed mostly on projects, and improvements are primarily assessed on projects [42]. Consequently, improvements often happen as a reaction to a problem and are confined to the project that is experiencing it. Addressing short-term goals and having a project perspective when implementing changes can raise problems. Especially for requirements engineering, project measures are only the first perspectives to consider [42]. Even if requirements contribute to a high-quality product and a project that is delivered on time and schedule, there is no guarantee that it will be successful on the market and generate revenue. Similarly for software architecture, an architecture might be successful within the project where it is developed but cause problems for subsequent projects using it. However, without fast feedback on improvements it can be hard to sustain commitment and discern whether the right direction has been taken, which means that measuring short-term is important but not enough. This means that some perspectives might need to be estimated by, for example, expert judgment to get early feedback on improvement success [43, 44]. When quantitative measures can be collected, these can be used to further the understanding of how improvements have performed.

**Challenge 2: Commitment to SPI**
One often cited problem is that SPI is expected to happen in addition to the regular workload without dedicating special resources [32, 45-48]. SPI initiatives need to obtain management support and commitment, since without dedicated resources the SPI effort will most likely be made in vain.

In addition to management support, the commitment of everyone involved in the SPI initiative must be ensured. It is a good idea to involve the people actively working with the process on a daily basis in the improvement work, as they are the ones who hold the knowledge of what needs to be improved [32, 45-50]. Börjesson and Mathiassen [51] present two driving forces regarding commitment and involvement that influence the success of SPI implementation. Process push depends on the competence, commitment and active participation of process engineers in developing and implementing new technologies, while practice pull depends on the competence, commitment and active participation of software practitioners in developing and adopting new technologies. The degree of process push and practitioner pull dictates the outcome of the SPI implementation effort. If the degree of both is low, the SPI effort will not happen. If only one of the pull or push forces is high the SPI may happen, and if both are high the SPI effort will most likely succeed.

**Challenge 3: Initiation threshold to SPI**
Extensive SPI frameworks, such as CMMI and SPICE, are viewed by many as too large to comprehend and implement [11, 12]. A typical CMM SPI cycle, for example, can take between 18 and 24 months to complete and demand large resources and long-term commitment [52]. This implies a long time to return on investment (ROI) and does not facilitate SPI success, as companies often require SPI efforts to show a
positive ROI within a year [48]. This makes it difficult for organizations, in particular small to medium sized enterprises (SMEs), to initiate and perform assessment and improvement efforts, as cost and time are crucial considerations [11, 12, 28]. This is a major concern for the SPI research field, as a vast majority of the current software industry is comprised of companies with fewer than 50 employees [53]. The problems faced by SMEs also pertain to small organizations within a larger company if it functions as a separate cost center [54]. This is confirmed by research initiatives to develop lightweight SPI frameworks such as IMPACT [55] and ADEPT [56] and to scope CMMI appraisals given organizations’ characteristics [57, 58].

The situation is not alleviated by the fact that many practices advocated by prescriptive frameworks do not comply with the needs of small organizations [54]. As prescriptive frameworks employ a one-size-fits-many policy, they might force practices on practitioners that they do not consider necessary, or miss issues important to the organizations [59, 60]. Inductive frameworks, on the other hand, address this by basing improvements on the organization’s situation. Finding possible improvement issues based on organizational needs, and not by following a prescribed framework, can help ensure support, from practitioners and management alike, for the assessment and subsequent improvements. This is a critical aspect in ensuring the success of process improvement efforts [32, 45-48, 50, 61, 62]. However, basing improvements on quantitative analysis, as proposed by QIP, produces useful results only if the process exhibits significant maturity [63].

**Challenge 4: Visibility of SPI results**

SPI initiatives need to show short-term benefits to keep the effort visible and maintain the commitment of staff and management [32, 48, 64-66]. To effectively provide short-term benefits, the improvements must be implemented in a continuous manner with small increments. Increments enable learning from experience, and thus the opportunity to correct failures and modify processes accordingly during the course of improvement [51]. Improving in increments implies giving priority to what needs to be improved and in which order things should be improved. When using prescriptive frameworks, these priorities are guided by the frameworks. Many organizations find that adopting processes in orders other than the prescribed one is more effective, e.g. early adoption of change control and code review [40]. Even though the continuous representation in CMMI and ISO/IEC 15504 improves visibility, in comparison to the staged representation, as practitioners can focus on one process at a time, the framework still guides priorities concerning the order of adoption of practices in each process area.

In order to adopt and institutionalize improvements on a broader scale, organizations need to ensure that their improvement initiatives provide the expected benefits. This is done through measurement, as it is the only way to demonstrate whether an improvement has actually occurred. However, measuring the success of SPI initiatives is not easy. Traditionally, SPI success has been measured in terms of gains in productivity, reductions of cost, improved ability to meet schedules, and improved quality and customer satisfaction [41, 67, 68]. While all these variables are important, different companies have different business goals and thus give different priority to these variables, or even emphasize others [68, 69]. Measurements should be aligned with business goals and the company’s strategy in order to be viable. Thus, there is a need to tailor measurement programs to align with the strategy and goals of the
company, to be able to evaluate the benefit of SPI initiatives for the company’s competitiveness.

Challenge 5: Tailorability of SPI methods
Enabling alignment of SPI initiatives with a company’s strategies and business goals requires a tailoring of SPI methods. A number of other contextual variables also need consideration, four of these being company size, development size, development mode and development speed [53]. Company size dictates how improvements may be pursued. For SMEs, resources and staff are always a consideration and there is a need for shorter time to return on investment compared to larger companies. Development size decides what practices are likely to be adopted. Software engineering methods are traditionally developed to fit large software projects. These methods are likely to be rejected in small and medium sized projects since they are perceived as ineffective [53, 70, 71]. This is because many methods for large scale software development found in the literature do not scale down well [71]. The process areas in which it is most difficult to achieve improvement by using CMM have been shown to be planning and tracking of software projects [32], which involve the methods that scale down poorly [71]. Development mode comprises the model under which development is performed. Much of the software engineering literature assumes the contract model for software development, where there is an external procurer that orders the software [53]. However, much of the software developed today includes pre-packaged software for mass markets or for internal use where there is no single customer but rather multiple, often contradictory, stakeholder views on what is to be developed. As time to market and coping with changing requirements become more important for companies to be competitive [48], many projects adopt agile methods to increase development speed. Agile methods are difficult to assess with traditional process assessment methods such as ISO/IEC 15504 and CMMI, even though some methods, such as Scrum and XP, map to ISO 9001 [72] and CMMI Levels 2 and 3 [73].

A concern for research is to understand how these contextual variables affect the success of SPI methods, not to embrace a one-size-fits-many approach, and to develop methods that facilitate SPI for a wider audience.

2.2. Requirements engineering
Requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem [74]. Requirements engineering spans all activities related to creating and maintaining a set of requirements for a product. Traditionally, requirements engineering takes part in the beginning of a project and results in a specification that defines the product to be developed [24]. However, in practice, requirements engineering is an integral part of the development life cycle as new and changed requirements need to be managed. Requirements engineering is often divided into the processes of elicitation, analysis and negotiation, specification, validation, and requirements management [74, 75] which are illustrated in Figure 6.
In reality the requirements engineering process is not as straightforward as depicted [76, 77]. It is an iterative and parallel process where requirements are continuously elicited, documented, analyzed and changed. The processes are briefly described below.

**Requirements elicitation**
Requirements elicitation concerns understanding, finding, learning and discovering the requirements for the product. Requirements can originate from several different sources. The usual sources of requirements are the stakeholders of the product including the users of the product, customers, stakeholders internal to the organization etc. Other sources include different types of documents such as laws and regulations, previous or related products, standards etc. Techniques traditionally used for eliciting requirements include requirements reuse, surveys, interviews etc. In addition, several different elicitation techniques have been proposed, including protocol analysis [78], repertory grid [79], creativity workshops [80] etc.

**Requirements analysis and negotiation**
Requirements analysis involves scrutinizing the elicited requirements for potential conflicts, ambiguities, dependences, overlaps, omissions and inconsistencies. In addition, the feasibility and risks of the requirements are assessed to gain an understanding of potential problems. Examples of techniques used for requirements analysis are modeling, interaction matrices, prototyping etc. At this point requirements can also be prioritized for importance. The goal is to reach the best trade-offs of what to include in the product by consolidating different stakeholders’ views. Examples of techniques for prioritizing requirements include the Planning Game, Analytical Hierarchy Process [81] and Cumulative voting [82].

**Requirements specification**
Requirements specification is an ongoing activity and involves documenting the elicited requirements in a specification or repository. Requirements are often specified using natural language, but other techniques such as Use cases [83] and formal specifications [84] are also used in practice. The specified requirements should ideally be correct, unambiguous, complete, consistent, verifiable, modifiable and traceable [85]. However, in practice, achieving these characteristics for large complex systems is impossible [86].
Validation and verification
Requirements verification concerns assessing the requirements to uncover deficiencies in consistency, accuracy and to ensure that they are adequate for development. Validation, on the other hand, involves showing that the requirements actually reflect the needs and wants of the customers. Validation and verification can be performed with different techniques such as reviews, prototypes or simulations.

Requirements management
Requirements management is an ongoing process spanning the entire development life cycle. As new requirements are introduced and old ones change, these changes need to be managed. This includes managing decisions regarding what changes to accept and dismiss, and how to communicate the changes to concerned stakeholders. To alleviate managing changes and keep track of how changes affect the development effort, different types of traceability can be used. In addition, version handling, i.e. documenting and keeping track of how requirements have changed, is part of requirements management.

2.2.1. Selecting requirements engineering practices
It is rather well understood that requirements have a big impact on final product quality, and that inadequate requirements are a major determinant of project failure. Thus, there are good reasons for practitioners to try to adopt successful RE practices. Improving the requirements engineering process often involves spending more time and resources on it. This includes, for instance, more time to elicit and document requirements to achieve good-enough input for development. The idea is that having better input to development will save time and resources, as catching and repairing defects during test or maintenance is costly [6].

Reasons for practitioners to select particular RE practices to use are the following [87]:
- It is the only practice that they know of
- It is the practice of choice for any situation/project
- The practice is prescribed by the used methodology used e.g. RUP [88] or SCRUM [89]
- The practice is understood to be effective in the current situation/project

The first three reasons for selecting practices can in the long run limit the potential for improvement, as organizations might become increasingly good at using outdated or underperforming practices [19]. Several frameworks for requirements engineering practice selection have been proposed in literature, aimed at increasing the likelihood of practitioners using and understanding the fourth reason for practice selection [90-94]. By characterizing and describing RE practices in terms of strength, weaknesses and applicability, practitioner awareness is expected to increase. Tools have also been proposed to aid in the selection of practices guided by the situation in which they are to be applied [91, 92]. However, given that empirical evidence of RE practice performance is very limited (which is investigated in Chapters III and IV) the recommendations embedded in the frameworks are at least open for discussion [95]. Without guidance or evidence on return for investment on RE practices, uptake into industry might be hampered. This poses new challenges for RE researchers, as they need not only to develop new practices but also to provide convincing evidence of
their performance. This includes comparative studies between different RE practices and in different contexts to understand under what conditions certain practices are applicable. These comparisons also need to consider the best alternative investment. Comparisons between different practices that have only been applied in academia offer little advice on what works in practice.

2.3. Knowledge management in software engineering

Knowledge management is a large interdisciplinary research field. There are many definitions of what constitutes knowledge management. One commonly used is Davenport’s which defines knowledge management as “a method that simplifies the process of sharing, distributing, creating, capturing and understanding of a company’s knowledge” [96]. Knowledge management has gained much attention in software engineering, and a systematic review investigating the empirical evidence in the field has been published [97].

There are two main strategies for knowledge management: codification and personalization. Codification entails making the knowledge explicit by documenting it and making it available in the organization. Personalization involves capitalizing on tacit knowledge. Personalization supports the flow of information in the organization by providing an inventory of knowledge sources (people). In addition to these strategies, knowledge management approaches can be divided into different schools [98]. The schools are broadly categorized as “technocratic”, “economic” and “behavioral”. A successful knowledge management approach often integrates elements from several of these schools. The schools are briefly described below.

Technocratic schools

The technocratic schools support and condition employees in their everyday tasks. Three schools are categorized as technocratic: the systems, cartographic and engineering schools. The goal of the systems school is to capture knowledge in knowledge bases that other people can access. This school has been criticized for relying only on codification of knowledge. The risk is that this kind of effort will end up as an information junkyard [99]. However, there are examples of successful implementations of knowledge repositories in software companies [97].

The cartographic school is concerned with mapping organizational knowledge by establishing directories or “yellow pages” of who knows what in the organization. The idea is to make knowledgeable people in the organization accessible for others seeking advice or knowledge exchange. Examples of this school in software engineering are [100, 101].

The engineering school focuses on processes and knowledge flows in the organization. This school is the focus of the majority of empirical research carried out in software engineering on this topic [97].

Economic school

The economic or commercial school focuses on managing intellectual assets such as patents, trademarks, copyrights and intellectual property to increase revenue. This means that it is more concerned with maximizing the benefit of knowledge already owned instead of acquiring new knowledge. Little interest has been devoted to this school in software engineering research [97].
Behavioral school
The behavioral school aims to stimulate management and managers to be proactive in the creation, sharing and use of knowledge. Three schools are categorized as behavioral: the organizational, spatial and strategic schools. The organizational school uses organizational structures or networks to share or pool knowledge. These are often called “knowledge communities”, a group of people with a common problem, interest or experience. The spatial school focuses on how office space can be designed to facilitate knowledge exchange. The strategic school focuses on knowledge management as a part of competitive strategy. This includes, for example, strategies for how to use existing knowledge and how to foster innovation.

The main assets for software organizations are the employees, their expertise and experiences. However, there are arguments pointing out software organizations’ inability to learn, and even that they have learned to fail, which makes knowledge management a promising concept in software engineering. The Experience Factory approach has been central in software engineering for leveraging knowledge internal to software organizations, and for learning from past successes and failures in order to improve development processes. EF is an infrastructure for sharing and reusing experiences gathered in projects carried out in an organization. Section 2.3.1 provides a description of the Experience Factory approach and presents related work. One practical way of gathering experiences from development projects is to use project postmortems. Postmortems are introduced in Section 2.3.2.

2.3.1. Experience factory
In software engineering, to reuse processes, product and life-cycle experience is often referred to as having an experience factory. The experience factory is a separate organization responsible for capturing and reusing knowledge in the organization. The infrastructure of the experience factory is illustrated in Figure 7.

![Figure 7: Overview of the Experience Factory approach.](image-url)
The goal of the development organization is to develop and deliver products. The development organization also provides the experience factory with experiences and products from projects.

The experience factory organization processes the information from the development organization, and provides feedback and support to the development organization in the form of goals and models from previous development projects.

Experiences from development projects are collected, packaged and finally stored in the experience base to facilitate easy reuse. Packaging entails comparing, analyzing and combining experiences from several projects to generalize, tailor and formalize the experiences for reuse. Packaging for reuse is usually based on measurements as these allow for easy project consolidation [106]. However, packaging is still a difficult activity [107, 108]. The improvement methodology used in the experience factory is QIP presented in Section 2.1.1, accompanied by the Goal/Question/Metric paradigm [109] for setting and measuring operational goals.

The most well-known implementation of EF is the Software Engineering Lab at NASA [49]. However, even though the EF at NASA may have had a positive return on investment, few organizations have the resources to implement and run a similar effort. Other implementations can be found in [108, 110-112]. The main difference is in how experiences are captured and used. At Daimler-Benz (DB) [111], experience bases for describing processes and best practice have evolved for a long time. DB established an experience base/electronic process guide (EPG) to make experiences available to users, and used workshops and interviews to elicit experiences. Other efforts have focused on capturing common problems encountered during development and how these have been solved [110]. Tool support enables users to find solutions to common problems encountered during software development [113]. Other efforts have focused on making EF and QIP more concrete to enable easier use by practitioners [114, 115].

One important prerequisite for EF is closed-loop feedback, in the form of objective measurements of the development process [104]. This integrates experience capture into the normal engineering work. Subjective measures or expert judgment can also be used, but need to be managed, as simply relying on people's willingness to share experiences is not enough [116]. Objective measurements have the advantage of not relying on the judgment of an expert. However, objective measurements are costly and might require a strict adherence to measurement procedure [13, 117-119]. Subjective measurement, on the other hand, requires no strict adherence to measurement procedure. This makes postmortems a good candidate for experience capture in an experience factory approach. Postmortems can be integrated into the normal development process, i.e., something you do after a development phase has ended or on project conclusion. Postmortems are described in the next section.

2.3.2. Postmortems

A practical approach to capture experiences and learn from development projects is the project postmortem [105]. A postmortem is a reflective activity carried out either at the end of a development phase or at project conclusion [105]. The goal of a postmortem is to learn from what happened in the project so as to improve future practice. The concrete result from a postmortem is often a postmortem report. Improvement can
happen on several levels: the individual, project and organizational levels [120].

Individuals can reflect on how they themselves have performed in the project and how they can improve until the next project, e.g., improve their skill or competence in relation to tools and techniques used. Project teams can reflect upon and discuss what they have learned during the project, and individuals share their experiences with the rest of the project team. On the organizational level, projects can reflect on organizational policies and processes and provide tacit insight into how they perform. Experiences from projects can also be communicated to other projects, to make them part of the learning experience [23].

Most research on postmortems in software engineering has focused on using the results for project learning. Several different approaches to postmortems for project learning and for capturing improvement potential in projects have been developed and tested in software engineering [121-124]. However, few have focused on approaches for increasing the impact of postmortems on the organizational level. Considering only the project level can raise problems as discussed in Challenge 1 in Section 2.1.3. The organization needs to consider what improvement is most important to pursue first, given the organization’s goals, not only the project’s goals. In addition, Challenge 2 emphasizes the need to gain management commitment to improvement efforts. Even though postmortems have uncovered issues needing improvement, the improvement effort might fail if management support (dedicated resources) is not assured.

Desouza et al. [120] have investigated the learning outcomes from using different ways of documenting postmortem outcomes, reports and stories. Stories are said to make more organizational impact, but the choice of documentation should depend on the project and learning outcome. Postmortem results can also be disseminated throughout the organization through, for example, workshops [23] or by connecting experience to an electronic process guide [125] to enable organizational learning. Analyzing a collection of postmortem reports can also unveil new insights into projects’ performance and provide useful guidance for process improvement initiatives [118, 119, 126]. However, analyzing postmortem reports to enable organizational learning comes at a high cost [126] and more efficient ways of analyzing postmortem data are needed. One potential way of increasing the efficiency of analyzing postmortem results for organizational learning is to discern what information is usable for organizational learning and document this in a way that is easy to analyze. This is the main idea behind PSF that is presented in Chapters V and VI. PSF is an approach for capturing experiences from postmortems and using these for organizational learning and improvement. PSF focuses on evaluating and sharing development practices. This means that the postmortem can be focused on eliciting and documenting information in a way that enables easy consolidation of experiences.

3. Research approach

The work presented in this thesis has evolved by using empirical studies to gather results. This section describes the research approach used in this thesis. First, the research context is presented in Section 3.1. In Section 3.2, the research model used is outlined as well as how it has been followed in this thesis. Then the research questions driving the research are presented in Section 3.3. Section 3.4 presents an overview of the research methodology used to find answers to the research questions.
3.1. Research context
Most of the studies part of the thesis has been performed in the automotive domain. The automotive domain is characterized by hard competition and the importance of software is increasing as a determinant in enabling innovations and cost savings [1, 127]. The amount of software in cars has been growing exponentially over the last 30 years [1]. Traditionally, automotive systems have been highly modular enabling quite independent development and production where individual sub-systems are sourced from suppliers. However, with the innovations enabled by software, the interactions between sub-systems have increased. The increase of software, increased interaction between sub-systems, and importance of non-functional requirements has increased both the product and process complexity. Most studies in this thesis have been performed at Volvo Car Corporation (VCC) and Volvo Technology (VTEC). VCC is a Swedish automotive manufacturer currently owned by Ford Motor Company. The company uses a top-down, prescribed development process with prescribed development practices. The development methodology used for software development is the MESC method [128]. VTEC is an automotive research and development organization in the Volvo Group. VTEC acts as a supplier to other companies, in or outside the Volvo Group. VTEC as an organization is ISO 9001 [129] certified and uses CMMI for process assessment. Projects carried out at the departments studied often involve 10-30 persons over 12-24 months. Projects carried out in the organization are free to choose what development methods, tools and techniques to use in projects in a bottom-up fashion. However, the development process used in projects is decided in a top-down manner.

3.2. Research process
The work presented in this thesis can be seen as part of a research cycle aimed at transferring research results into industrial use, developed by Gorschek et al. [130, 131]. This section outlines the generic research cycle and then provides an overview of how it has been followed in this thesis. The overall research process is illustrated in Figure 8 as seven steps.
Figure 8: Research model (adapted from [131]).

**Step 1** is where the main research direction and research questions are outlined, based on issues and problems identified in industry.

In **Step 2**, current knowledge and state-of-the-art in relation to problems identified in the previous step are investigated. There is no use in inventing new solutions to problems that already have been solved. In the best case, solutions to the identified problems already have solutions that can be adopted to improve industry. In the worst case, no solutions are available but knowledge of state-of-the-art has been established.

If no existing solutions are found in **Step 2**, **Step 3** involves inventing a new solution based on the problems identified and state-of-the-art.

**Steps 4 and 5** involve validating the solution in a limited scale to initially test whether it addresses and solves the intended problems. Validation in academia offers a low-cost and risk alternative to understanding and testing the basic assumptions of the developed solution.

**Step 6** involves using the solution in real projects to validate applicability in real situations.

**Step 7**. When the solution is validated and deemed viable for industry use, it is released and implemented into industry practice.

The motivation behind the research model is to start with a thorough understanding of real problems and to gradually validate and refine the solution to the point where it can be successfully used in industry. The overall goal of this thesis is to find practical means for organizations developing software-intensive products to improve their development processes and practices. Working close to industry and having the goal of
transferring research results into industry use has made the thesis mostly exploratory in nature. The use of the research cycle in this thesis is detailed below.

1. Identify problem

First, an assessment of current problems and challenges at two departments at Volvo Technology was performed to establish a research agenda. A framework for process assessment and improvement planning (iFLAP) is presented in Chapter II along with these two cases of application. Basing the research agenda on real issues is critical for successful research in software engineering [49, 131]. However, all problems found in industry are not suitable to be addressed with a research approach. Often there is a gap between what is used in industry and state-of-the-art. Suitable problems for research to address are ones that generate or confirm knowledge whilst being solved [132]. Other problems can be solved by considering current state-of-the-art or practice.

The improvement packages established during the process assessment and improvement planning are given in Figure 9 with descriptions in Table 1. The improvement issues are given in the rectangles in Figure 9, while the circles above them indicate the priority given to them in the improvement planning. The arrows between improvement issues in the case of Department B indicate the improvement order established. At Department B, the Requirements Engineering Process was seen as needing improvement before the issues of Abstraction and contents of requirements, and Requirements specification could be improved.

![Figure 9: Improvement packages at Departments A and B.](image)

<table>
<thead>
<tr>
<th>Improvement issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction and contents of</td>
<td>The main issue concerning the contents of requirements is the abstraction level at which they are specified. In most of these cases requirements are not detailed enough, giving room for different interpretations and resulting in changes late in the projects.</td>
</tr>
<tr>
<td>requirements</td>
<td></td>
</tr>
<tr>
<td>Requirements Engineering Process</td>
<td>There is a need for a usable requirements engineering process describing all activities, roles and responsibilities.</td>
</tr>
<tr>
<td>Requirements specification</td>
<td>In many projects there is a need to either specify requirements internally or improve the customers’ requirements. This is not recognized by management or process, giving rise to unplanned costs</td>
</tr>
</tbody>
</table>

These improvement issues were used as input to the subsequent steps in the research cycle. Further, even though the improvement packages chosen at the departments are
similar, many priorities of other issues differed. Discussing these differences with representatives from the organization indicated that there was a potential for learning between the departments and between projects. Learning does take place within projects and between projects where experience are exchanged, e.g. between project managers. These projects can be seen as communities of practice [133] where knowledge and expertise are shared. However, the exchange to outside the community can be improved upon, to offer a cost-effective way of alleviating many of the other issues uncovered in the assessment. Thus, evolving learning between projects in the organization and between different organizational units was also seen as a potential research direction.

2. Study state-of-the-art

Having established two different research themes, state-of-the-art was studied. In software engineering, reusing life-cycle experience and processes for software development is often referred to as having an “Experience Factory” [104]. EF is a major undertaking and includes everything from reusing life-cycle experiences, i.e., processes, tools, methods etc., to products such as code, design and requirements. For this thesis, devising a lightweight, practically usable experience factory focusing on reusing experience concerning practices in the organization emerged as a possible solution to the learning issue. Practices are a candidate vessel for transferring knowledge, as it is a usual way of learning. Improvement often entails learning a new tool or method for a task.

Focusing only on exploiting experiences, i.e., learning from internal experience, in pursuing improvement limits the potential for process innovation [19]. To enable experience exploration, a study of state-of-the-art related to the issues uncovered in the process assessment was planned. However, focusing only on the immediate improvement issues offers a band-aid solution for the organization. In order to enable learning from others’ success, the scope of the state-of-the-art study was expanded upon to include not only research related to the issues identified in the previous step. Instead, the study focused on gauging the available evidence in requirements engineering, and the degree to which practitioners looking for methods/techniques/models/tools to adopt can use it as decision support. This was done by considering the industrial relevance of evaluations, i.e., the realism exhibited, and the rigor of the presentation. The state-of-art study of requirements engineering is presented in Chapters III and IV.

3. Candidate solution

The empirical evidence available in requirements engineering research was found to be rather limited. One of the few solutions which had been validated and tried in industry and which relates to the issues uncovered in the process assessment is the Requirements Abstraction Model (RAM) [134, 135]. The decision was to hold a workshop to tailor and try RAM. Testing RAM revealed that it was not the ideal fit for the organization. The organization where the trial was performed usually implements systems according to specification from an original equipment manufacturer (OEM). RAM could in this context be used to analyze and restructure the incoming specification. However, this meant that many of the advantages of using RAM were lost, e.g. requirements triage by comparison to company strategies. It also meant that the added overhead for using RAM in this context, as compared to organizations that
specify requirements themselves, would be higher e.g. in the form of added traceability to the original requirements specification.

Related to the second research direction, Practice Selection Framework (PSF) was developed as a tailoring of an Experience Factory. PSF is presented in Chapters V. PSF is developed to provide decision support for project managers looking for practices to use in projects and processes managers in finding improvement potential in the organization. To be practical to use, PSF needs to relate to state-of-practice. Learning is an important activity for software development organizations, but it is not the core value. This means that introducing PSF must not impose too many new activities and responsibilities, which would raise the initiation threshold for usage. In PSF, experiences from using practices in projects are captured in postmortems and stored in an experience base to enable other projects to take part of the experiences. Project managers from other projects can use the experiences to get an understanding of how practices have performed, and also to gain incentive to try new practices that have worked well in other projects. Process managers use the experiences to find practices that have not worked well, as these constitute an improvement potential. This can motivate more thorough investigation into these issues [119]. An example of how retrospective analysis can be used to further investigate issues regarding practice use is presented in Chapter VII.

Most of the information captured in postmortems is expert opinion on how practices have performed. The reasons for mostly relying on expert judgment in PSF to discern practice performance instead of using objective measurements are:

- It relates to state-of-practice at the studied organizations: i.e., the cost of implementing extensive measurement programs that enable measuring performance of individual RE practices was seen as a step too big for this thesis. In addition, measuring the requirements engineering process in general is difficult [42].
- Collection of less formal metrics requires less effort and can provide insight into a broader range of issues [117, 119]. For example, holding a workshop with test experts in an organization often requires fewer resources than collecting and analyzing bug reports from the same organization [119].

However, if objective measurements of practice performance are available, these should be incorporated in judging practice performance as they limit the subjective influence in the evaluation. Ideally, a mix of objective and subjective measurements should be used to discern practice performance [136]. Objective measurements increase reliability, and expert judgment enables capturing issues that are not easily measured in an objective manner. The subjective measurements gathered can also be used to select what objective measurements are needed [119].

4. Validation in industry and academia

Several validations have been performed as part of this thesis. First, the application of iFLAP in industry is a dynamic validation of lightweight process assessment and improvement planning in industry. The validation shows that iFLAP can be used to uncover improvement issues and plan for how to pursue these.
Second, PSF was developed, refined and evaluated in steps. An initial static evaluation in industry focused on collecting feedback on industry applicability and testing whether PSF could be tailored to fit an organization’s needs. The evaluation is presented in Chapter V. To improve the realism of the evaluation, tool support was developed. Tool support enables potential users to test how PSF can support them in selecting practices and identifying improvement potential. Two evaluations were performed using the tool, one in academia and one in industry. The evaluations using tool support are presented in Chapter VI. The results from the two evaluations in Chapters V and VI differ and more evaluations are needed.

In addition, Chapter VII shows an example of a practice that has been transferred to, and used in industry - Use cases. Table 2 gives an example of an experience captured in the workshop presented in Chapter V from using Use cases. Even though Use cases are estimated to be successful, there are opportunities to improve use of the practice. The study presented in Chapter VII can be used to strengthen the trust in the experiences collected: i.e., experiences collected from development projects (in postmortems) can be used to initiate collection of objective data on issues [119]. In this case, the experiences collected from the workshop are strengthened by the study in Chapter VII, as the need for domain models for establishing a common taxonomy is strengthened. In addition, cross-team reviews can be used to share knowledge on how to write and review Use cases. The experiences from this study can be stored in PSF as an experience to improve the decision support available for Use cases in the organization.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practice</td>
<td>Use cases</td>
</tr>
<tr>
<td>Project</td>
<td>Project Alpha</td>
</tr>
<tr>
<td>Value added</td>
<td>High</td>
</tr>
<tr>
<td>Success</td>
<td>80%</td>
</tr>
<tr>
<td>Fixed costs</td>
<td>0</td>
</tr>
<tr>
<td>Time</td>
<td>~100 hours</td>
</tr>
<tr>
<td>Pros</td>
<td>Elucidate wanted functionality and can thus be used for handshaking between product planning and systems development.</td>
</tr>
<tr>
<td>Cons</td>
<td>Use cases are not as easy to use as is often assumed. In addition, the terminology used in them differs a lot, which makes them harder to understand.</td>
</tr>
</tbody>
</table>

3.3. Research questions

The overall goal of this thesis is to find practical means for organizations developing software-intensive products to improve their development processes and practices. The research questions presented below have emerged along with a growing understanding of the research area and feedback from industry collaboration. The main research questions investigated are:

RQ1. How can process assessment and improvement planning be performed in an organization where time and resources are limited while still considering the organization’s specific needs, experiences and knowledge?
The starting point for any improvement effort is to understand the current situation and needs. To this end a lightweight framework for process assessment and improvement planning was developed. The process assessment considers the organization’s own experiences and priorities in establishing the improvement needs. This is important for establishing what is most important to improve first for the particular organization. Assessment accuracy is improved by considering multiple internal perspectives on the development process, as well as triangulation of several sources (interviews and documents). To address the issue of cost, improvement planning focuses on establishing what is most important to improve first. This focuses the following improvement effort on the most important things and limits the number of improvements at a time.

The two remaining research questions were developed as a response to the issues uncovered as part of answering RQ1. The second research question was devised in order to find suitable practices/techniques/models/tools ready for adoption that addressed the needs identified during the process assessment.

RQ2. What is the level and quality of available evidence in requirements engineering research?

Software engineering is an applied research field. This means that the ultimate goal is the transfer and widespread use of research results in industry [131, 137]. Thus, one major source of new practices/techniques/models/tools for industry to adopt should be those developed in software engineering research. However, the uptake of software engineering research results has been criticized over the years [138, 139]. This research question aims at investigating the evidence of usefulness and usability of requirements engineering practices/techniques/models/tools published in literature. If there are suitable ones ready for industry adoption, there is no point in developing new ones.

The third research question addresses the second research direction that emerged when answering RQ1. Learning from internal development projects constitutes a relatively low-risk alternative for improvement.

RQ3. How can experiences from several development projects be used to improve organizations, i.e., looking beyond project boundaries, without adding a large overhead for collecting and analyzing experiences?

As the amount of evidence available in RE research was found to be very limited, the third research question is focused on capitalizing on evidence and experience internal to the organization. Chapter V presents Practice Selection Framework (PSF), aimed at supporting project managers in selecting practices to use in projects, and process managers in identifying improvement potential in the organization. PSF uses postmortems for experience capture, and consolidates experiences across projects to enable analyzing practice performance over several projects.

### 3.4. Research methodology

In 1994, Glass [138] discussed the “software-research crisis”. He called attention to the issue of researchers in software engineering studying problems not relevant for
industry, and to the fact that research thus had little impact on practice. To improve the situation, research should be based on problems identified in industry, and solutions should be validated in industry. Performing research in close collaboration with industry poses the challenge of being able to make sense of a complex and poorly controlled situation [136]. To tackle the situation, iteratively improved understanding of problems and solutions is often emphasized over revolutionary discoveries [139]. This implies that several different research methods need to be employed to get a understanding of the problem and potential solutions.

Figure 10 shows the research model used in this thesis, with research methods and data collection procedures superimposed.

![Research Model Diagram](diagram.png)

**Figure 10:** Research methods and data collection approaches used for different stages.

The remainder of this section presents each research method and how data were collected.

### 3.4.1. Research methods

A number of different methods can be used to conduct empirical research. The most commonly used in software engineering are case studies, experiments and surveys [140].

**Case study**

A case study is a study of a contemporary phenomenon in its natural context [141, 142]. In a case study, data can be collected with interviews, observations, document analysis etc. [136]. As case studies are conducted in typical situations, the results are often hard to generalize to other environments.

**Experiments**

Experiments are performed to investigate causal relationships between factors by controlling surrounding variables [140]. Usually, subjects are randomly assigned to
groups with different treatments (tools, techniques) and the outcomes for the different groups are compared. The study presented in Chapter VI is a controlled empirical evaluation, but cannot be characterized as a formal experiment as only one treatment and group were used [140].

Survey
Surveys constitute research in the large where several terms and projects are considered at the same time [143]. Surveys are often undertaken with interviews or questionnaires where a large amount of data is collected. This increases the importance of identifying the population, using proper sampling and to phrasing the questions in an unambiguous manner. Chapters III and IV present a variant of a survey, a systematic literature review (SLR) [144]. The major difference between a traditional review and a systematic one is that the latter aims to minimize error and bias to increase the quality of the review. This is done by using explicit and rigorous methods to identify, appraise and synthesize research on particular research questions established prior to the actual review [144].

3.4.2. Data collection approaches
It is common to discuss different data collection procedures as quantitative and qualitative approaches. Qualitative approaches are concerned with collecting words to describe a situation. Usually, qualitative data are collected with interviews or workshops (group interviews). Quantitative approaches are concerned with measuring objective facts. Quantitative data are often collected as process or product measurements, but questionnaires can also be used to measure the opinions of people. Quantitative data are not suitable for finding previously unknown information, but have the advantage of improving generalization. Often, it is beneficial to combine the approaches [136] to facilitate both generalization and compensating for issues not planned for in the research design. Triangulation can be used to combine different data collection methods while improving reliability of results. The study presented in Chapter II uses triangulation of several different sources (both line and project perspective, and interviews and document analysis).

3.4.3. Validity
Empirical research is always affected by threats that affect the validity of the results obtained. As described by Wohlin et al. [140], validity can be discussed in the terms of construct, internal, external, and conclusion validity.

Construct
Construct validity is concerned with the design of the study instrumentation and with its measuring what is intended. This threat can be limited by piloting the instruments used, and by using methods such as GQM [109] to derive questions used in interviews and questionnaires from goals of the study. Similarly, keeping traceability from research questions to actions taken in the study can limit this threat.

External
External validity is concerned with the ability to generalize the results to other environments. When performing studies in industry, the ability to generalize results is often limited because of the specifics of the context studied. However, studies in industry can lead to generalization by recognition [145]. The ability of generalizing the results depend on the observers and how they react to the problems and causal
relationships uncovered in the study. If they recognize the problems and identify with the context, the results might hold for their situation. However, there is a trade-off between studying real issues in industry and providing solutions that are applicable in other contexts. Researchers need to avoid studying too specific problems that will never generalize to other environments.

**Internal**
Internal validity is concerned with the relationship between the treatment and the outcome. In industry studies there is a real threat that the researcher will influence the results with his/her presence. To limit this type of threat, participants in the study - or even the organization as a whole – can be assured anonymity.

**Conclusion**
Conclusion validity is concerned with the statistical relationship between the treatment and the outcome. To limit this threat, methods for increasing the reliability of data gathered can be used, e.g. triangulation. In addition, one should be careful not to use statistical analysis techniques where the assumptions are not met.

### 4. Research results

The main research results are summarized in this section together with directions for future work. Software engineering can be seen as an applied research field. The engineering part implies applicability of research results and that research should influence practice. Achieving applicability speaks to addressing real problems and testing usability and usefulness of solutions. The overall goal of the research presented in this thesis is to aid industry with improvement in the area of requirements engineering. To achieve this goal, research has been driven by real problems identified in industry, and the solutions proposed have been applied and refined in collaboration with industry (albeit in limited scale). This also means that the methods and techniques presented in this thesis are by no means perfect. The aim has been to develop solutions that are good enough to improve industry whilst still being applicable, i.e., considering the issues of cost, time and having an initiation threshold that enables testing during the research project.

Researchers need to consider the context in which they apply the solution and the goals they intend to achieve. Acknowledging this fact, PSF was developed to be tailored to fit an organization to enable more organizations using it. This also means that the methods proposed in this thesis, iFLAP and PSF, can and should be used in combination with other methods and techniques to form an overall knowledge management and improvement approach. The work presented in this thesis is not intended to compete with or replace any existing model or process. The goal is to complement existing research in the area of software process improvement and knowledge management. For example, PSF focuses on expert judgment in valuing how practices have performed in development projects. This can be combined with prescriptive process assessment techniques, e.g., CMMI, to enable capturing issues that are not yet known by the organization and quantitative measurements to increase the reliability of measurements.

The individual contributions of this thesis are given in Section 4.1 and future work is outlined in Section 4.2.
4.1. Contributions
This section presents the main contributions of the thesis, denoted C1-C7. The contributions together with the corresponding research question and chapters in the thesis are given in Table 3. After each contribution, its relation to the challenges identified in Section 2.1.3 is highlighted.

Table 3: Contributions of this thesis.

<table>
<thead>
<tr>
<th>Contribution</th>
<th>Research question</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Further understanding of lightweight inductive process assessment and improvement planning.</td>
<td>RQ1</td>
</tr>
<tr>
<td>C2</td>
<td>Evaluation of the state-of-research in requirements engineering.</td>
<td>RQ2</td>
</tr>
<tr>
<td>C3</td>
<td>Method for evaluating rigor and relevance of software engineering technology validations.</td>
<td>RQ2</td>
</tr>
<tr>
<td>C4</td>
<td>Method for organizational learning and improvement from experiences.</td>
<td>RQ3</td>
</tr>
<tr>
<td>C5</td>
<td>Initial evaluation of the method for learning and improvement from experiences.</td>
<td>RQ3</td>
</tr>
<tr>
<td>C6</td>
<td>Tool support for the learning and improvement method.</td>
<td>RQ3</td>
</tr>
<tr>
<td>C7</td>
<td>An example of how experiences from using practices can be captured and used to identify improvement proposals.</td>
<td>RQ3</td>
</tr>
</tbody>
</table>

C1: Further understanding and refinement of lightweight inductive process assessment and improvement planning
Chapter II reports on two cases of process assessment and improvement planning in industry. The contribution for research and practice is not in the results from the assessment, but rather in the illustration of lightweight process assessment. The framework presented is not just another model but a packaging and refinement of two previously developed methods [146, 147] into the packaged improvement framework, iFLAP. This packaging into one unified framework, together with concrete lessons learned from industry, is expected to lower the threshold for practitioners to understand and use the approach. In addition, the illustration of process assessment and improvement planning will potentially further lower the initiation threshold for using the framework in practice, as practitioners can see what type of results they can expect from using the framework.

Table 4: Contribution 1’s relation to SPI challenges.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description of how the contribution addresses the challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 1: Focus of SPI</td>
<td>iFLAP does not only consider individual projects in assessment and improvement planning. Several projects as well as the line organization are considered in finding improvement potential.</td>
</tr>
<tr>
<td>Challenge 2: Commitment to SPI</td>
<td>iFLAP involves both management and project personnel in process assessment and improvement planning.</td>
</tr>
<tr>
<td>Challenge 3: Initiation threshold to SPI</td>
<td>Improvement planning establishes a improvement path and identifies what is most important to improve first.</td>
</tr>
<tr>
<td>Challenge 4: Visibility of SPI results</td>
<td>Improvement planning enables stepwise improvement, improving the visibility of how improvement performance.</td>
</tr>
</tbody>
</table>

C2: Evaluation of the state-of-research in requirements engineering
Chapters III and IV report on a systematic literature review of requirements engineering evaluations published in software engineering journals. This contribution refers to the second research question, which was meant to gauge the practically usable
evidence available in requirements engineering research. However, from a practical standpoint the contribution is slim, as the state-of-research offers little practical advice. However, the results could also be seen as motivation to demand relevant problems being tackled when sponsoring research funding, as there are exemplary evaluations. From a research perspective, the evaluation should make the need for changing the research approach abundantly clear. A problem for researchers investigating relevant problems seems to be that novelty, not relevance, is prioritized when publishing papers in software engineering. This reward system promotes research that is not relevant for industry. Performing realistic evaluations adds elements to the publishing process in addition to just developing the idea, which in turn means added risk and effort for the researcher. Researchers in software engineering who prioritize novelty of research over relevance will continue to keep research and practice separate.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description of how the contribution addresses the challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 3: Initiation threshold to SPI</td>
<td>Identifying existing experiences in relation to practices/tools/methods/models can limit the effort for practitioners seeking to adopt new ways of working.</td>
</tr>
</tbody>
</table>

C3: Method for evaluating rigor and relevance of software engineering technology validations
The evaluation presented in Chapter IV is performed using a model for evaluating rigor and relevance of software engineering technology evaluations. Rubrics are used for scoring research, which should make the classification understandable by other researchers to use. In addition, visual aids for evaluating research with respect to type and progress are provided. The model has been refined and tested on requirements engineering evaluations and found to relate well to the state-of-research in requirements engineering.

C4: Method for organizational learning and improvement from experiences
Chapters V presents PSF, a tailored Experience Factory approach for lightweight learning and process improvement using postmortems for experience gathering. PSF builds on two successful ideas, EF and postmortems. PSF is a step towards making EF practically useful in organizations where resources are limited, and to increase the organizational impact of postmortems. PSF integrates experience capture into the engineering process by using postmortems, and thereby assures that experiences are captured. The experiences captured in postmortems relate to how practices have performed in development projects. Project managers, when selecting what practices to use in projects, then use these experiences to discern what practices have worked well in the organization. Process managers use the experiences to pinpoint what practices need improvement or replacement. Experiences are connected to specific practices, which makes the effort needed to consolidate and compare practice performance lower than when collecting experiences without any specific focus.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description of how the contribution addresses the challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 1: Focus of SPI</td>
<td>PSF aids in evaluating and identifying problems with practices over several projects.</td>
</tr>
<tr>
<td>Challenge 3: Initiation threshold to SPI</td>
<td>Subjective measurements collected in project postmortems lower the initiation threshold for starting with an experience factory approach as compared to objective measurements. However, if objective measurements are available, these should be</td>
</tr>
</tbody>
</table>
Incorporated.

| Challenge 4: Visibility of SPI results | PSF enables continuous evaluation of practices and improvements used in projects. |

**C5: Initial evaluation of the method for learning and improvement from experiences**

Two evaluations of PSF have been performed. The first one, presented in Chapter V, is a workshop preformed at Volvo Car Corporation, with the aim to collect initial feedback from industry and refine PSF to suit industry application. The evaluation also shows that PSF can be tailored to an organization’s needs. The second evaluation is intended to collect additional input on PSF usage and to improve the understanding of how organizational characteristics influence PSF usage. This is achieved by using a prototype tool in the evaluation to enable users to use PSF for selecting practices to use in projects and for identifying improvement potential. As using the prototype gives the subjects in the evaluation a better understanding of how PSF would work in practice, the estimate should be more reliable than the workshop.

The evaluation is limited in scale and more participants evaluating PSF are needed to ensure validity of the results. However, the results thus far points to that for sharing practices and experiences as in PSF, the organization needs to already have documented practices and these needs to reflect practice to a fairly large degree. What is documented needs to be worth transferring to other projects. If the documentation have little to do with what is actually performed in projects, the value from transferring it diminishes. Organizations where practices are not documented but rather build on experience and knowledge of staff have a high initiation threshold to using PSF.

For organization where different projects use different practices, PSF can be used as decision support when deciding on what practices to use. Practices and experiences from using them in the organization, can be used as input for deciding on what practices to use in projects. For organizations with prescribed practices, experiences collected in postmortems might not influence practices used in the organization.

**Table 7: Contribution 5’s relation to SPI challenges.**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description of how the contribution addresses the challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 5: Tailorability of SPI methods</td>
<td>The initial validation shows that PSF can be tailored to fit an organization’s needs.</td>
</tr>
</tbody>
</table>

**C6: Tool support for the learning and improvement method**

Based on the experiences from the initial validation presented in Chapter V, a prototype tool was developed to make the process of working with PSF more efficient and to enable testing the usage of PSF. The prototype includes functionality for performing all activities in PSF. However, the tool is a prototype and the functionality and usability needs to improve to enable industry applicability.

**Table 8: Contribution 6’s relation to SPI challenges.**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description of how the contribution addresses the challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 3: Initiation threshold to SPI</td>
<td>Providing tool support should limit the threshold for testing and starting to use PSF.</td>
</tr>
</tbody>
</table>

**C7: An example of how experiences from using practices can be captured and employed to identify improvement proposals**
Use cases illustrate an approach that has been adopted in industry for eliciting, specifying and communicating requirements. Chapter VII presents an retrospective analysis of use cases developed in industrial practice and shows how improvement proposals can be captured. The results from the evaluation can be stored in PSF as experiences. Having experiences collected both from postmortems in development projects and from evaluations that are more objective, improves the decision support related to the practice.

4.2. Future work
This section discusses future directions in which the research can be continued. First, the second research question is only partly answered. To consider all empirical evidence in requirements engineering, other sources than just journals need to be included in the review - e.g. conferences, workshops and technical reports. However, the page limitations for workshops and conferences might mean that the reporting on aspects related to rigor will be similar or worse than for journal articles. Concerning PSF, it should be investigated how postmortems can be tailored to effectively elicit the information needed, whilst considering reliability and validity of measurements.. In addition, further validation including dynamic validation is needed. Ideally, this should take place in different types of organizations to understand the context in which PSF is applicable. In addition, focusing only on experiences inherent to the organization might, in the long run, mean becoming increasingly good at using obsolete practices. Thus, external experiences and evidence needs to be integrated into PSF e.g. the results from the systematic review in Chapters III and IV. In addition, decisions for when to go for exploration or exploitation, when selecting improvements needs to be investigated.

5. List of papers
This section provides a detailed description of the papers included in the thesis, by giving information about each paper including the abstract in Section 5.1. A short description of the contribution of the author in each paper is given in Section 5.2. In addition, papers not included in the thesis are referred to and a short description of how these would fit into the thesis is provided in Section 5.3.

5.1. Included papers
Chapter II: A practitioner’s guide to light weight software process assessment and improvement planning
Abstract
Software process improvement (SPI) is challenging, particularly for small and medium sized enterprises. Most existing SPI frameworks are either too expensive to deploy, or do not take an organizations’ specific needs into consideration. There is a need for light weight SPI frameworks that enable practitioners to base improvement efforts on the issues that are the most critical for the specific organization. This paper presents a step-by-step guide to process assessment and improvement planning using improvement framework utilizing light weight assessment and improvement planning (iFLAP), aimed at practitioners undertaking SPI initiatives. In addition to the guide itself the industrial application of iFLAP is shown through two
Introduction

industrial cases. iFLAP is a packaged improvement framework, containing both assessment and improvement planning capabilities, explicitly developed to be light weight in nature. Assessment is performed by eliciting improvements issues based on the organization’s experience and knowledge. The findings are validated through triangulation utilizing multiple data sources. iFLAP actively involves practitioners in prioritizing improvement issues and identifying dependencies between them in order to package improvements, and thus establish a, for the organization, realistic improvement plan. The two cases of iFLAP application in industry are presented together with lessons learned in order to exemplify actual use of the framework as well as challenges encountered.

Chapter III: Technology transfer decision support in requirements engineering research: a systematic review of REj


**Abstract** One of the main goals of an applied research field such as requirements engineering is the transfer of research results to industrial use. To promote industrial adoption of technologies developed in academia, researchers need to provide tangible evidence of the advantages of using them. This can be done through industry validation, enabling researchers to test and validate technologies in a real setting with real users and applications. The evidence obtained, together with detailed information on how the validation was conducted, offers rich decision support material for industrial practitioners seeking to adopt new technologies. This paper presents a comprehensive systematic literature review of all papers published in the Requirements Engineering journal containing any type of technology evaluation. The aim is to gauge the support for technology transfer, i.e., to what degree industrial practitioners can use the reporting of technology evaluations in the journal as decision support for adopting the technologies in industrial practice. Findings show that very few evaluations offer full technology transfer support, i.e., have a realistic scale, application or subjects. The major improvement potential concerning support for technology transfer is found to be the subjects used in the evaluations. Attaining company support, including support for using practitioners as subjects, is vital for technology transfer and for researchers seeking to validate technologies.

Chapter IV: A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations

**Paper information:** Ivarsson, M., Gorschek, T., Accepted with revisions to Empirical Software Engineering.

**Abstract** One of the main goals of an applied research field such as software engineering is the transfer and widespread use of research results in industry. To impact industry, researchers developing technologies in academia need to provide tangible evidence of the advantages of using them. This can be done through step-wise validation, enabling researchers to gradually test and evaluate technologies to finally try them in real settings with real users and applications. The evidence obtained, together with detailed information on how the validation was conducted, offers rich decision support material for industry practitioners seeking to adopt new technologies and researchers looking for an empirical basis on which to build new or refined technologies.
This paper presents a model for evaluating the rigor and industrial relevance of technology evaluations in software engineering. The model is applied and validated in a comprehensive systematic literature review of evaluations of requirements engineering technologies published in software engineering journals. The aim is to show the applicability of the model and to characterize how evaluations are carried out and reported to evaluate the state-of-research. The review shows that the model can be applied to characterize evaluations in requirements engineering. The findings from applying the model also show that the majority of technology evaluations in requirements engineering lack both industrial relevance and rigor. In addition, the research field does not show any improvements in terms of industrial relevance over time.

Chapter V: Practice Selection Framework


Abstract: Knowledge management (KM) in software engineering and software process improvement (SPI) are challenging. Most existing KM and SPI frameworks are too expensive to deploy or do not take an organization’s specific needs or knowledge into consideration. There is thus a need for scalable improvement approaches that leverage knowledge already residing in the organizations. This paper presents the Practice Selection Framework (PSF), an Experience Factory approach, enabling lightweight experience capture and use by utilizing postmortem reviews. Experiences gathered concern performance and applicability of practices used in the organization, gained from concluded projects. Project managers use these as decision support for selecting practices to use in future projects, enabling explicit knowledge transfer across projects and the development organization as a whole. Process managers use the experiences to determine if there is potential for improvement of practices used in the organization. This framework was developed and subsequently validated in industry to get feedback on usability and usefulness from practitioners. The validation consisted of tailoring and testing the framework using real data from the organization and comparing it to current practices used in the organization to ensure that the approach meets industry needs. The results from the validation are encouraging and the participants’ assessment of PSF and particularly the tailoring developed was positive.

Chapter VI: Tool support for disseminating and improving development practices


Abstract: Knowledge management (KM) in software engineering and software process improvement (SPI) activities pose challenges as initiatives are deployed. Most existing approaches are either too expensive to deploy, or do not take an organizations’ specific needs into consideration. There is thus a need for scalable improvement approaches that leverage knowledge already residing in the organizations. This paper presents tool support for an Experience Factory approach for disseminating and improving practices used in an organization. Experiences from using practices in development projects are captured in postmortems and provide iteratively improved decision support for identifying what practices work well as well as improvement needs. An initial evaluation of using the tool for organizational improvement has been
performed utilizing both academia and industry. The results from the evaluation indicate that organizational characteristics influence how practices and experiences can be used. Experiences collected in postmortems are estimated to have little effect on improvements to practices used throughout the organization. However, in organizations where different practices are used in different parts of the organization, making practices available together with experiences from use, as well as having context information, can influence decisions on what practices to use in projects.

**Chapter VII: An Empirical Quality Assessment of Automotive Use cases**


**Abstract:** As functionality in vehicles grows more complex and development becomes distributed over several geographical sites, elicitation and visualization of requirements become more critical. This paper presents a set of evaluation criteria for the quality of use cases. The criteria are applied to use cases that are currently used in industry and developed according to current industrial practice. The paper presents statistics of quality defects that occur in industry and proposes academic solutions that may be applied to solve them. The study is based on 43 use cases from Volvo Car Corporation spanning three different function areas of a vehicle. The most common quality defect classes of the evaluated use cases are missing elements, irrelevant steps, incorrect linguistics and level of detail. Furthermore, it is concluded that a common taxonomy and cross team reviews are needed to further improve the quality and usefulness of use cases.

**5.2. Statement of contribution**

Martin Ivarsson is the main author of all papers in the thesis. For Chapter II, Fredrik Pettersson and Martin Ivarsson share main authorship and performed the design and analysis of the results together. For Chapter VII, Fredrik Törner, Fredrik Pettersson and Martin Ivarsson share main authorship and performed design and analysis of the results together. In addition, Martin Ivarsson is the originator of the ideas for all papers except Chapter II and Chapter VII. Chapter VII was a joint idea between the main authors.

**5.3. Papers not included**


   **Information:** This paper increases the understanding of the identified problems by comparing and contrasting the issues uncovered in Chapter II with a process assessment performed on the OEM side.

**Information:** A licentiate thesis presented in 2007. It includes some of the work presented in this thesis, namely Chapters II and VII. The licentiate thesis focuses on how to improve the requirements engineering process, especially a particular technology, Use cases.


**Information:** This paper presents an evaluation of a particular requirements engineering technology, namely Use cases. The paper is a complement to the work presented in Chapter II and Chapter VII. The paper presents both a method for evaluating Use cases and results from applying it in industry.

4. Ivarsson, M., Pettersson, F., Öhman, P., “Improved control of automotive software suppliers”, *Product focused process improvement (PROFES)*, 2005

**Information:** This paper presents a specific requirements engineering technology aimed at improving the communication between OEM and supplier when working with Use cases. This is a solution innovation and an initial validation, as the technology is illustrated with an industry example.


**Information:** This paper presents an initial investigation into the issues affecting industry-research collaboration when working with the automotive industry.


**Information:** This paper presents a specific improvement proposal for a requirements engineering technology, namely Use cases. The proposal is a template aimed at capturing important aspects when documenting embedded systems with Use cases.

**References**


In the introduction, several key references are cited as follows:


[34] CMMI-PDT, "Appraisal Requirements for CMMISM, Version 1.1 (ARC, V1.1)," 2001, p. 245.


Introduction


A Practitioner’s Guide to Light Weight Software Process Assessment and Improvement Planning

Reprinted from

A Practitioner’s Guide to Light Weight Software Process Assessment and Improvement Planning

F. Pettersson¹, M. Ivarsson¹, T. Gorschek², P. Öhman¹

¹Department of Computer Science and Engineering
Chalmers University of Technology
{fredrik.pettersson,martin.ivarsson,peter.ohman}@chalmers.se

²School of Engineering, Blekinge Institute of Technology
tony.gorschek@bth.se

Abstract
Software process improvement (SPI) is challenging, particularly for small and medium sized enterprises. Most existing SPI frameworks are either too expensive to deploy, or do not take an organization’s specific needs into consideration. There is a need for light weight SPI frameworks that enable practitioners to base improvement efforts on the issues that are the most critical for the specific organization.

This paper presents a step-by-step guide to process assessment and improvement planning using iFLAP (improvement Framework utilizing Light weight Assessment and improvement Planning), aimed at practitioners undertaking SPI initiatives. In addition to the guide itself the industrial application of iFLAP is shown through two industrial cases. iFLAP is a packaged improvement framework, containing both assessment and improvement planning capabilities, explicitly developed to be light weight in nature. Assessment is performed by eliciting improvements issues based on the organization’s experience and knowledge. The findings are validated through triangulation utilizing multiple data sources. iFLAP actively involves practitioners in prioritizing improvement issues and identifying dependencies between them in order to package improvements, and thus establish a, for the organization, realistic improvement plan. The two cases of iFLAP application in industry are presented together with lessons learned in order to exemplify actual use of the framework as well as challenges encountered.

1. Introduction
To maintain and increase competitive advantages, software organizations must continually strive to refine and improve their development practices. Process assessment and improvement is the means by which development organizations assure and improve their processes and tools to support their specific needs.

Several well known software process improvement (SPI) frameworks exist, most of them cyclic with four main steps: an evaluation of the current practices, planning for improvements, implementation of the improvements and an evaluation of the effects of the improvements. These SPI frameworks can be classified into two main categories: inductive and prescriptive [1]. Inductive methods, such as the Quality Improvement
Paradigm (QIP) [2], take their starting point in a thorough understanding of the current situation, basing improvement efforts on the issues most critical to the specific organization. Prescriptive, or model based, frameworks, such as the Capability Maturity Model Integration (CMMI) [3] and ISO/IEC 15504 (SPICE) [4], take an approach based on a set of best practices that has proven successful in other organizations. The improvements that shall be carried out are established by comparing the studied organization against this set of practices, not however taking any consideration to that particular organization’s specific needs.

Extensive SPI frameworks, such as CMMI and ISO/IEC 15504, are by many viewed as too large to comprehend and implement [5, 6]. A typical Capability Maturity Model (CMM) [7] SPI cycle can for example take between 18 and 24 months to complete, demanding large resources and long term commitment [8]. This makes it difficult for organizations, in particular small and medium sized enterprises (SMEs), to initiate and perform assessment and improvement efforts, as cost and time are crucial considerations [5, 6, 9]. As prescriptive frameworks employ a one-size-fits-all policy, they might also force practices on practitioners that they do not consider necessary or miss issues that are important to the organizations [10, 11]. Inductive frameworks on the other hand address this by basing improvements on the organization’s situation. Finding possible improvement issues based on organizational needs, and not by following a prescribed framework, can help assure support from practitioners and management alike for the assessment and subsequent improvements. This is a critical aspect in assuring the success of process improvement efforts [12-19]. However, basing improvements on quantitative analysis, as proposed by QIP, produces useful results only if the process exhibits significant maturity [20]. Thus there is need to further develop inductive methods, applicable even if the organization does not exhibit extensive maturity, in which the stakeholders’ common knowledge of the process is explored.

When improvement issues have been identified it is crucial for the organization to establish an appropriate way to pursue improvements. Prescriptive frameworks stipulate both what to improve and the order in which to implement improvements and may thus prescribe an improvement plan not compatible with the organization’s needs. There is a need to establish a step-wise improvement plan, allowing organizations to focus on the most critical issues first, based on the inherent knowledge shared by practitioners and managers.

This paper provides a practitioner’s guide to process assessment and subsequent improvement planning through the introduction of iFLAP (improvement Framework utilizing Light-weight Assessment and Planning). The constituents of iFLAP have previously been successfully applied in industry [21, 22]. These have been refined and synthesized in the creation of the framework and are described in more detail in this paper to provide a comprehensive guide to process assessment and improvement planning. The main feature of iFLAP is that it takes an inductive approach to SPI, involving practitioners in both identifying improvement issues and establishing how to implement improvements, while being applicable regardless of process maturity. In contrast to most existing SPI frameworks, iFLAP is explicitly designed to be a light weight improvement framework that makes it possible to assess any process area individually. Thus the framework can be tailored in size and coverage depending on organizational needs. The assessment is based on using multiple data sources as input,
and a triangulation of sources is used to confirm issues. Following process assessment, iFLAP supports the involvement of practitioners in prioritizing and mapping dependencies between identified improvement issues to enable the creation of suitable improvement packages that are realistic for implementation. This enables organizations to focus their improvement efforts to address the most critical issues first, benefiting from the inherent knowledge of the practitioners.

The practitioner’s guide presented here provides a step-by-step description of iFLAP enabling SPI practitioners to carry out process assessment and improvement planning. In addition, the paper presents two cases of industrial application of iFLAP. In relation to the cases, lessons learned are specified. From a research perspective the cases can be seen as a validation of the framework, and from a practitioner’s viewpoint they can be seen as examples of actual use of iFLAP in industry.

The paper is structured as follows:
- Section 2 gives an overview of the background of software process assessment and improvement frameworks and discusses their characteristics and potential limitations in the area as a whole.
- Section 3 introduces and gives a step-by-step overview of iFLAP. This section can be seen as a short manual describing the inherent steps and the different choices afforded to practitioners using iFLAP for process assessment and improvement planning.
- Section 4 presents Volvo Technology (VTEC), the company at which the multiple case study presented in Section 5 was carried out.
- Section 5 presents two cases of iFLAP application in industry, targeting requirements engineering (RE) practices at VTEC. These cases illustrate practical application and show results obtained using iFLAP in practice. The two cases go through each step described in Section 0 presenting results and lessons learned in the actual application of iFLAP.
- Section 6 analyzes the threats to validity of the study, Section 7 discusses general observations from the case studies and finally the conclusions are presented in Section 8.

2. Background and Related Work

Several well known and established SPI frameworks used for process assessment and improvement exist. Most of them are cyclic and based on a general principle of four fairly straightforward steps, “evaluation of the current situation”, “plan for improvement”, “implement the improvements” and “evaluate the effect of the improvements” [9]. A classic example of this continuous and in theory never ending cycle of improvement is seen in Shewhart–Deming’s PDCA (Plan-Do-Check-Act) paradigm, which embraced the necessity of cyclic and continuous process improvement as early as 1939 [23].

SPI frameworks can be divided into two main categories, either bottom-up (inductive) or top-down, model based (prescriptive). In the following sections inductive and prescriptive SPI frameworks are characterized through well known examples [1].

2.1. Inductive Frameworks

Inductive SPI methods take a bottom-up approach, basing what is to be performed in terms of improvements on a thorough understanding of the current situation [24]. A
well-known example is Basili’s Quality Improvement Paradigm (QIP) [2], who proposes a tailoring of solutions based on critical issues identified in the project organization. The solutions are subsequently evaluated in pilot projects before an official change is made in the process [25]. The idea is to base improvements on experiences from executing processes in projects, i.e. there is no general initial assessment or comparison with a pre-defined set of practices. Instead quantifiable goals are set and, based on these, improvements are chosen, which can be in the form of e.g. new processes, methods, techniques or tools.

2.2. Prescriptive Frameworks

In contrast to inductive frameworks, prescriptive or model based process improvement is an approach that is based on a collection of best practices describing how e.g. software should be developed. The prescriptive nature of such models lies in the fact that one set of practices is to be adhered to by all organizations. No special consideration is taken to an organization’s situation or needs, other than how the development process (at the organization subject to SPI) compares to the one offered by the framework [8, 24]. A general trait common to most model based frameworks is that assessments are performed as a benchmarking against the set of practices advocated by the model in question. Interviews, questionnaires and so on are used as tools in the assessment when designed towards benchmarking.

There exist several examples of inherently prescriptive frameworks. Below some of the more well known are listed together with a brief description.

**CMMI**: CMMI (Capability Maturity Model Integration) is an integration and evolution of SW-CMM (Capability Maturity Model for Software) V.2.0C [26], IPD-CMM (Integrated Product Development Capability Maturity Model) V0.98A [27] and SECM (Systems Engineering Capability Model) [28] with the aim to eliminate the need to employ multiple models. CMMI comes in two basic versions: *Staged* and *Continuous Representation*. Both are based on the same 22 Key Process Areas (KPAs) (i.e. the same content), but they are represented differently and thus address SPI in different ways. *Staged Representation* is aimed towards assessing and improving overall organizational maturity. Organizations are evaluated against five different maturity levels and practices (KPAs) are implemented to achieve an overall increase in organizational maturity. *Continuous Representation* on the other hand is adapted towards assessing individual process areas, such as requirements engineering, and improving related practices.

However, it should be noted that even if CMMI allows for targeted improvements it still guides priorities, stating what practices should be improved or added and in what order. Hence it is still prescriptive in nature [29].

The appraisal methodology that is a part of CMMI is based on several appraisal requirements called ARC (Appraisal Requirements for CMMI). These requirements are a basis on which appraisals can be developed, and are hence of primary interest when developing new appraisal methods. The official appraisal method for CMMI is called SCAMPI (Standard CMMI Appraisal Method for Process Improvement) [30] and is developed to meet all requirements described in ARC as well as those needed to be compliant with ISO/IEC 15504 [29]. In general CMMI supports three different classes of appraisals [3, 30].Class A appraisals are the most comprehensive, covering
the entire CMMI model and providing a maturity level rating of the organization as a whole. Class B appraisals are less in depth and focuses on specific process areas that are in need of attention, hence not providing an overall maturity rating. A Class C appraisal is even less comprehensive, often described as “a quick look” at specific risk areas. The effort needed to complete a Class A SCAMPI appraisal is considerable, ranging from 800 to 1600 person hours while Class B appraisals take 80 to 640 person hours and Class C appraisals as little as 20 to 60 person hours [8].

It should be noted that these estimations are of the effort that pertains to the assessors and that the effort of other personnel, e.g. the ones being interviewed/filling out questionnaires, is not included. Some members of the assessment team in the case of Classes A and B appraisals must be trained and certified, while Class C appraisals require little training.

**ISO/IEC 15504:** ISO/IEC 15504 or, as it is commonly known, SPICE (Software Process Improvement and Capability Determination), is an international standard for software process assessment influenced by the now retired CMM [26], but in comparison it is closer to CMMI as it only provides continuous representation. ISO/IEC 15504 is not a reference model in itself but rather sets requirements on models to be used in process assessment. Thus ISO/IEC 15504 relies on externally defined reference models such as ISO 12207 [31] and ISO 15288 [32]. This has also led to the development of domain specific assessment frameworks and reference models that conform to the requirements advocated by ISO/IEC 15504, such as Automotive SPICE [33] and SPICE for SPACE [34].

ISO/IEC 15504 assessments are similar to SCAMPI (CMMI Class A) in that both have similar requirements. A fundamental difference is that while CMMI can use both internal (if trained) or external (SEI) assessment group members ISO/IEC 15504 demands that an external assessor heads the assessment [35, 36]. The effort needed to perform ISO/IEC 15504 assessments ranges from 33 to 824 person hours (with a median of 110) [36, 37].

**2.3. SPI Success Factors**

The motivation for carrying out process assessment and improvement activities is to collect information as to what needs to be changed and to establish how to pursue the improvements in order to minimize development cost and maximize the quality of products produced. Looking at industry experience reports [5, 6, 8, 9, 12-14, 16-19, 38-42] of SPI activities, several critical factors can be identified that influence the success of assessment and improvement activities. Of these factors, the ones relevant for this paper are summarized below.

**2.3.1. SPI Initiation Threshold**

The initial critical success factor is of course that an SPI initiative is adopted in the first place. *The threshold for initiating and committing to an SPI effort* is often high because of the resources that have to be committed. An assessment-improvement cycle is often rather expensive and time consuming [38]. A typical SPI cycle using e.g. CMM can take anything from 18 to 24 months to complete and demands a great deal of resources and a long term commitment in order to be successful [8]. El Emam and Briand report that it takes organizations 30 months to move from Level 1 to Level 2.
(median 25 months), and 25 months to move from Level 2 to Level 3 (median also 25 months) [37, 43, 44]. These figures should also be valid for the staged representation of CMMI.

In addition, the threshold is not lowered by the fact that many view extensive SPI frameworks, such as CMMI and ISO/IEC 15504, as too large and bulky to get an overview of and to implement [5, 6, 9]. This is particularly the case for small and medium sized enterprises (SMEs) (e.g. less than 250 employees) [45] where time and resources are always an issue when it comes to process assessment and improvement [5, 6, 9].

The problem of SPI frameworks being too large, costly and running over extended periods of time (long time period until return on investment) is confirmed by some initiatives in research to develop SPI frameworks of a lightweight type. Examples of this can be seen in the IMPACT project [46] where a QIP inspired framework is presented. Adaptations and versions of prescriptive frameworks such as CMM have also been presented, see e.g. Dynamic CMM [47] and Adept [48].

2.3.2. Commitment and Involvement

Assuming that there is a genuine desire and need for SPI in an organization there has to be commitment from management, which is considered one of the most crucial factors for successful SPI. SPI efforts need to be actively supported and management must allow resources to be dedicated to the SPI effort. An example of a reoccurring problem is assuming that SPI work will be accomplished in addition to the organization’s regular work load [12, 16-19, 42] without dedicating further resources. Management commitment is to some extent connected to the cost and resource issues presented above, as management is less likely to commit to an SPI effort if it is very costly and time consuming.

Commitment from management is however not enough to ensure success. There must be commitment and involvement by management, middle management and the staff, e.g. developers. It is a genuinely good idea to let the ones working with the processes every day be actively involved in the improvement work [12-19]. One reason for this is that people that are a part of the organization often have insight into and knowledge about what areas are in need of improvement, and this knowledge often becomes explicit during an assessment activity [49].

The use of inductive SPI frameworks is based on collecting and using experiences as a basis for all SPI work, which speaks to the advantage of e.g. QIP as the work is based on the experience of coworkers. However, as there is no set of best practices (i.e. like in CMMI or ISO/IEC 15504) improvements might be limited in an organization with low maturity (inexperienced) [20]. Prescriptive frameworks could provide structure and a well defined roadmap to the SPI activity. On the other hand, these frameworks might force practices on e.g. developers that they do not consider relevant or necessary, or miss issues that are important to the organization [10].

2.3.3. The guiding principles of iFLAP

The basic constituents used in iFLAP were first presented by Gorschek and Wohlin [21, 22] in two parts. The methodology has been refined and detailed further in the
creation of iFLAP, a packaged improvement framework holding both assessment and planning utilities.

iFLAP is an inductive process assessment framework that uses multiple data sources for triangulation of the results that are obtained. The framework enables the study of multiple projects and the line organization, as well as the study of multiple sources within each of these organizational units. The triangulation improves the reliability of findings and limits the number of issues, as unconfirmed improvement issues can be dismissed early. iFLAP does not assume that one-size-fits-all; rather it is dependent on the knowledge already residing in the organization under evaluation. This addresses the issue of user involvement as multiple roles on several levels of the organization are involved in the assessments, while documentation is used as a secondary data source. This enables practitioners to be actively involved in the assessment activities contributing to the end result. Without practitioner support, any improvement activity is seriously threatened.

Three main aspects of iFLAP are designed to minimize the SPI initiation threshold and obtain management commitment. First, iFLAP is light weight in nature, adapted to suit smaller organizations, unlike more rigorous frameworks. It is also possible to use iFLAP to evaluate single process areas (RE in the case of the example presented in this paper), but it is scalable so that any or all process areas can be assessed.

Second, the assessment itself is cost effective but relatively accurate, utilizing multiple data sources and investigating several projects without expending large resources. As an example the process evaluations of two departments, which are described in this paper (See Section 5), took approximately 280 person hours to complete, including the hours spent by both assessors and staff.

Third, in addition to process assessment, part of iFLAP is devoted to prioritizing and mapping dependencies between the triangulated improvement issues. This gives the organization possibilities to choose between issues identified and put them together into realistic improvement packages that have a suitable (for the organization) time to return on investment, implementation cost and risk.

3. iFLAP – An Overview

This section gives an overview of iFLAP and a detailed step-by-step account of how it can be used to plan, execute and analyze a process assessment and improvement activity. It is assumed that the scope of the SPI effort has already been defined when commencing with the first step.

Figure 1 shows the three main steps. Step 1 – Selection focuses on the selection of projects in an organization and the roles relevant for the assessment. Step 2 – Assessment deals with the actual elicitation of information through interviews and document study, as well as the analysis of the data gathered and the triangulation of results. Step 3 – Improvement Planning involves the prioritization and choice of what to improve first, based on dependencies, needed resources and cost. These steps are described in detail in the following sections.
3.1. Step 1 - Selection

In order for the findings of the assessment and the subsequent packaging of improvement issues to reflect the opinions of the entire staff, as correctly as possible, it is essential to select the right people as participants in the study. As including everybody working for a company is not feasible in most cases, a careful sampling need to be carried out. This is done in three major steps: first choosing projects to study, then selecting roles (both in project and line organizations) and finally appointing actual people that can represent each role.

To be able to perform this selection it is necessary for the assessors, whether they are external or not, to first have a basic understanding of the organization. This includes knowledge about the business model, the domain, the different products produced, the customers, the main activities performed, roles and projects, and the vocabulary used. A further necessity is to have someone from the studied organization, who is familiar with and committed to the process improvement work and the method used, participate in the selection activities.

To facilitate the assessors’ understanding of the company, workshops with representatives from the organization should be held to establish an overview of the process area under study, including activities and stakeholders involved. A company specific dictionary to be used in the interviews, enabling the assessors to use the studied organization’s own notions and designations, can also be established during this activity. Using an overview of the studied process as a basis for discussion may aid in establishing what activities are performed and which roles are affected. An example of such a process overview, used when selecting participants in the assessment of a RE process, is given in Figure 2. The arrow shape holds the activities part of a generic RE process and phases common in a software development project. Potential sources of requirements are listed to the left, participants in the activities at the top and possible users of the resulting requirements at the bottom. In the workshop, the assessors help the representatives of the company to transform the overview to reflect the organization’s current process.
The results of the workshops are used to select projects and roles to include in the assessment. The following sections describe in more detail what to consider when selecting projects and roles to include, and how to assign participants to the studied roles.

3.1.1. Selection of Projects

As the character of projects often varies even in a single company or department the ones selected for inclusion in process assessment should be chosen with care. For the findings of the assessment to reach high validity, the chosen projects should be representative of the entire population of current and, to the largest extent possible, future projects. However, as mentioned earlier, it is often not feasible to include the entire staff and, similarly, to include all available projects in the assessment. This is because people from all identified roles shall be interviewed in each project in order to maintain the representativeness of the selection.

To get an up-to-date view of the state of the practice, it is recommended that projects that have recently been completed or are close to completion are chosen for the study. It is however not recommended to choose ongoing projects that are far from completion as people involved in these do not yet know the final outcome of the projects. It is thus difficult to evaluate the success of current practices.

Ultimately, it is preferable to rely on the expert judgment of one or more representatives of the management organization of the studied company in the selection of projects. The reasons for this are twofold: (1) they have a better understanding of the organization, the projects and the availability of staff and (2) in the end, it is up to management to grant access to practitioners and documentation.
3.1.2. Selection of Roles

Roles should be chosen such that representatives of all roles that are influenced by the process under study are interviewed. However, it is only meaningful to select roles that actually take part in the activities or are affected by the resulting products of the process being assessed. When selecting roles from the line organization, the roles that are influenced by the assessed process may not be as obvious as when selecting project roles. It is however equally crucial to select appropriate line organization roles and, when doing so, to include those governing the studied process.

The selection of roles is guided by the results of the preceding workshop and, similar to when selecting projects, the expert judgment of representatives of the studied organization.

Selecting the Practitioners

The number of subjects that will assume each role is highly dependent on the nature of the assessment. If the projects studied have similar organizations (i.e. if the same roles are present) and are of similar size, it is a good rule of thumb to have the same distribution of participants in all projects. This is because, as mentioned in the previous section, the projects chosen and the people participating should be representative of the entire organization. If there is a prominent imbalance between the projects concerning the number of practitioners in a particular role it may affect the extent to which the results of the assessment are applicable. Furthermore, the number of subjects in each role can either be determined by the relative influence that the assessed activities and resulting products have on that particular role or by using quota sampling to reflect their distribution in the entire population. Appointment of actual personnel for each of the roles is preferably done by expert judgment by representatives of the studied organization who are familiar with and committed to the process improvement work.

3.2. Step 2 - Assessment

Assessment using iFLAP entails eliciting improvement issues from the organization through interviews with practitioners. The improvement issues gathered are triangulated with project and process documentation for confirmation. An assessment consists of two main parts: a project study, scrutinizing one or more projects, and a line study, which examines the relevant parts of the organization that are not part of a particular project. The two studies utilize two data sources each. The project study consists of project interviews and an analysis of project documentation (A and B in Figure 3) while the line study consists of line interviews and an analysis of process documentation (C and D in Figure 3). Triangulation of multiple data sources increases the validity of the findings compared to relying on a single source [50]. In both studies the interviews are the leading data sources, meaning that issues are always identified in interviews and are either supported or contradicted by the documentation. This ensures that the improvement issues identified reflect the views of the organization. Using documentation as a leading data source would require a definition of state-of-the-art practices on which to base the assessment, similar to the prescriptive frameworks discussed in Section 2.2, which is inconsistent with the use of an inductive method such as iFLAP. The following sections describe in more detail the activities needed to perform an assessment.
3.2.1. Interviews

The interviews in this type of process assessment are primarily exploratory in nature. However, to achieve the best results possible it is recommended that the interviews have a certain level of structure in order not to drift away from the relevant subjects, as the questions asked set the context of the assessment. This can for example be achieved by having a certain number of prepared questions that should be covered in each interview but that are not necessarily asked in the order or form written down. Other recommended practices include asking a number of warm-up questions to gather basic information about the person being interviewed, his/her current and former positions at the company, and his/her project. Furthermore, it can be a good idea to wrap up the interviews by asking for the three things the interviewee considers the organization to be the best at and the three things he/she thinks have the greatest improvement potential. This summarizes the interview in a good way and can also help cover aspects that have not come up earlier. Regarding the length of interviews it should be noted that anything shorter than half an hour is unlikely to produce any valuable results while an interview time of over an hour would probably make too great a demand on busy interviewees.

Establishing which subjects to cover in the interviews is an activity that can be of vital importance to the success of the assessment. What to cover is dependent on the targeted process areas and the organization undertaking the SPI effort. Interview content can for example be based on the assessors’ previous experience, around the practices dictated by prescriptive frameworks such as CMMI or ISO/IEC 15504, or on any other practice list such as the SWEBOK guide. Regardless of the source of interview questions it is however important to remember the inductive nature of the assessment. If for example basing the questions on model based frameworks, they should be used to elicit the opinions of the interviewees, not to compare or benchmark the current process against the practices advocated by the model. Consequently, elicited improvement issues are solely based on the opinions of practitioners. These can differ from what is advocated by the sources of the interview questions, contradicting or transcending them.

When it comes to the collection of data from interviews there are essentially two ways to go: taking notes or recording the interviews. If taking notes the presence of at least two assessors is recommended; otherwise, it will be difficult to keep up the flow of the
interview. On the other hand analysis of the resulting notes will probably be more effective than if the answers were on tape. Writing down what is being said also provides a real-time “sanity check” that can help identify aspects that need to be discussed further. Whether to write down everything said or only certain parts is up to the assessor. An alternative is to transcribe only when the interviewee expresses an opinion about the current state of the practice (either positive or negative) or a possible improvement. The main advantage of recording what is said, apart from that the interviews can be carried out by a single assessor, is that the risk of missing something important is minimized.

3.2.2. Interview and Documentation Analysis
The way to analyze the resulting data (i.e. the answers) from the interviews varies somewhat depending on how they were collected and their level of detail. This section describes a recommended approach, similar to that described by Miles and Huberman [53], in as general terms as possible. Thus it should be applicable regardless of collection method and area studied. For further information on how to analyze qualitative data see for example Miles and Huberman [53], and Robson [51].

The initial set of data is first classified as either describing an opinion or not, to single out what are potential sources of improvement issues, while comments and reflections are simultaneously added. Next, the resulting material is gone through to identify an initial set of categories that will later be elaborated into improvement issues. This second step is an iterative activity where the categories are tweaked by dividing and joining them to correspond to the opinions of the interviewees. As a rule of thumb the categories should reflect concepts of improvement instead of problems in order to be applicable in different contexts, such as several projects. Comments describing the problems faced in development can additionally be added to the categories to reach a deeper understanding of each improvement issue. The number of interviewees among the representatives of project and line organizations, respectively, that support each improvement issue should also be counted. While analyzing the interview data it is recommended to try to identify additional project and line documentation that need to be collected.

The gathered project and line documentation is then analyzed to investigate whether or not the improvement issues identified are supported. This analysis should be guided by the actual interview answers and is unavoidably subject to the interpretation of the analyst.

3.2.3. Triangulation of Improvement Issues
In order to increase the validity of assessment results, avoiding the effects of bias, iFLAP includes triangulation of four data sources: project and line interviews, and project and line documentation. The first step of the triangulation is to establish the number of sources that support the claims made by each improvement issue. This is done by compiling the results of the interview and document analysis in a triangulation matrix. Before proceeding, it is a good idea to define a threshold; a cut-off at two or three is recommended, that defines how many data sources are needed to support a certain improvement issue in order for it to be considered confirmed. Next the number of supporting data sources of each improvement issue is compared to the threshold. The issues with support numbers at or above the threshold are considered confirmed and shall be included in the subsequent improvement planning activities. The
unconfirmed improvement issues are sorted out for the time being but should be considered in later iterations of SPI.

3.3. Step 3 - Improvement Planning

After the improvement issues are established it is important that the organization can determine an appropriate way to pursue improvements. In most cases the risks and cost of implementing all improvements at once are too high. Thus smaller improvement packages need to be singled out so that the improvement effort focuses on a limited number of issues, taking small evolutionary steps. Another aspect is the time to return on investment (TTROI). The TTROI is minimized by delimiting the number of selected issues to address at any one time.

A number of factors such as the needs of the organization, practical restrictions and the cost of implementation must be considered when the choice is made of what to include in each improvement effort. Before starting the improvement planning activities it has to be decided which of the company representatives that should be involved. One alternative is to include the same roles and practitioners as in the assessment step. Roles not directly associated with system development may however be removed when identifying dependencies between improvement issues, as practitioners in these roles may lack necessary knowledge about the practical implications of improvement implementation.

Section 3.3.1 describes how to prioritize improvement issues in order for practitioners to establish a collected view of what is most important to improve. How to determine practical restrictions on the order of improvements by identifying dependencies is described in Section 3.3.2. Analyzing the results of the prioritization is covered in Section 3.3.3, while the packaging of improvement issues is described in Section 3.3.4.

3.3.1. Prioritization of Triangulated Improvement Issues

To be able to select and plan improvements in order to decide what to do first, it is necessary to prioritize the issues identified in the assessment step. This activity aims to distinguish a critical few improvement issues from the entire collection that it is crucial to tackle as soon as possible while simultaneously sorting out those that have been misjudged and have been incorrectly included in the list of possible improvements. This activity is highly dependent on the involvement of the personnel involved in earlier activities of the SPI work (i.e. the people interviewed), who constitutes a selection of representatives from the studied company (see Section 3.1) [22]. To achieve this, methods that are normally used to prioritize requirements can be applied to the improvement issues identified as these are essentially requirements on the development process (as opposed to requirements on a software product). A number of such methods exist, of which the Analytical Hierarchy Process (AHP) [55], Cumulative Voting [54], the Top-ten Approach and Ranking are those most commonly used. A summary of these techniques is given in Table 1 using three different properties that distinguish them from each other. In AHP scaled pair-wise comparisons are made between all items that shall be prioritized, hence the participants make a conscious decision regarding each pair. Using AHP also offers the possibility of calculating the consistency ratio, which indicates the amount of contradictory comparisons. In Cumulative Voting, a pre-defined amount of points, often denoted as money, is distributed among the items. Using this method the participant is not forced to make conscious decisions regarding each pair of items; hence the possibility of
checking the consistency is not provided. Both Cumulative Voting and AHP do however produce results on the ratio scale, which gives the difference in priority between items a magnitude. Ranking and the Top-ten Approach do not provide results on the ratio scale, hence giving only a relative order of items.

Table 1: Summary of prioritization techniques (inspired by Berander and Andrews [54]).

<table>
<thead>
<tr>
<th>Technique</th>
<th>Scale</th>
<th>Granularity</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHP</td>
<td>Ratio</td>
<td>Fine</td>
<td>Low</td>
</tr>
<tr>
<td>Cumulative Voting</td>
<td>Ratio</td>
<td>Fine</td>
<td>Medium</td>
</tr>
<tr>
<td>Ranking</td>
<td>Ordinal</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Top-ten</td>
<td>-</td>
<td>Extremely coarse</td>
<td>High</td>
</tr>
</tbody>
</table>

The choice of prioritization technique depends on how many improvement issues are to be prioritized, as the scalability of the methods varies. If many improvement issues are to be prioritized, the more sophisticated methods require extensive efforts. Prioritizing with AHP requires for example \(n(n-1)/2\) pair-wise comparisons, where \(n\) is the number of items. Thus, if 15 improvement issues are to be prioritized, 105 comparisons would be needed.

Before commencing prioritization the aspects considered when establishing what is most important to improve must be agreed upon. Examples of aspects often included are the quality of the produced products, the development cost and the time to market. The criteria to consider in each specific case depend on the business goals of the studied organization and could thus be identified by looking back on the reasons behind the improvement effort. The aspects chosen need to be communicated to the participants in order to reach a common understanding of what to base the prioritization on.

3.3.2. Identification of Dependencies between Improvement Issues

The order established by prioritizing the improvement issues that have been identified is not necessarily the order in which the improvements are best implemented. Practical restrictions may exist, which make implementation in such a way less than ideal. Letting the same participants as were involved in the prioritization identify dependencies between the improvement issues can help to recognize such restrictions and establish a more practically sound way to pursue improvement.

The identification of dependencies can be done in the same workshop as the prioritization or in an additional one. Each participant draws the identified dependencies between improvement issues, giving each relationship a direction and a motivation. The direction is denoted by an arrow pointing from the dependant towards the issue on which it depends. Each arrow should also be given a motivation to enable the assessors to scrutinize dependencies when compiling the results. This is to be able to identify arbitrary and unclear relationships as well as to enable sorting and comparison of dependencies identified by different participants.

After the workshop each of the dependencies identified is analyzed by the SPI team, who remove those that are vague or irrelevant, and the results are compiled in a list of dependencies that includes the relative weight of each relationship. The weight equals the number of participants that has specified the dependency. Next, dependencies with low weights are removed in order to avoid ending up with a great number of weak
dependencies that need to be considered when packaging the improvement issues. Each relationship shall however be scrutinized to ensure that only those that are the result of misunderstandings or discrepancies are removed and that all valid dependencies are kept. What is considered a low weight can be established before hand by defining a threshold below which dependencies are candidates for removal.

3.3.3. Data Analysis

When planning introduction of improvements to solve the identified issues it is important to have a sense of the level of agreement between the different participants in the prioritization. If a strong agreement can be identified the improvement proposals can be packaged on the basis of the results of the prioritization and the identification of dependencies. However, if there is disagreement between participants, additional measures may be needed to assure commitment to the improvement effort.

Depending on the prioritization method used different methods can be used to analyze the results. The applicability of a selection of analysis methods to the results of the prioritization methods presented in section 3.3.1 is given in Table 2.

If prioritization results are on the ratio scale, disagreement charts [56] can be used to visualize the variation in priority between the individual participants. These can aid in evaluating the level of disagreement between participants on individual improvement issues and thus give an indication of the potential commitment that can be obtained for improvement efforts targeted at each issue.

Satisfaction charts [56] illustrates how the priority ranking of each individual participant or role compare to the resulting ranking of improvement issues for the entire group. The Spearman correlation coefficient is used to calculate the level of satisfaction for each participant, thus the method is applicable when prioritization results are at least on the ordinal scale.

If AHP is used for prioritization, the consistency ratio [55], indicating the amount of contradictory comparisons, can be calculated. Calculating the consistency of provides insight into the reliability of the results, enabling the assessors to sort out results from participants that have been inconsistent in their prioritization.

If disagreement is found using disagreement or satisfaction charts, Principal Component Analysis (PCA) [57] can be used to analyze whether intuitively identifiable groups account for the differences in opinion. If the PCA discovers for example that one of the studied projects forms a group that disagrees with the rest of the participants, it may be a good idea to treat that project separately or further investigate the reasons for the difference in priorities.

<table>
<thead>
<tr>
<th>Table 2: Applicable prioritization analysis methods.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>AHP</strong></td>
</tr>
<tr>
<td><strong>Cumulative voting</strong></td>
</tr>
<tr>
<td><strong>Ranking</strong></td>
</tr>
<tr>
<td><strong>Top 10</strong></td>
</tr>
</tbody>
</table>
3.3.4. Packaging

The last step before implementing changes in the process is the packaging of improvement issues in order to guide planning and implementation. The size and content of each such package are determined from the priorities and dependencies between improvement issues as well as the effects of candidate solutions and the cost of implementation. The main concern here is to package the improvement issues so that an appropriate compromise between these factors is reached. The difficulty and means to achieve this are highly dependent on the content of the improvement issues. Diagrams combining priorities and weighted dependencies serve as decision support when creating packages, each suitable for a single SPI cycle, and establishing the order in which to implement them. Candidate solutions to the improvement issues can be established by relating them to current best practices and state-of-the-art methodologies. The time and resources needed to implement the process improvements govern the size of each package and are decisive factors in choosing what to implement first. When resources are being committed to the improvement efforts the indications given by the data analysis must be taken into account. The level of agreement among the practitioners is an indicator, as is the prioritization itself, of the level of commitment that can be obtained and should thus be considered when allocating resources.

4. Research Context

The case studies presented in this paper were carried out at Volvo Technology Corporation (VTEC), an automotive research and development organization in the Volvo Group. Two departments, denoted A and B, were considered relevant for the SPI effort described as they are concerned primarily with software development projects where VTEC acts as a supplier to other companies, in or outside the Volvo Group. VTEC as an organization is ISO 9001 [58] certified and both of the departments studied have previously undergone ISO/IEC 15504 assessments that showed that improvements are needed to reach the level required by their customers. The assessments identified certain key areas with an improvement potential. However, as the results of the ISO/IEC 15504 assessments did not provide a sufficient roadmap for improvement, a need was identified to elicit issues that the development organization faces and establish how to pursue improvements dealing with these issues. One area with an identified improvement potential was RE, and this was thus chosen for further assessment in the study presented in this paper.

Departments A and B have previously been fairly independent from each other and have thus not shared development processes or collaborated in SPI. One indication of this heterogeneity is that the maturity profiles provided by the ISO/IEC 15504 assessments show differences between the two departments. Now, facing a joint need to improve software development practices, a closer collaboration in these fields was deemed favorable so that they could learn from each other and be able to pursue similar improvements in order to limit risks and overhead cost.

5. iFLAP in Practice

To illustrate the usage of iFLAP, this section presents two case studies carried out at VTEC. The case studies were done at two different departments and their aim was to understand and improve requirements engineering practices. The operation of each activity in iFLAP, the results thereof and lessons the learned are discussed in the
following subsections. The disposition of this section corresponds to the disposition of
Section 3 to give easily accessible illustrations of each part of iFLAP used in practice:
Step 1 - Selection (of projects, roles and participants) is covered in Section 5.1, the
Step 2 - Assessment activities are presented in Section 5.2 and Step 3 – Improvement
planning is described in Section 5.3.

5.1. Selection
To accommodate the selection of projects and roles for inclusion in the study,
workshops were held that included selected leading representatives of each
department. These aimed to establish an overview of the requirements engineering
process, including activities and the stakeholders involved. During these workshops
dictionaries to be used in the interviews, capturing each department’s vocabulary, were
also established.

5.1.1. Projects
Representatives from each department selected the studied projects to reflect all
projects at the department. The selection was based on expert judgment under the
limitations that candidate projects should have been completed in the recent past and
that enough personnel should be available for participation in the interviews.

The Selected Projects
All selected projects are bespoke as VTEC is the supplier to a specific customer. Even
though all projects are externally initiated and VTEC does not sell proprietary
products, all projects essentially entail maintaining and evolving existing products. The
projects selected at each department are described in Table 3 below.

Table 3: Selected projects.

<table>
<thead>
<tr>
<th>Department</th>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Alpha</td>
<td>The Alpha project includes refinement of a car cellular system and adapting it to new hardware platforms. Requirements received from the customer are in the form of detailed requirement and interface specifications.</td>
</tr>
<tr>
<td>A</td>
<td>Beta</td>
<td>The Beta project comprises evolution of a telematics system used in trucks. Requirements received from the customer are mostly high level system requirements that are elaborated internally.</td>
</tr>
<tr>
<td>B</td>
<td>Gamma</td>
<td>The Gamma project consists of evolution of an existing windows application used to facilitate adjustment of engine control in road vehicles. New requirements on the system are received from the customer in the form of wish lists with content ranging from high level needs to detailed bug reports.</td>
</tr>
<tr>
<td>B</td>
<td>Delta</td>
<td>The Delta project comprises maintaining and refining an existing automatic climate control system for different types of vehicles and customers. This includes adapting it to different target vehicles and thus different hardware platforms. Requirements on the specific systems are received from the customers in a number of different forms ranging from high level wishes of attributes of the control loop to detailed interface specifications.</td>
</tr>
</tbody>
</table>
5.1.2. Roles
The roles selected to represent each department were chosen on the basis of the results of the preceding workshops. Those positions that were either involved in the requirements engineering activities or directly affected by the results were included in the study, as were those responsible for maintaining the development processes. This includes roles in both the projects and the line organization. There was a slight difference in the roles chosen between the two departments. All of the roles chosen are described in Table 4 below.

Table 4: Selected roles.

<table>
<thead>
<tr>
<th>Department</th>
<th>Organization</th>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Line</td>
<td>Domain Expert</td>
<td>Technical expert in a specific technical area. Domain Experts are one of the sources when eliciting requirements and are often involved in the specification.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Group Manager</td>
<td>Responsible for resources and competence development. Group Managers also have responsibility for the development process and its documentation.</td>
</tr>
<tr>
<td>A</td>
<td>Project</td>
<td>Product Area Manager</td>
<td>Responsible for initiating (selling) and follow-up on projects. This also includes responsibility for the communication with customers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tester</td>
<td>Traditional role responsible for the verification of the system. The testers take part in the verification of the requirements specification and use the requirements specification when developing test cases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developer</td>
<td>The developers use the requirements specifications when designing and implementing the system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project Manager</td>
<td>Responsible for planning, resource allocation, development and follow-up. The project manager is responsible for the requirements specification, including that all requirements engineering activities are performed.</td>
</tr>
<tr>
<td>B</td>
<td>Line</td>
<td>Group Manager</td>
<td>See Department A above.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quality Manager</td>
<td>Responsible for software quality assurance and process improvement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advanced Engineer</td>
<td>Technical expert that often works in research projects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Product Area Manager</td>
<td>See Department A above.</td>
</tr>
<tr>
<td>B</td>
<td>Project</td>
<td>Project Manager</td>
<td>See Department A above.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developer</td>
<td>The developers use the requirements specifications when designing and implementing the system. At Department B there is no specific test role, developers themselves perform all the testing and thus produces test specifications based on the requirements.</td>
</tr>
</tbody>
</table>
Selecting the Practitioners
The number of participants from each of the chosen roles and who were appointed were decided using quota sampling based on the expert judgment of the company representatives. This was restricted by the availability of staff, as many of the people eligible for participation had important roles in other projects at this point or were unavailable for other reasons. The resulting samplings for departments A and B are presented in Figure 4. The light gray boxes represent the two cases, departments A and B, and present the appointment in the line organization roles as well as the selected projects, the dark gray boxes. The appointment of participants in each project role is presented in the dark gray boxes.

Lessons learned
- Using expert judgment for the selection of participants not only enables the inclusion of official roles (as identified in the previous step), but also the inclusion of unofficial ones not explicitly defined, such as experts in different fields.

![Figure 4: Appointment of participants at departments A and B.](image)

5.2. Assessment
This section describes the operation and results of the process assessment step of the process improvement effort at the two departments at VTEC. The focus of the assessment was the requirements engineering process, thus covering the activities considered best practice in that area. This includes elicitation, analysis and negotiation, specification, verification and validation, and management of requirements. The assessment consisted of conducting interviews, analyzing interview data and documentation, and triangulating the results. The results of these activities are presented in the following sections.
5.2.1. Interviews

The assessment incorporated interviews with each of the 27 selected participants (see Section 5.1.2), carried out over a three week period. Each interview was planned in a two hour time slot of which one and a half hour was to be used for the actual interview and the rest as a buffer for additional interview time and recuperation. Further descriptive data on the time spent on interviews is shown in Table 5.

<table>
<thead>
<tr>
<th>Department</th>
<th>Interviews</th>
<th>Mean time</th>
<th>St. dev. For time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11</td>
<td>89 h</td>
<td>12.4 h</td>
<td>979 h</td>
</tr>
<tr>
<td>B</td>
<td>16</td>
<td>86 h</td>
<td>12.5 h</td>
<td>1376 h</td>
</tr>
</tbody>
</table>

Each interview consisted of three stages. A fixed number of warm-up questions were first asked to gather background information on the interviewee and, if applicable, the relevant project. A semi-structured part that covered requirements engineering topics came next. This included descriptions of current practices, elaborating on reasons behind activities being carried out or not carried out and the interviewee’s opinion thereof. In the last stage interviewees were asked to rate the department’s success with respect to a number of attributes such as accuracy of estimations and customer satisfaction. The participants were further asked to list the three things that work best with respect to RE and the three that they felt have the highest improvement potential. Each of the interviews was carried out by two assessors, one conducting the interview and one primarily transcribing. It was chosen to take notes instead of recording the information as it was considered less intrusive given that two interviewers were available. In total, the 27 interviews resulted in 1313 transcribed records.

Lessons learned

- If the choice is made to transcribe the interviews it is highly recommended that two assessors participate.
- In these case studies the prepared interview questions were influenced by CMM [26] and SWEBOK [52], thus covering what is considered requirements engineering best practice. Semi-structured interviews do however facilitate finding improvement issues beyond the prepared material, thus uncovering organizational specific needs.
- Most records cover facts that do not pertain to the goal of the assessment. Thus an experienced assessor would benefit from minimizing the amount of data transcribed by making on-the-fly decisions on what is important.
- The questions asked set the context of the interviews, thus keeping the focus on the areas identified as interesting. If the interviewee is allowed to elaborate freely there is a risk that the discussion focuses only on details of how things are done instead of also covering what is done and why things were done, all of which are the aim of the interview.
- It is positive to have external assessors, as the anonymity of the interviewees can be assured, which enables them to speak freely.

5.2.2. Interview and Documentation Analysis

To be able to extract improvement issues from the interviews the data were coded and analyzed in several steps. A coarse classification of the data was first made, in order to limit the following steps of the analysis to include only interview records that potentially contained an improvement issue. The remaining records were next
scrutinized to establish an initial list of improvement issues. An iterative procedure followed, in which the remaining records were coded according to the list of improvement issues. The improvement issues were continually updated, merging and splitting issues, in order to most accurately reflect the data and the opinions expressed. Further screening was carried out during this step, removing the records that were considered not to describe an improvement issue. The resulting improvement issues are listed in Table 6 and the number of participants supporting each issue at departments A and B is shown in Table 7. It can for example be seen that improvement issue 3 had the highest support number (9) at department A while issue 1 had the highest support number (14) at department B.

Table 6: Identified improvement issues.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abstraction and Contents of requirements</td>
</tr>
<tr>
<td>2</td>
<td>Allocation of requirements</td>
</tr>
<tr>
<td>3</td>
<td>Customer relations</td>
</tr>
<tr>
<td>4</td>
<td>Dependencies between requirements</td>
</tr>
<tr>
<td>5</td>
<td>Estimation</td>
</tr>
<tr>
<td>6</td>
<td>Project vs. Product</td>
</tr>
<tr>
<td>7</td>
<td>Requirements Engineering Process</td>
</tr>
<tr>
<td>8</td>
<td>Requirements Prioritization</td>
</tr>
<tr>
<td>9</td>
<td>Requirements Specification</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Traceability</td>
</tr>
<tr>
<td>11</td>
<td>Requirements Upkeep</td>
</tr>
</tbody>
</table>
It is important to know who is responsible for each activity and for each artifact produced, in order to adhere to deadlines. One important issue is that roles are not explicitly defined, especially when it comes to requirements engineering.

It is an issue that the state of each requirement, throughout the development lifecycle, is not known. This makes it hard to track the overall progress of the project.

When developing test cases based on the implementation, instead of the requirements, the product is not verified against customer expectations. It is therefore an issue that the quality of requirement specifications is not good enough to be able to develop tests from them and that tests instead are developed after implementation.

Elicitation concerns finding out, from the needs of all stakeholders, which the requirements on the system are. It is an issue that there is no established guidance on how to elicit requirements.

Version handling is necessary to be able to identify different versions of the same requirement and thus be able to compare them.

Table 7: Improvement issue support numbers.

<table>
<thead>
<tr>
<th>ID</th>
<th>Issues</th>
<th>Support number at Department A</th>
<th>Support number at Department B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Project</td>
<td>Line</td>
</tr>
<tr>
<td>1</td>
<td>Abstraction and Contents of requirements</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Allocation of requirements</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Customer relations</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Dependencies between requirements</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Estimation</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Project vs. Product</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Requirements Engineering Process</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Requirements Prioritization</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Requirements Specification</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Traceability</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Requirements Upkeep</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Roles and responsibilities in the RE process.</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>State/progress of requirements</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>System tests from requirements</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Requirements Elicitation</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Version handling</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Even though the problems raised vary between departments, projects and individual participants, the corresponding concepts of improvement are similar. Thus it was decided that the same descriptions of improvement issues could be used for both cases. Of the total of 16 improvement issues three were exclusive to department A and one to department B. Of the 15 issues identified at department A, five were identified in the project interviews only, while three were identified only in the line interviews. Similarly, the department B project and line interviews identified one exclusive issue each.
The document analysis was guided by the results of the interviews and covered both line and project documentation. For each of the identified improvement issues the relevant documentation was scrutinized to check whether or not it confirmed the claim. Most of the documentation analysis was straightforward, making it easy to confirm or refute the issues. Generally, if the line documentation did not provide a description of how to address an issue, it was considered supporting it. In contrast, if the line documentation described practices that addressed an improvement issue, it was regarded as not supporting that issue. Similarly, if the project documentation contradicted what was included, it was considered not to support the improvement issue. I.e. if the interviews yielded that requirements not being prioritized was an issue but the project documentation stated that requirements had in fact been prioritized, the project documentation was considered not to support that improvement issue. The results of the document analysis at departments A and B are presented in the documentation columns of Table 8 and Table 9 respectively. It can for example be seen that improvement issue 1 was supported by both line and project documentation at department A, while not being analyzed at department B as it was not identified in the interviews held with representatives of that department. The results of the document analyses show that, at department A, two of the improvement issues identified in the interviews, Requirements Prioritization and Version handling, were not supported by either line or project documentation. At department B, all of the improvement issues were confirmed by either line or project documentation, or both.

Lessons learned
- While the problems faced are different, concepts about improvements dealing with these problems seem to be similar over different projects and departments.
- It cannot be assumed that it is possible for all necessary documentation to be collected by the assessors before or in conjunction with the interviews, as additional documentation and sources are often identified during the interviews and the analysis of the interview data. It is not uncommon that the assessors have to elicit additional material post-interview.

5.2.3. Triangulation of Improvement Issues
At least two sources supporting an improvement issue were chosen by the SPI team as the threshold for considering it to be confirmed. The results of the interview and documentation analyses were compiled in triangulation matrices, one for each department, to investigate which improvement issues could be considered to be confirmed. The triangulation matrices for departments A and B are found in Error! Reference source not found. and Table 9, respectively.
Table 8: Triangulation matrix for department A.

<table>
<thead>
<tr>
<th>ID</th>
<th>Issues</th>
<th>Interviews</th>
<th>Documentation</th>
<th>Supporting sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abstraction and Contents of requirements</td>
<td>X</td>
<td>X</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Allocation of requirements</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Customer relations</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Dependencies between requirements</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Estimation</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Project vs. Product</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Requirements Engineering Process</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Requirements Prioritization</td>
<td>X</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Requirements Specification</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Traceability</td>
<td>X</td>
<td>X</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>Requirements Upkeep</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>Roles and responsibilities in the RE process</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>State/progress of requirements</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>System tests from requirements</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Requirements Elicitation</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>Version handling</td>
<td>X</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

The triangulation for department A identified two improvement issues, Requirements Prioritization and Version Handling, which could not be confirmed as they were supported by only one data source, thus falling below the set threshold. For department B all improvement issues were supported by at least two sources.

Table 9: Triangulation matrix for department B.

<table>
<thead>
<tr>
<th>ID</th>
<th>Issues</th>
<th>Interviews</th>
<th>Documentation</th>
<th>Supporting sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abstraction and Contents of requirements</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Allocation of requirements</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Customer relations</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Dependencies between requirements</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Estimation</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>Project vs. Product</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Requirements Engineering Process</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Requirements Prioritization</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Requirements Specification</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Traceability</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>Requirements Upkeep</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>Roles and responsibilities in the RE process</td>
<td>X</td>
<td>X</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>State/progress of requirements</td>
<td>X</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>System tests from requirements</td>
<td>X</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>Requirements Elicitation</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>Version handling</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
5.3. Improvement Planning

The planning of the improvement effort was based on input from two workshops at each department that were held after the analysis of the interview data and the documentation was complete. The same personnel that had previously been interviewed were invited to participate. A number of people were not able to attend the workshops, resulting in a drop-out of two participants (one PAM and one developer) at department A and three (all developers) at department B. Each workshop was one hour long, totaling two hours at each department. In the first pair of workshops improvement issues identified were presented and discussed among the participants to reach a common understanding of their meaning and mitigate possible misinterpretations. After these workshops the definitions were tweaked to fit the vocabulary of the participants. In the second pair of workshops the participants were asked to prioritize and find dependencies between the improvement issues. The results of these workshops are presented in the following subsections.

5.3.1. Prioritization of Triangulated Improvement Issues

The first assignment given to the participants in the second pair of workshops was to prioritize the identified improvement issues using cumulative voting. The participants were asked to each divide exactly $100 between the improvement issues and each issue had to be given at least one dollar. Thus an issue given five dollars was considered five times more important than one given only one dollar. All improvement issues, independent of which department they were identified at, were included at both departments in order to minimize the risk of overlooking something important. Each priority was normalized by dividing it with the total amount given to each participant to distribute. The normalized average priorities given to each improvement issue are presented in Table 10. At department A, Estimation and Requirements Engineering Process received the highest priorities (both 0.088) among the project representatives, while Allocation of requirements had the highest priority (0.117) among the line representatives. Combining the priorities of project and line representatives, the department’s overall result shows that Abstraction and Contents of requirements has the highest priority (0.089), while Requirements Traceability has the lowest priority (0.038). At department B both project and line representatives gave Requirements Engineering Process the highest priority (0.100 and 0.158 respectively). Thus, in the combined results, Requirements Engineering Process had the highest priority (0.125), while Requirements Prioritization received the lowest priority (0.036). Figure 5 and Figure 6 visualize the prioritization results, presenting the improvement issues in descending order according to priority in each department. The figures illustrate the rank order of the improvement issues, based on the priorities given to them, while highlighting the differences in priority.
**Table 10: Normalized prioritization results.**

<table>
<thead>
<tr>
<th>ID</th>
<th>Issues</th>
<th>Priorities at Department A</th>
<th>Priorities at Department B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Project</td>
<td>Line</td>
</tr>
<tr>
<td>1</td>
<td>Abstraction and Contents of requirements</td>
<td>0.085</td>
<td>0.097</td>
</tr>
<tr>
<td>2</td>
<td>Allocation of requirements</td>
<td>0.048</td>
<td>0.117</td>
</tr>
<tr>
<td>3</td>
<td>Customer relations</td>
<td>0.072</td>
<td>0.097</td>
</tr>
<tr>
<td>4</td>
<td>Dependencies between requirements</td>
<td>0.055</td>
<td>0.047</td>
</tr>
<tr>
<td>5</td>
<td>Estimation</td>
<td>0.088</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>Project vs. Product</td>
<td>0.053</td>
<td>0.067</td>
</tr>
<tr>
<td>7</td>
<td>Requirements Engineering Process</td>
<td>0.088</td>
<td>0.057</td>
</tr>
<tr>
<td>8</td>
<td>Requirements Prioritization</td>
<td>0.072</td>
<td>0.043</td>
</tr>
<tr>
<td>9</td>
<td>Requirements Specification</td>
<td>0.062</td>
<td>0.07</td>
</tr>
<tr>
<td>10</td>
<td>Requirements Traceability</td>
<td>0.042</td>
<td>0.03</td>
</tr>
<tr>
<td>11</td>
<td>Requirements Upkeep</td>
<td>0.082</td>
<td>0.057</td>
</tr>
<tr>
<td>12</td>
<td>Roles and responsibilities in the RE process.</td>
<td>0.062</td>
<td>0.047</td>
</tr>
<tr>
<td>13</td>
<td>State/progress of requirements</td>
<td>0.087</td>
<td>0.063</td>
</tr>
<tr>
<td>14</td>
<td>System tests from requirements</td>
<td>0.062</td>
<td>0.07</td>
</tr>
<tr>
<td>15</td>
<td>Requirements Elicitation</td>
<td>0.043</td>
<td>0.09</td>
</tr>
</tbody>
</table>

**Figure 5: Department A improvement issue prioritization.**
Lessons learned

- It is valuable to hold a workshop at which the improvement issues are discussed and clarified before letting the participants prioritize and identify dependencies. This assures a common understanding of the improvement issues that is hard to achieve through only written text. In addition, discussing improvement issues may also help catch misunderstandings even at this relatively late stage, thus validating the findings.

5.3.2. Identification of Dependencies between Improvement Issues

The second assignment given to the participants in the second pair of workshops was to identify dependencies between the improvement issues. If an improvement issue had the mitigation of another issue as a prerequisite, that dependency was to be motivated. To consider a dependency mapping significant, it was decided that at least 20% of the participants needed to identify it. With nine and 14 remaining participants, respectively, the threshold was thus two people at department A and three people at department B. The dependencies with enough support to rise above the threshold are presented in Table 11. These are also illustrated in Figure 7, where it can be seen for example that, at department A, the improvement of System Tests from Requirements depends on the preceding improvement of Abstraction and Contents of Requirements.
### Table 11: Weight of dependencies above 20% threshold at departments A and B.

<table>
<thead>
<tr>
<th>Department A</th>
<th>Department B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>From</strong></td>
<td><strong>To</strong></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 7: Dependencies between improvement issues.
Lessons learned

- It is important to make sure that all participants have the same perception of what a dependency is and that it is the same as that of the assessors. Participants may also have a difficult time separating the importance of improvement issues from the order in which they should be addressed.

5.3.3. Data Analysis

The level of agreement between participants at each department was analyzed to gain an understanding of how much confidence could be placed on the results of the prioritization. This was done using disagreement charts, satisfaction charts and principal component analysis (PCA). If the level of disagreement is high for a certain improvement issue there is a potential risk that a sufficient commitment cannot be obtained for improvement efforts targeted at that issue.

The disagreement charts, giving the variation coefficient for each improvement issue, are presented for both departments in Figure 8 and Figure 9. The satisfaction charts, showing how well each participant’s priorities correlate with the department’s average, are found in Figure 10 and Figure 11.

![Disagreement chart for department A](image1.png)

**Figure 8: Disagreement chart for department A (variation coefficient = std. var./mean %).**

![Disagreement chart for department B](image2.png)

**Figure 9: Disagreement chart for department B (variation coefficient = std. var./mean %).**
The two disagreement charts show that what is considered important varies between the participants in the prioritization of the improvement issues. However, the variation coefficients for the improvement issues with the highest priorities are not significantly greater than for the issues with lower priority. Thus, the disagreement between the participants should not influence the planning of the initial improvement effort but could have negative effects on the commitment to subsequent efforts.

The satisfaction charts further support the claim that there are differences in the priorities of the participants as the correlation between the individual prioritizations and the overall order for the departments vary.

Furthermore, the principal component analysis showed that no distinct groups that correspond to organizational units or roles in either of the studied departments could be identified. Thus, the differences in priority given by the disagreement and satisfaction charts cannot be explained by underlying groups. Hence, there is no motivation to tailor improvements for each project. Instead a small number of highly prioritized improvement issues should be chosen for initial implementation.
Lessons learned

- Analyzing the participants’ consensus regarding the prioritization is a valuable, if not necessary, decision support when packaging improvement proposals. It is for example advisable to further investigate improvement issues that are given a high overall priority due to a few of the participants, as improvements addressing these issues may not receive necessary commitment from the rest of the staff.

- Assessing the quality of the prioritization, as can be done when calculating a consistency ratio for pair-wise comparisons [59], would be valuable as it could indicate whether the improvement issues were interpreted consistently.

5.3.4. Packaging

The analysis of the prioritization showed a varying degree of consensus on what is most important to improve. Thus, the SPI team chose to plan for only one initial improvement package and for the solution to include only an introduction of one new method or technology. For the initial selection of improvement issues the prioritization was chosen as a leading data source restricted by the identified dependencies.

At department A, Abstraction and contents, Customer relations and Requirements Engineering Process were candidates for inclusion in the improvement package as State/progress of Requirements depended of the improvement of Allocation of Requirements.

Even though Requirements Engineering Process is the improvement issue with the highest priority at department B it is not suitable to base the selection of candidate solutions solely on this issue. This is because the improvement issue is too broad, in essence entailing all other improvement issues. However, the issues depending on the Requirements Engineering Process can be included as candidates, as improving them is part of the improvement of the process. Thus the Requirements Engineering Process improvement issue puts restrictions on the chosen solutions to be documented and useable, but do not prevent improvement issues depending on it from being candidates for inclusion in the initial package. Therefore, the candidates at department B were Requirements Engineering Process, Abstraction and Contents and Requirements Specification as they are the improvement issues with the highest priority.

The introduction of the Requirements Abstraction Model (RAM) [60] was desired by the representatives from VTEC, if this was in agreement with the findings of the assessment. At department B, the candidate improvement issues are all addressed by the introduction of RAM. At department A, Customer Relations is not address explicitly by RAM but both Abstraction and Contents and Requirements Engineering Process are. As Abstraction and Contents is given a higher priority than Customer Relations and including both would result in two solutions being introduced at the same time, it was decided not to include Customer Relations in the initial improvement package at department A. Furthermore, introducing RAM at both departments is consistent with the aim of pursuing joint improvement efforts, as described in Section 4. The resulting packaging is presented in Figure 12.
Lessons learned

- Packaging improvement issues is inherently difficult and finding one candidate solution that corresponds to the most pressing issues at hand may be impossible. Thus the content of the packages may need to be adjusted.
according to the possible solutions if introducing several new methods is to be avoided.

6. Validity Evaluation

This section discusses the threats to the validity of the study described in this paper. As described by Wohlin et al. [61], validity can be discussed in terms of construct, internal, external and conclusion validity. Construct validity concerns the mapping of the real world to the laboratory. Thus it is not relevant to discuss this type of validity threat here as the study described in this paper was carried out in industry.

6.1. Internal Validity

Participants in the assessment and improvement planning not expressing their real opinions, because they feel restricted by the recording of what they say on paper, is a threat to the internal validity. This threat can be limited by participants being guaranteed anonymity by the researchers in interviews, prioritization and dependency mapping. In the study at VTEC, all participants were guaranteed that their answers would be used only by the researchers and that no information that could be used to identify them would be included in reports to the company or in published work.

6.2. External Validity

The results of the assessment and the improvement planning cannot be generalized to other environments. Other organizations may face different needs and have other ways in which to improve. However, this is not a threat to the validity as generalizing the results is not an objective of the study. The generalization that it is important to be able to make here is for the methods used in assessment and planning to be applicable in other environments. The methods have been used successfully in similar form in previous studies [21, 22] and nothing has been tailored specifically for these nor for this study. Hence, nothing in this approach would cause it not to be applicable in other small and medium sized enterprises or, as in this case, enterprises with small and medium sized projects.

6.3. Conclusion Validity

To ensure the reliability of data that are gathered, interviews should be held without interruptions so that the interviewee is not influenced by discussions with other practitioners. Similarly, the prioritization and dependency identification workshops should be held without breaks or other interruptions and the participants must be asked not to exchange opinions during or between the workshops. In the study carried out at VTEC, all interviews and workshops were held without breaks, thus limiting this threat. Due to restrictions in the availability of the staff, however, several workshops were needed at each of the departments. Even though the participants were asked not to discuss the subject between these workshops, it cannot be guaranteed that no opinions were exchanged.

Regarding the reliability of treatment implementation, the different questionnaires used in interviews, in prioritization and in dependency mapping had previously been used in similar studies and are therefore not considered a threat to conclusion validity.

The sampling technique used in selecting the projects, roles and participants to include in the investigation can pose a threat to the conclusions drawn. If the sampling is
flawed, results may not be representative of the studied organization. This type of threat is considered to be under control here as several projects are included and several data sources are triangulated to draw conclusions. In the VTEC study, expert judgment by experienced company representatives was used in the selection, thus further limiting this type of threat.

7. Discussion

7.1. Focus

Most existing process assessment methods, such as CMM and ISO/IEC 15504, do not emphasize organizational issues. Previous research has however pointed out the necessity to include organizational factors in the SPI effort [20], in addition to the technical aspects of software development.

The need to expand the focus of process assessment to cover organizational aspects is supported by the identification of the Project vs. Product improvement issue at one of the departments, which was part of the assessment of requirements engineering practices at VTEC presented in Section 5.

In the assessment of requirements engineering practices at VTEC, a number of improvement issues were identified only by the line organization. At department A, three improvement issues, Requirements Upkeep, Requirements Elicitation, and Allocation of Requirements, were identified in the line study but not in the project study. At department B, the only issue identified solely by the line study was State/progress of Requirements. Even though the issues given the highest priorities were identified in both the project and line studies, important experience would have been missed if no line study had been carried out.

The correlation between line and project prioritizations was analyzed to further investigate the need to include practitioners beyond the project organizations in process assessments. A non-parametric Spearman correlation was calculated to analyze the agreement between the different perspectives at each department. At department A no agreement could be found between the line and project organizations ($r = -0.049$), while agreement was higher at department B ($r = 0.453$). Thus, to assure a correct view of what is most important to improve, both perspectives must be included in both process assessment and improvement planning.

7.2. Evaluation of Improvement Planning

The relationship between frequencies (i.e. the number of participants raising the issue in the interviews) and priorities was analyzed to investigate the need to prioritize improvement issues during planning. Spearman correlation coefficients were calculated for each department to compare the ordering given by the interviews and the prioritizations. The correlation coefficients ($r$) were 0.465 and 0.822 for departments A and B respectively. Thus, even though the order given by the frequencies corresponds quite well to the order established by the priorities at one of the departments, the prioritization seems to provide valuable information for improvement planning. The prioritization strengthens the reliability of the assessment as it could even out potential shortcomings in the interviews affecting the frequencies of the improvement issues.
Having shown that the prioritization adds necessary information for improvement planning, it has to be investigated whether dependencies identified in the workshops also affect the improvement order. The dependencies identified at department A in the VTEC study put additional restrictions on the improvement order established by the prioritization. The effects of dependencies on the packaging of improvement issues can also be seen in the study done at Danaher Särö [22]. Thus, it seems that both prioritization of, and dependency mapping between, improvement issues add valuable information in the planning of the SPI effort.

7.3. Resources

Many small and medium sized enterprises view existing SPI frameworks, e.g. CMMI and ISO/IEC 15504, as too resource intensive to be adopted. iFLAP is light weight in nature and was developed to enable SMEs to execute SPI initiatives without consuming unreasonable resources. To exemplify the resource usage for process assessment and improvement planning using iFLAP the time spent in the requirements engineering process evaluations of two departments at VTEC is presented in Table 12. The resources used, by both company staff and the assessors, in each of the iFLAP steps are shown. The resources used also cover dissemination of knowledge among the participants and establishment of commitment to improvement issues as continuous validation of findings is achieved in the workshops.

<table>
<thead>
<tr>
<th>iFLAP step</th>
<th>Activity</th>
<th>Company</th>
<th>Assessor</th>
<th>Total (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 - Selection</td>
<td>Selection workshop</td>
<td>6</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>Step 2 - Assessment</td>
<td>Interviews</td>
<td>39,25</td>
<td>78,5</td>
<td>117,75</td>
</tr>
<tr>
<td></td>
<td>Interview and document analysis</td>
<td>0</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Step 3 - Improvement planning</td>
<td>Workshop 1</td>
<td>22</td>
<td>4</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Workshop 2</td>
<td>22</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Workshop analysis</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Packaging</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Total (hours)</td>
<td>92,25</td>
<td>184,5</td>
<td>276,75</td>
</tr>
</tbody>
</table>

The figures presented give the combined resource usage for both departments, thus representing two separate instances of in-depth process evaluation. The bulk of resources were spent on interviews and interview analysis. However, as two assessors took part in each of the 27 interviews in these evaluations, there is a possibility to reduce the effort spent by the assessors in this activity. Furthermore, making on the fly decisions as to what to record in the interview transcripts would decrease the amount of recorded data and thus the time needed for interview analysis. Therefore, the effort spent by the assessors in the assessment step should be taken with a pinch of salt, as significant possibilities exist to reduce the time needed for these activities.

7.4. Potential Limitations and Further Work

Even though iFLAP and its constituents has been the subject of several case studies, its applicability to process areas other than RE remains to be evaluated. Similarly, the
The applicability in other organizations and domains should be subjected to further evaluation.

The skill and backgrounds of the assessors may affect what is covered in the interviews as well as how the gathered data is analyzed. Furthermore, the areas covered in assessments with frameworks such as CMMI and ISO/IEC 15504 are predetermined, while assessments using iFLAP is more exploratory in nature. Hence, utilizing iFLAP may put other demands on the skills of the assessors than commonly employed prescriptive frameworks do. The effects of the backgrounds and competence of the assessors, on the outcome of iFLAP assessments, need to be evaluated in future research.

Furthermore, it can be argued that solely relying on improvements based on issues elicited from the organization may prevent the influence of external sources that could lead to significant improvements. This needs to be investigated, and ways to combine iFLAP with adoption of practices on other grounds than in-house experience should be explored.

8. Conclusions

This paper presents a practitioner’s guide to iFLAP, a packaged, light weight improvement framework containing both assessment and planning utilities. iFLAP is inductive in nature, drawing on the knowledge already residing in the organization, utilizing multiple sources to enable elicitation of improvement issues from both projects and the line organization of the studied company. iFLAP also supports choosing between identified improvement issues and packaging these into improvement packages suitable for introduction into the assessed organization.

iFLAP guides SPI practitioners in the improvement effort as it:

I. Facilitates sampling of projects, roles and practitioners enabling
   a. Choosing the size of the assessment.
   b. Capturing multiple viewpoints on what needs to be improved and how to pursue improvements.

II. Describes the basic methods that need to be applied when performing and analyzing process assessment and improvement planning, including
   a. Descriptions of how to perform interviews, interview and documentation analysis and triangulation of data sources.
   b. Guidance on how to choose an appropriate prioritization method.
   c. A method for identifying practical restrictions on the implementation order, through dependency mappings between improvement issues.
   d. Guidance on how to package improvement issues based on prioritization, dependencies and cost of implementation.

III. Assures that the organization’s needs are acknowledged as
   a. Interviews with practitioners are the leading data source in elicitation of improvement issues.
   b. Prioritization and identifications of dependencies between improvement issues are performed by practitioners.

IV. Assures agreement on and commitment to improvement issues by
   a. Providing continuous validation of identified improvement issues.
   b. Being inductive throughout assessment and improvement planning as identification, prioritization and dependency mapping of improvement issues are performed by practitioners.
V. Assures the reliability of findings thorough
   a. Triangulation of multiple data sources.
   b. Continuous validation of improvement issues by practitioners in workshops.

The involvement of practitioners in both finding out what needs to be improved, and how to pursue these improvements, addresses commitment and involvement, which are identified SPI success factors. In addition, iFLAP lowers the SPI initiation threshold by providing a light weight framework with reasonable resource usage, making it applicable to SMEs, while still providing a focused way to pursue improvements tailored to the organization.

To demonstrate the practical application of iFLAP a multiple case study involving two departments at Volvo Technology is presented together with lessons learned. The process evaluations presented focused on requirements engineering practices, a key process area identified to have high improvement potential. The assessments identified 15 improvement issues and subsequent planning resulted in tailored improvement packages for each department.

Acknowledgements
The authors would like to thank everybody involved at Volvo Technology for their contribution and support throughout the study.

This research has been funded in part by the Swedish industry and government joint research program IVSS – Intelligent Vehicle Safety Systems through the CEDES project.

References


A Practitioner’s Guide to Light Weight Software Process Assessment and Improvement Planning


[34] Synspace, "SPIE for SPACE (S4S)," 2005.


Chapter II


Chapter III

Technology transfer decision support in requirements engineering research: a systematic review of REj

Reprinted from

Requirements Engineering, 14(3), pp. 155-175.
Technology Transfer Decision Support in Requirements Engineering Research: A Systematic Review of REj

M. Ivarsson¹, T. Gorschek²

¹Department of Computer Science and Engineering
Chalmers University of Technology
martin.ivarsson@chalmers.se

²School of Engineering, Blekinge Institute of Technology
tony.gorschek@bth.se

Abstract
One of the main goals of an applied research field such as requirements engineering is the transfer of research results to industrial use. To promote industry adoption of technologies developed in academia, researchers need to provide tangible evidence of the advantages of using them. This can be done through industry validation, enabling researchers to test and validate technologies in a real setting with real users and applications. The evidence obtained, together with detailed information on how the validation was conducted, offers rich decision support material for industry practitioners seeking to adopt new technologies.

This paper presents a comprehensive systematic literature review of all papers published in the Requirements Engineering journal containing any type of technology evaluation. The aim is to gauge the support for technology transfer, i.e., to what degree industry practitioners can use the reporting of technology evaluations in the journal as decision support for adopting the technologies in industry practice. Findings show that very few evaluations offer full technology transfer support, i.e., have a realistic scale, application or subjects. The major improvement potential concerning support for technology transfer is found to be the subjects used in the evaluations. Attaining company support, including support for using practitioners as subjects, is vital for technology transfer and for researchers seeking to validate technologies.

1. Introduction
To maintain and increase competitive advantages, software organizations must continually strive to improve the processes and practices they use to produce software. This is especially true for requirements engineering (RE) as good RE practices are a decisive success factor in large-scale software development. Following process assessment, any issues identified and subsequent improvement efforts are centered on selecting and introducing new technologies to counter weaknesses, and in this context technologies can be anything from methods, techniques and procedures to models and tools (or combinations thereof) [1, 2].
When searching for new technologies, practitioners in industry need adequate decision support material on which to base their decision to adopt. In essence, they need to be able to evaluate the evidence offered in relation to the technology to be able to make an informed decision concerning what choice provides the best investment and to minimize risks associated with the introduction of a new technology. Researchers developing new technologies need to provide compelling evidence of advantages over already existing ones, but also base information on the extent to which the technology has been validated (tested).

Ultimately, success in an applied engineering field such as requirements engineering is reflected in the level of adoption and use of its research results [3]. Still, research in requirements engineering is suggested to have little impact on requirements engineering practices [4, 5]. Reasons might be found in the criticism received for not being relevant to practice and not providing technologies usable in real environments [6, 7]. In addition, the research that has been done has been found lacking proper evaluation [8, 9] leading to little tangible evidence that can be used by practitioners as decision support. However, the arguments are not one-sided, as requirements engineering research is also claimed to provide practitioners with useful results [10].

This paper is intended to provide an objective view of what technologies are present in requirements engineering research, more specifically published in the Requirements Engineering journal (REJ), and to what extent papers describing these technologies provide decision support for practitioners seeking to adopt technologies. A systematic literature review of all research papers published in REJ forms the basis for evaluating the evidence presented in a technology transfer perspective. The major differences between a traditional review and a systematic one lies in that systematic reviews aim to minimize error and bias to increase the quality of the review. This is done by using explicit and rigorous methods to identify, appraise and synthesize research on particular research questions established prior to the actual review [11].

For evidence to be convincing to industry, thus facilitating technology transfer, it needs to be realistic. This means that the environment, subjects and scale of the evaluation should be as realistic as possible [3]. Making evaluations in a realistic setting aids technology transfer as practitioners can evaluate the context to see whether the results are transferable to their own environment [12]. It also addresses aspects such as scalability and usefulness, which are often hard to evaluate in a small-scale evaluation [6].

The research method used to produce the evidence also influences the way practitioners perceive it. In this case practitioners value methods that are most relevant to their own environments, such as case studies and lessons learned [13]. In addition to the evidence itself, the manner in which it is presented is also important. Without detailed descriptions of the study’s design, validity and context, an assessment of the evidence produced is difficult. All these aspects are taken into consideration in the systematic literature review presented in this paper.

The paper is structured as follows. Section 2.1 introduces technology transfer in software engineering and outlines a generic technology transfer model and Section 2.2 provides an overview of related work. The design of the study is described in Section 3. The execution of the study is briefly described in Section 4, and a discussion of the
validity of the study is presented in Section 5. The results of the review are presented and discussed in relation to the research questions in Section 6. Conclusions are given in Section 7.

2. Background and related work

This section gives an introduction and background to technology transfer and related work.

2.1. Technology transfer in software engineering

There has been found to be a wide gap between what is used in requirements engineering practice and what is proposed in the research [4, 5]. The technology transfer from academia to industry is at the center of this gap.

Technology transfer is the process of moving new technologies from academia, and a laboratory environment, to industry and an organization, where they are used to perform engineering tasks. There are two main perspectives of technology transfer. In a research (academic) perspective, technology transfer enables researchers to validate (test) technologies in a real setting. This is only accomplished by enabling and attaining industry transfer [14], as usability and usefulness in industry is the ultimate test of a technology. The second perspective has to do with the research impact on industry. RE practice is not likely to improve if no technology is transferred into practice [15].

Little is known about technology transfer in software engineering. Redwine and Riddle [16] investigated the time needed for maturation of technologies developed in the 1960s and 1970s. They found that it takes in the order of 15-20 years to mature a technology to a state where it can be popularized and disseminated to the technical community, which is a long time in a rapidly changing industry [1]. Looking at these numbers as an argument for sustaining technology maturation lead-times has been called a circular argument [17], i.e., there is nothing inherent in software engineering that prohibits shortening the time needed for technology transfer. Zelkowitz [18] studied the infusion processes, i.e., the process a organization undergoes to adopt a new technology. He found that infusion at NASA took two to four years, which included training, and two to four pilot projects to tailor the technology to the given environment. He also noted that infusion of software engineering technologies differs from other technologies in that they are not products and that it is crucial to understand software engineering technologies to support technology transfer.

Even though technology transfer has received relatively little interest in software engineering, other disciplines have well-established frameworks. Rogers [19] suggests five technology attributes relevant to the relative speed at which technologies are adopted:

- **Relative advantage** – the degree to which a new technology is better than one already available.
- **Complexity** – the degree to which the new technology is easy to understand and use.
- **Compatibility** - the degree to which a new technology is consistent with existing values, past experiences and needs of potential adopters.
- **Testability** – the degree to which the new technology can be tried out and tested on a limited basis.
- **Observability** – the degree to which the new technology has visible results for adopters.

Based on Rogers’ framework, recommendations for software engineering research to facilitate technology diffusion have been suggested [1, 2, 17]:

- Software engineering researchers should provide tangible evidence through empirical studies that show the advantages, risks and potential benefits of the new technologies.
- Software engineering researchers need to communicate the context in which and the assumptions under which the evidence presented was observed.
- Software engineering researchers must develop technologies that carry with them a relative advantage of using them.
- Software engineering researchers need to package their technologies with training materials and CASE tool support to enable easier adoption by practitioners.

Several technology transfer models in software engineering embody the need for maturing technologies through different kinds of studies that provide different types of evidence [1, 6, 14, 20, 21]. A general model for technology transfer developed for software engineering is shown in Figure 1 (see Gorschek et al. [14]). The model consists of innovation, static validation, dynamic validation and, finally, the release of the technology for wider use.

![Figure 1: Technology transfer process (adapted from [14])](image)

The innovation step is where the idea is born and developed into a technology. The technology is then tested and evaluated in different settings, from laboratory to industry, utilizing different types of research methods as needed. Static validation of technologies often involves experimentation to investigate the basic concepts of the technology in order to sort out teething problems before the technology is tested in production projects. Given the results of the static validation, one can either move back to the innovation step and refine the idea or move on to dynamic validation to test the idea in a real life setting. Dynamic validation is carried out in case studies in either
“real” software projects or a smaller pilot project aimed at evaluating the technology in question. The last step in the technology transfer process is to release the technology for wider use when it has been shown to be useful and usable.

From an academic research perspective, each step of validation provides different kinds of evidence in relation to the technology being tested. In static validation, variables can be controlled and hypotheses on effectiveness, efficiency and scalability can be tested initially. Dynamic validation does not offer the same level of control; instead the results of case studies provide answers having to do with contextual factors and how the technology scales to real life industry use.

From an industry perspective, the staged technology transfer model described here is an example of how evidence can be obtained through a gradual validation (testing) of a technology in several steps. This evidence, together with detailed information about how the validation rounds were performed (e.g. contextual data), can offer rich decision support for technology transfer.

2.2. Related work

A comprehensive, systematic review was made of all papers published in RE journal to evaluate the support offered to technology transfer in current RE research. Systematic reviews have been used in other fields such as medical research and have received increasing attention in software engineering since they were introduced as a means to evaluate and interpret a field’s collective evidence related to a particular research question in evidence-based software engineering [21][22]. To be able to answer the research questions posed in this systematic review, a data extraction form aimed at collecting information is established. Several different data extraction forms have been used in systematic reviews of software engineering research that range from properties for characterizing the research investigated, e.g. the topic investigated, the research method used [23], to properties aimed at synthesizing several studies carried out in a field, e.g. accuracy and variance [24].

Several reviews, both systemic and traditional, have been published that relate to requirements engineering, some of which appear in Table 1. The studies’ purpose, scope, literature delimitation, data extraction properties, and major results are summarized. It is also noted whether the review is systematic. It can be seen that the studies utilize a diverse set of data extraction properties (for additional ones see [25]). The main idea of constructing the data extraction form is that the properties chosen address the research questions. Thus, as this paper presents a characterization of RE research and identifies evidence from the perspective of technology transfer, a data extraction form aimed at capturing properties relevant for technology transfer is applied (see Section 3.1.3).
In a systematic review, a further factor is the delimitation of the literature that should be included in the review, depending on the purpose of the review. When a particular phenomenon is investigated, all relevant papers should be identified and reviewed, regardless of the kind of source. An example of this type of delimitation is seen in the review by Davis et al. [27] (summarized in Table 1) that investigates empirical studies concerning the effectiveness of elicitation techniques. However, when the purpose is to characterize a field, delimitation with respect to time and source might be appropriate, which is done to limit the effort needed to do the review while still providing usable results. An example of a review characterizing research in software engineering is found in [23] (summarized in Table 1), where papers from six leading journals are

<table>
<thead>
<tr>
<th>Table 1: Reviews related to requirements engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><strong>Systematic review</strong></td>
</tr>
<tr>
<td><strong>Scope</strong></td>
</tr>
<tr>
<td><strong>Literature delimitation</strong></td>
</tr>
<tr>
<td><strong>Properties used for data extraction</strong></td>
</tr>
<tr>
<td><strong>Major results</strong></td>
</tr>
</tbody>
</table>
reviewed. This is the type of delimitation used in this literature review, as the purpose is to investigate the requirements engineering field from a technology transfer perspective.

3. Study design overview

This section gives a detailed account of the design of the systematic literature review.

3.1. Systematic review design

The process used to conduct the review is shown in Figure 2. It consists first of identifying papers that should be included in the study based on a database search. The second step is to include papers on the basis of the title and abstract of the papers that were identified in the search. The papers included are then classified according to the data extraction form presented in Section 3.1.3 to enable analyzing the research identified. The review process is based on guidelines provided by Kitchenham and Charters [11] with the difference that the study quality assessment is included in the inclusion criteria and scoping, i.e., only papers that present any type of evidence or evaluation related to RE technologies are included in the study.

The following sections describe the details of each step in the review process.

3.1.1. Step 1: Identification of papers

A comprehensive and unbiased identification of research is one of the factors that differentiate a systematic review from a traditional one. The main criterion used in establishing the key words used is that they identify requirements engineering technologies that have been evaluated. The search term was elaborated over several test searches and trial reviews, which aimed to establish synonyms of terms relevant to the review. The population is requirements technologies and the intervention is that the technology has been evaluated in some form. The search terms used for population and intervention are presented in Table 2 and the resulting search term used is then population AND intervention. No search terms were used to discern whether the paper presented a technology; this was deemed impossible as the terminology differed too much between the papers.

<table>
<thead>
<tr>
<th>Table 2: Search terms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Population:</strong> Requirement*, specification</td>
</tr>
<tr>
<td><strong>Intervention:</strong> empiric* OR experience* OR &quot;lesson learned&quot; OR &quot;lesson learnt&quot; OR &quot;lessons learned&quot; OR &quot;lessons learnt&quot; OR evaluat* OR validation* OR experiment* OR stud* OR case* OR example* OR survey OR analys* OR investig* OR demonstrate*</td>
</tr>
</tbody>
</table>

The database search is done using Inspec and is only applied to the title and abstract as full text searches would yield too many irrelevant results [29].

The next section describes the criteria used to determine whether papers identified in the search should be included in the review, i.e., remain for data extraction.
3.1.2. Step 2: Paper selection

The papers identified in the search were examined so that only the ones relevant for answering the research questions would be included (see Section 3.1.3). The main criterion for including papers is that they present an evaluation (of any sort) of a technology that has bearing on requirements engineering. Evaluation is defined to cover a wide range of activities, making the selection “include heavy”, i.e., avoiding dismissing papers that have some sort of evaluation. The term evaluation thus covers the range from application (test/illustration) of a technology on a toy example invented by the researchers themselves, to experiments and any sort of empirical evaluation. In addition, only research papers were considered for inclusion, not editorials, news, correspondence, comments and so on.

The inclusion procedure was applied to the title and abstract, that is no papers were excluded based solely on the title. This is an inclusion heavy selection, i.e., the selection of papers is inclined towards inclusion rather than exclusion. The motivation for including papers based on reading abstracts is that, if the evaluation presented in the paper is a major point of the paper, it is likely to be mentioned in the abstract. Missing including papers that present some form of evaluation on the grounds of only reading the abstract is not considered a threat to the validity of the study as the evaluation is not likely to be a major point of the paper in these cases. In addition, having an inclusion heavy selection enables more papers to pass on to a more elaborate scrutiny and limits the risk of missing papers that are relevant. All papers included were entered into a database for further analysis, which is described in the next section.

3.1.3. Step 3: Research questions and data extraction

Before elaborating on data extraction and analysis, it is important to have a clear idea of the research questions posed for the evaluation. Each research question is listed in Table 3, with a description/motivation. Each of these research questions can then be mapped to the data extraction form shown in Table 4.
Table 3: Research questions

<table>
<thead>
<tr>
<th>Research question</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1: What RE technologies exist?</td>
<td>To provide an inventory of all RE technologies that have been subjected to some sort of evaluation. In this study, the minimum level of evaluation is that the technology has been applied to an example. The sub research questions further detail the characteristics of the technologies identified.</td>
</tr>
<tr>
<td>RQ1.1: What RE sub-process areas do the technologies target?</td>
<td>The requirements engineering sub-process areas targeted by the technologies.</td>
</tr>
<tr>
<td>RQ1.2: What is the timeline focus of the technologies?</td>
<td>In what timeline perspective the technologies are used, i.e., whether the technologies are used before the project has started, in the project or after it has finished.</td>
</tr>
<tr>
<td>RQ2: What is the state of technology evaluation?</td>
<td>The state of evaluation is in this study evaluated by combining the aspects considered in the sub research questions. Each aspect is further detailed in the sub research questions.</td>
</tr>
<tr>
<td>RQ2.1: What research methods are most common in the evaluation of technologies?</td>
<td>Different research methods offer different kinds of evidence. The evidence produced varies in terms of level of control and realism. Experiments usually offer higher levels of control while case studies offer more realism.</td>
</tr>
<tr>
<td>RQ2.2: In what context are the technologies evaluated?</td>
<td>For evaluations to be realistic, they need to be performed in a real setting. This research question investigates the context in which the evaluations are performed. As the purpose is to investigate the amount of realistic evaluation, the only distinction made is between evaluations in real industrial settings and academic settings.</td>
</tr>
<tr>
<td>RQ2.3: What subjects are used in the evaluation?</td>
<td>The subjects used in the evaluation will influence the results. A distinction is made between industrial practitioners, researchers and students.</td>
</tr>
<tr>
<td>RQ2.4: What scale do evaluations have?</td>
<td>The scale of evaluations is estimated by looking at the applications used. The scale can thus range between toy examples to real industrial applications.</td>
</tr>
<tr>
<td>RQ2.5: What degree of realism do the evaluations have?</td>
<td>Combining research method, context, subjects and scale gives the realism of the evaluation.</td>
</tr>
<tr>
<td>RQ3: To what extent does the state of research support the actual adoption of technologies?</td>
<td>To support adoption of technologies, evaluations need to present plausible evidence of the benefit of using technologies. The valuation of evidence is based on the state of technology evaluation treated by RQ2. In addition, the presentation of the evaluation needs to provide a basis for appraising it. This is evaluated by looking at the extent to which the study design, context of evaluation and the validity of the study are presented. These three aspects are further examined in the following sub research question.</td>
</tr>
<tr>
<td>RQ3.1: To what degree are the evaluations described?</td>
<td>To understand whether the results presented can be transferred to another context, the presentation of the evaluation needs to be understandable. The context of the study needs to be described to be able to understand whether the results presented can be transferred to another environment. Technologies evaluated in a small project might not perform the same in a large project. Description of how a study is set up and executed increases the understanding of the results presented. Finally, the validity of a study concerns how valid the results are for the population of interest. To comprehend the results of a study the validity must be described.</td>
</tr>
</tbody>
</table>

The papers included in the study are reviewed and data are extracted according to the data extraction form presented in Table 4. The data extraction form is derived from the research questions, and the mapping to these can be seen in the table. For example,
RQ1 concerns the technologies presented and is characterized by three different properties (see properties number 1-3 in Table 4). All properties are noted in accordance with what is reported in the papers reviewed (where possible). If the property is not mentioned, it is marked in accordance to the reviewer’s understanding; as the aim is to investigate the support for technology transfer, all evidence presented in the papers is considered relevant.

The first three properties (1-3) are marked once for each paper and the rest (4-10) are marked once for each study appearing in the paper. This enables capturing cases where several evaluations of a particular technology are presented in one paper. For the first three properties (1-3), one paper can have several different values, e.g. a paper can present several different technologies. In these cases, the property is marked once for each suitable value.

The remaining seven properties (4-10) can be mapped to the research questions treating state of technology evaluation (RQ2) and support for technology transfer (RQ3). The properties all concern the credibility of the evidence, i.e., how the evidence is produced and presented. This is derived from technology transfer being able to be supported by providing trustworthy, realistic evidence [3] and descriptions of how and where, i.e., in what context the evidence has been produced. Each of these properties is marked only once per study. For example, one study utilizing different research methods will only be scored as using one. In these cases papers are scored in a favorable way, and the research method providing the strongest evidence is used.

The Technology (1) property is scored according to what is presented in the papers. It is also noted whether the technology is a refinement or further development of another technology.

Project focus (2) deals with the perspective of timeline of the technologies in relation to projects, i.e., the technology’s main use is either before, during or after the project. Sub-process area (3) categorizes technologies with respect to what sub-process area of requirements engineering they are used in. The values of this property are adapted from [30].

Research method (4), adapted from [23], captures the research method used to produce the results.

Context described (5), Study design described (6) and Validity discussed (7) all relate to the paper conveying an understanding of context and the assumptions under which the evidence for a particular technology have been derived. This is one of the aspects that has been identified to support technology transfer, as practitioners can evaluate the similarity to their own environment and thus assess whether the evidence is valid for them [2, 12, 13, 19].
Table 4: Data extraction form

<table>
<thead>
<tr>
<th>Nr</th>
<th>Property</th>
<th>Values</th>
<th>Description</th>
<th>Mapping to RQs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Technology</td>
<td>From the reviewed literature.</td>
<td>The name of the technology under evaluation. These values are compiled from the reviewed papers.</td>
<td>RQ1</td>
</tr>
<tr>
<td>2</td>
<td>Project focus</td>
<td>Pre-project, In-project, Post-project, N/A</td>
<td>Project focus specifies whether the technology is meant to be used before starting the project, e.g. market driven, in the project or after the project is concluded.</td>
<td>RQ1</td>
</tr>
<tr>
<td>3</td>
<td>Sub-process area</td>
<td>Elicitation, Analysis &amp; negotiation, Management, Validation, Specification, N/A</td>
<td>Process area comprises the requirements process areas to which the technology evaluated belongs.</td>
<td>RQ1</td>
</tr>
<tr>
<td>4</td>
<td>Research method</td>
<td>Action research, Conceptual analysis, Lessons learned, Conceptual analysis/mathematical, Case study, Field study, Laboratory experiment (human subject), Laboratory experiment (software), Interview, Descriptive/exploratory survey, Other, N/A</td>
<td>Captures the research method used to evaluate the technology.</td>
<td>RQ2, RQ3</td>
</tr>
<tr>
<td>5</td>
<td>Context described</td>
<td>Strong, Medium, Weak</td>
<td>Specifies the degree to which the context of the study is described.</td>
<td>RQ3</td>
</tr>
<tr>
<td>6</td>
<td>Study design described</td>
<td>Strong, Medium, Weak</td>
<td>Specifies the degree to which the design of the study is described.</td>
<td>RQ3</td>
</tr>
<tr>
<td>7</td>
<td>Validity discussed</td>
<td>Strong, Medium, Weak</td>
<td>Specifies the degree to which the validity of the study is discussed.</td>
<td>RQ3</td>
</tr>
<tr>
<td>8</td>
<td>Subjects</td>
<td>Practitioner, Researcher, Student, Not mentioned</td>
<td>Specifies who uses the technology in the evaluation.</td>
<td>RQ2, RQ3</td>
</tr>
<tr>
<td>9</td>
<td>Context</td>
<td>Academia, Industry</td>
<td>Specifies the context in which the evaluation is made.</td>
<td>RQ2, RQ3</td>
</tr>
<tr>
<td>10</td>
<td>Scale of evaluation</td>
<td>Toy example, Down-scaled real example, Industrial, Not mentioned</td>
<td>Specifies the scale on which the evaluation is made.</td>
<td>RQ2, RQ3</td>
</tr>
</tbody>
</table>

*Context described* (5) is scored on a three-level scale where *Weak* implies that the context relevant for the study is not mentioned in the paper at all. The property is marked as *Medium* when the organization/company and development effort is
presented in brief, while Strong involves characterizing the development effort/organization. Characterization involves development mode, e.g. contract driven, market driven etc., development speed, e.g. short time-to-market, company maturity, e.g. start-up, market leader etc.

The Study design described (6) property captures the degree to which the design of the study is described. This involves presenting the variables measured, the treatments, the control etc. The Weak value indicates no description at all of the design, while Medium indicates a brief description of how the study was carried out, e.g. “ten students did step 1 step 2 and step 3”. Strong study design indicates that the selection/sampling of subjects, variables measured and so on is present. For example, “ten students were randomly assigned to two groups in a one-factor, two-treatments experimental design”. The Validity described (7) property captures how the validity of the studies is presented. The validity of a study is important as it divulges how valid the results are for the population of interest [31]. In this case, Weak indicates that the validity of the study is not discussed at all, while Medium refers to the case in which the author has mentioned the validity but not discussed it in detail, e.g. “The action research approach means that claims for the generalizability of our findings are limited” [32]. The description of validity is considered to be Strong when it is discussed according to classification schemes like the ones presented in [31, 33] or the equivalent.

In addition to Research method (4), Subjects (8), Context (9) and Scale of evaluation (10) are also related to the RQ2 (the properties are influenced by [3]).

Subjects (8) captures the subjects that use the technology in the evaluation. A case study of a real development effort where practitioners use the technology under evaluation would imply that the practitioners are the subjects. On the other hand, if a researcher takes part in a real development effort and the researcher himself uses a new technology to see whether it can be used in real software development, the subject would be noted as being the researcher.

Context (9) captures the context in which the study has been carried out and is coarse grained in that it distinguishes only between academia and industry.

Finally, Scale of evaluation (10) captures what type of application the technology was used on in the evaluation. Scale of evaluation (10) ranges from Toy examples used for student projects and small examples to Industrial scale applications.

4. Execution

The execution1 of the review involved searching the Requirements Engineering journal (REj) using the Inspec database and manually searching the issues not indexed in the database. The reason for including papers exclusively from REJ is that it is the premier publication venue for researchers in requirements engineering. Thus, it should contain the most mature research in the field, implying a higher degree of empirical evidence and higher quality than research presented in workshops or conferences. This makes the selection suitable for answering the research questions posed here. The database

1 The search was carried out on June 23 2008 and two of the articles were published as online first articles. Inspec indexed articles from issue 1:1 up until issue 12:2. The rest of the issues were searched manually.
search resulted in the identification of 181 papers for possible inclusion in the review. This high number found by the search in comparison to the total in the journal is expected, as papers published in REJ most often discuss requirements engineering topics. Approximately half of the papers remained for data extraction when the inclusion criteria had been applied to the title and abstract of the papers. Upon completing the search, it was noted that Inspec did not index all issues in REJ. Thus, the non-indexed ones were searched manually.

To test whether the search or inclusion criteria missed relevant papers, ten percent of the issues were randomly selected and the full paper was read. No relevant papers were found that had not already been included, indicating that the inclusion/exclusion criteria were robust.

Of the 99 papers considered for data extraction, two were removed from the study when the full paper was read on the basis of their not having presented an evaluation of a requirements engineering technology. In total, 97 papers were classified in which 125 studies were identified and classified.

5. Threats to validity
The main threats to validity of this study are publication and selection bias, and data extraction, each detailed below.

5.1. Publication and selection bias
Excluding papers not published in REJ limits the possibility of generalizing the results as it is only one of the forums in which research in requirements engineering is published. Moreover, there is a chance of missing technologies and evaluations when other publication venues are not considered, e.g. other journals, conferences, technical reports etc. However, as stated in Section 4, REJ is the premier publication venue for researchers in requirements engineering and should thus include the most mature technologies and evaluations.

Another threat to the validity is the selection of papers from the journal. First, the key words used to search the publication database could miss papers relevant for inclusion in the review. As few papers (27) were removed by the search, however, it is reasonable to assume that this threat is limited. Second, the inclusion criteria used to include papers in the review are based on a reading of the abstract. This introduces a threat, as abstracts might not reflect the actual contents of the papers. This threat was investigated, as described in Section 4, and found to be limited.

5.2. Data extraction
A potential threat to validity is that the judgments used to include/exclude and to classify papers is biased. To limit this threat, the process used to include/exclude papers and the classification scheme was pilot tested prior to the study. The classification scheme itself is also derived from the research questions and based on previous studies [23], which limits the threat.

One threat to validity inherent to classifying research presentations is that the classification will be of the presentation instead of the actual research performed. This is not a primary concern, however, as the presentation of the research is of central
importance since the support for technology transfer is dependent on the descriptions of the work in the publications. In addition, not presenting the details used in a study limits the possibilities for other researchers to replicate the studies and thus directly affects the credibility of the research.

Another potential problem in reviewing research literature from the perspective of technology transfer is that research papers are written for researchers and not practitioners. However, requirements engineering is an engineering field, speaking towards application, meaning that research presented should not be too far from reality or use. Therefore, we feel that it is feasible to evaluate the research in this way.

With respect to the actual data extraction, the classification of papers is prone to some subjective variations owing to the inherent complexity of classifying software engineering research. There may be several plausible classifications for one paper [23], meaning that, if the study is replicated, the results might vary to some degree. This is primarily because the classification scheme is not mutually exclusive. However, in the cases in which several different classifications were possible, the papers were classified favorably with respect to the subsequent analysis. This means that the results presented in this systematic review are in some sense a best-case scenario, i.e., gives the researchers the benefit of the doubt.

6. Results and analysis

This section presents the results of the classification and is arranged in the order of the research questions presented in Section 3.1.3.

6.1. RQ1: What RE technologies exist?

A plethora of technologies was found, presented in Appendix A together with the number of papers discussing each technology (given in the column denoted “#”). The indentation in front on the technologies in Appendix A indicates that the technology is a refinement of the technology specified on the row above it (e.g. “--Cost-value approach” is a refinement of “AHP”). This division and sorting is not an invention of this review but is rather taken from the papers included in the review.

The papers are classified such that several technologies can be noted for each paper. This was necessary as several papers presented evaluations that used a synthesis of technologies. Only the technologies that are the focus of each paper is noted as being discussed, i.e. counted. This means that technologies that have refinements in Appendix A might be marked as N/A (see e.g. Goal-orientation).

In Appendix A, it can be seen that few of the technologies are discussed in more than one or two papers. The technologies that have received the most attention are Language extended lexicon (LEL) [34-37] and Use cases [38-42], with four and five publications on each technology, respectively. In total, 101 technologies were found in a total of 97 papers included in the review.

6.1.1. RQ1.1: What RE sub-process areas do the technologies target?

The requirements engineering sub-process area(s) targeted by the technologies treated were seldom explicitly stated in the papers and thus had to be mapped. Technologies often address several different requirements engineering sub-process areas, which was
noted when marking this property. Table 5 shows that technologies aimed at the sub-process area of *Analysis and Negotiation* has by far the largest representation (44 technologies), followed by *Specification* (30 technologies) and *Elicitation* (23 technologies). It is notable that technologies for *Management* have the lowest representation.

**Table 5: Requirements engineering sub-process areas addressed**

<table>
<thead>
<tr>
<th>Sub-process area</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis &amp; Negotiation</td>
<td>44</td>
</tr>
<tr>
<td>Elicitation</td>
<td>23</td>
</tr>
<tr>
<td>Management</td>
<td>9</td>
</tr>
<tr>
<td>N/A</td>
<td>3</td>
</tr>
<tr>
<td>Specification</td>
<td>30</td>
</tr>
<tr>
<td>Validation</td>
<td>19</td>
</tr>
</tbody>
</table>

### 6.1.2. RQ1.2: What is the timeline focus of the technologies?

Looking at the perspective of timeline, most technologies are developed for use *In-project* (i.e. for use in already started projects), as can be seen in Table 6. Not a single technology is meant to be used *Post-project* and only nine technologies for use in *Pre-project RE*.

The property of timeline was seldom mentioned explicitly in the papers. The technologies were classified as *In-project* if the understanding of the system under construction was perceived as being rather mature and as *Pre-project* if pre-project work was mentioned in the paper. If neither of these were fulfilled, the paper was classified as *N/A* as no article addressed requirements engineering technologies used after the project was concluded. Most of the articles that were classified as *Pre-project* had a distinct market driven requirements engineering perspective (see e.g. papers [43] and [44]).

**Table 6: Timeline focus of the technologies**

<table>
<thead>
<tr>
<th>Timeline focus</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-project</td>
<td>9</td>
</tr>
<tr>
<td>In-project</td>
<td>64</td>
</tr>
<tr>
<td>Post-project</td>
<td>0</td>
</tr>
<tr>
<td>N/A</td>
<td>24</td>
</tr>
</tbody>
</table>

**Summary and Discussion of RQ1:** To summarize the findings for RQ1 and its sub research questions, one can notice that researchers in requirements engineering more often than not develop new technologies that they evaluate and then move on to develop new ones. This is noted in Section 6.1, where few technologies have several studies that are related to them. This may imply that new technologies are presented once and that replication by other authors or additional evaluations (e.g. reporting industry trials etc.) are not common.

Looking at the sub-process area addressed by the technologies, researchers tend to focus on technologies to be used in a project that is already started (in-project). This could potentially be a problem, as uncovering requirements prior to projects is difficult, and failure to do so makes the choice of subsequent analysis tools irrelevant.
[4]. From a market driven perspective, pre-project RE is critical and many companies working with long-term product development (e.g. automotive, automation, telecommunication and so on) are completely dependent on pre-project RE to decide what requirements they should include in a release [45, 46]. This perspective is not the focus at all of the technologies presented. This is confirmed by the low focus on management technologies, as management of large-scale RE is a central part of market driven RE [47, 48].

6.2. RQ 2: What is the state of technology evaluation?

Assessing the state of technology evaluation involves examining several different properties of the evaluations presented in the papers. The properties are research method used, context in which the evaluation takes place, subjects that take part in the evaluation and the scale of the evaluation. The results in these four properties are all related to the evidence produced and thus related to decision support enabling technology transfer.

6.2.1. RQ2.1: What research methods are most common in the evaluation of technologies?

Looking at the results for research method (see Table 7), it can be seen that the most common research methods used to evaluate technologies are conceptual analysis (39%) followed by case studies (35%). In relation to earlier studies, conceptual analysis has been found to be the most commonly used research method in software engineering [23]. An interesting point is that our study shows a very large number of case studies. The reason might be that this study focuses on evaluations of technologies.

The results also show a relatively high number of experiments with human subjects in relation to previous studies [23].

<table>
<thead>
<tr>
<th>Research method</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action research</td>
<td>5</td>
<td>5%</td>
</tr>
<tr>
<td>Case study</td>
<td>34</td>
<td>35%</td>
</tr>
<tr>
<td>Conceptual analysis/Assertion</td>
<td>38</td>
<td>39%</td>
</tr>
<tr>
<td>Interview</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Laboratory experiment (human subjects)</td>
<td>14</td>
<td>14%</td>
</tr>
<tr>
<td>Laboratory experiment (software)</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Descriptive exploratory survey</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Field study</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Conceptual analysis/mathematical</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>N/A</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>97</td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

The research methods used in the majority of the evaluations all provide evidence with different traits. Conceptual analysis is an ad-hoc research method and thus does not provide realistic evaluations or control of the variables studied. This makes it difficult to evaluate and use the evidence produced to make technology transfer decisions. Experiments with human subjects generally have a high level of control of the variables studied. High level of control often amounts to trustworthy evidence and
understandability of cause-effect relationships. However, the price of control is often a lack of realism [3, 6, 7], as the setting of the experiment is artificial (often a research laboratory). Case studies offer the possibility of realism if conducted in a real setting (e.g. in industry).

6.2.2. RQ2.2: In what context are the technologies evaluated?

The contexts in which the technologies are evaluated were classified as either *Academia* or *Industry*. This classification is made to distinguish the evaluations performed in industry from the ones performed in academia, as one major point of technology transfer is to transfer results produced in one context to another. If the evaluation presented is done in industry, it increases the possibility of transferring it to another industry context [12].

The results for context (see Table 8) show that a significant number of evaluations take place in industry (37%), even though most of the evaluations are carried out in academia (63%).

<table>
<thead>
<tr>
<th>Context</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academia</td>
<td>61</td>
<td>63%</td>
</tr>
<tr>
<td>Industry</td>
<td>36</td>
<td>37%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>97</td>
<td><strong>100.00%</strong></td>
</tr>
</tbody>
</table>

6.2.3. RQ2.3: What subjects are used in the evaluation?

The similarity of the subjects participating in the evaluation with those envisioned to use the technology affects the ease of technology transfer [19]. With respect to subjects performing the tasks in the evaluation, Table 9 shows that only 19% of the evaluations involve industry practitioners.

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not mentioned</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Practitioner</td>
<td>18</td>
<td>19%</td>
</tr>
<tr>
<td>Researcher</td>
<td>66</td>
<td>68%</td>
</tr>
<tr>
<td>Student</td>
<td>12</td>
<td>12%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>97</td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

The most common scenario is that in which the researcher him/herself performs the evaluation (68%), for example uses the technology on either invented data or data from industry.

The reason might be that it is difficult to achieve industry trials (they are associated with risk and cost for the company). However, it may also mean that researchers have a hard time convincing the company of the benefits of using the new technology and thus a hard time gaining commitment to participation in the evaluation.

Cost and risk might also influence researchers not to use practitioners in evaluations, as carrying out evaluations in industry introduces higher risks and lead times for the research itself. That is, researchers might not be able to publish studies that do not show positive results and it takes considerable time to follow a real project.
6.2.4. RQ2.4: What scale do the evaluations have?
The scale of the evaluations also affects the evidence produced by the evaluations. Some factors might be impossible to evaluate in small-scale evaluations, e.g. scalability [6]. To provide evidence of usefulness in a real setting, evaluations must have a more realistic scale. In this study, the scale of evaluations is investigated by looking at the scale of the applications used in the evaluations.

Table 10 shows that most evaluations consist of applying technologies to Toy examples (46%), followed by Industrial applications (41%).

<table>
<thead>
<tr>
<th>Scale</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down-scaled real example</td>
<td>12</td>
<td>12%</td>
</tr>
<tr>
<td>Industrial</td>
<td>40</td>
<td>41%</td>
</tr>
<tr>
<td>Toy example</td>
<td>45</td>
<td>46%</td>
</tr>
<tr>
<td>Total</td>
<td>97</td>
<td>100%</td>
</tr>
</tbody>
</table>

6.2.5. RQ2.5: What degree of realism do the evaluations have?
The realism offered by the evaluations is evaluated by considering the combination of research method, context, subjects and scale of the evaluations. These combinations, together with the number of papers that fit into each combination, are presented in Table 11.

<table>
<thead>
<tr>
<th>Research method</th>
<th>Context</th>
<th>Subjects</th>
<th>Scale</th>
<th>#</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action research</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Action research</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>20</td>
<td>21%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Researcher</td>
<td>Down-scaled</td>
<td>7</td>
<td>7%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Researcher</td>
<td>Industrial</td>
<td>3</td>
<td>3%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Industry</td>
<td>Researcher</td>
<td>Down-scaled</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Industry</td>
<td>Researcher</td>
<td>Industrial</td>
<td>4</td>
<td>4%</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>Industry</td>
<td>Researcher</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>8</td>
<td>8%</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Researcher</td>
<td>Down-scaled</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Researcher</td>
<td>Industrial</td>
<td>6</td>
<td>6%</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Case study</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>8</td>
<td>8%</td>
</tr>
<tr>
<td>Case study</td>
<td>Industry</td>
<td>Researcher</td>
<td>Down-scaled</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Case study</td>
<td>Industry</td>
<td>Researcher</td>
<td>Industrial</td>
<td>6</td>
<td>6%</td>
</tr>
<tr>
<td>Case study</td>
<td>Industry</td>
<td>NM</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Academia</td>
<td>Practitioner</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>9</td>
<td>9%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>3</td>
<td>3%</td>
</tr>
<tr>
<td>Laboratory experiment (SW)</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (SW)</td>
<td>Industry</td>
<td>Researcher</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Interview</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Toy example</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Interview</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Total</td>
<td>97</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The distribution in the table suggests that:

- The most frequently occurring combination of research method, context, subject and scale is conceptual analysis applied to a toy example by the researcher in academia (shown as A in Table 11).
- Of the 34 case studies given in the table, only eight are carried out in a real setting, utilizing industrial applications and practitioners as subjects (shown as B in Table 11).

**Summary and Discussion of RQ2:** There seems to be a lack of in-depth studies on the actual use of technologies in real-life evaluations. Even though several studies are presented as case studies, only a few (eight of 34) use practitioners as subjects in a real environment and with industrial scale applications. This lack of reality in these studies could potentially hamper technology transfer, as it makes evidence more difficult to interpret. If papers presenting an evaluation consisting of a technology applied to a toy example by a researcher in academia provides the researcher with as much merit as a case study using practitioners in industry, there is no clear incentive for researchers to carry out the more time consuming research [49].

This is not to say that all evaluations should be case studies performed in industry but rather that different aspects of the technology may require different types of evaluation. For example, the correctness of the results produced by a method or tool might be validated by proofs of correctness or in an experiment in academia. A case study conducted by researchers using e.g. interviews with practitioners could be a step in an initial evaluation of a technology.

Issues such as the usability and usefulness of a technology, as perceived by potential users, must ultimately be evaluated in industry using practitioners. This is especially relevant regarding usefulness, i.e., the degree to which the technology adds value when used: in essence, ROI and comparison with the best alternative investment.

Looking at the papers in this review, there does not seem to be a spread; rather, a clear majority of the evaluations was made by researchers (68%) and most made on the scale of downscaled or toy examples (46+12=58%). This result is not encouraging from a technology transfer perspective, as it means that the decision support material offered to practitioners has its basis far removed from a real setting. This is especially surprising in an engineering field that per its very definition should be applied. At least, the final products of research in the form of technologies should have the goal of being applied.

**6.3. RQ 3: To what extent does the state of research support the actual adoption of technologies?**

To analyze the support for technology transfer offered in the papers, the evidence in Table 11 is mapped to the technology transfer model in Section 2.1 and visualized in Figure 3. The papers are classified as dynamic and static evaluations in industry and evaluations in academia. A distinction is also made between evaluations carried out with practitioners as subjects and without (shown as the top and bottom part of the circles in Figure 3). Articles classified as dynamic evaluations in industry consist of either case studies, action research or lessons learned. These should also be applied in industry on industry scale applications. Other research methods applied in industry are
classified as static evaluations. Studies performed in academia are all mapped to evaluations in academia.

Figure 3 shows that 21 evaluations are classified as dynamic evaluations in industry (top left part of Figure 3), of which 11 were carried out using practitioners as subjects (labeled A in Figure 3).

As previously noted, the clear majority of the papers presents evaluations performed in academia (shown in the bottom part of Figure 3). A minority of the papers is found to be static evaluations in industry (shown as C and D in Figure 3), which is somewhat surprising as it is the natural step before carrying out larger scale evaluations. For evaluations to provide support for technology transfer, they need to offer evidence valued by practitioners and be performed in a relevant environment [1, 2, 12, 13, 19]. This implies that the 11 papers labeled A in Figure 3 potentially offer the greatest support for technology transfer. These papers are further detailed in Table 12, which shows the degree of description of context, design and validity of the study.
Table 12: Dynamic validation in industry (depicted A in Figure 3)

<table>
<thead>
<tr>
<th>Article</th>
<th>Technologies</th>
<th>Research method</th>
<th>Context described</th>
<th>Study design</th>
<th>Validity discussed</th>
</tr>
</thead>
<tbody>
<tr>
<td>[50]</td>
<td>Scenarios, usability tests</td>
<td>Action research</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>[51]</td>
<td>Finite state machine</td>
<td>Action research</td>
<td>Strong</td>
<td>Medium</td>
<td>Weak</td>
</tr>
<tr>
<td>[45]</td>
<td>Requirements Abstraction Model (RAM)</td>
<td>Lessons learned</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
</tr>
<tr>
<td>[52]</td>
<td>Participatory design</td>
<td>Case study</td>
<td>Medium</td>
<td>Weak</td>
<td>Weak</td>
</tr>
<tr>
<td>[44]</td>
<td>AHP, Disagreement charts, Distribution charts, Influence charts, Satisfaction charts</td>
<td>Case study</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>[36]</td>
<td>Language extended lexicon (LEL)</td>
<td>Case study</td>
<td>Strong</td>
<td>Medium</td>
<td>Weak</td>
</tr>
<tr>
<td>[53]</td>
<td>RM-Tool</td>
<td>Case study</td>
<td>Weak</td>
<td>Medium</td>
<td>Weak</td>
</tr>
<tr>
<td>[54]</td>
<td>MAGERIT, SIREN</td>
<td>Case study</td>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
</tr>
<tr>
<td>[55]</td>
<td>Misuse oriented quality requirements engineering (MOQARE)</td>
<td>Case study</td>
<td>Weak</td>
<td>Medium</td>
<td>Weak</td>
</tr>
<tr>
<td>[43]</td>
<td>Requirements Abstraction Model (RAM)</td>
<td>Case study</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>[56]</td>
<td>Knowledge based Approach for the Selection of Requirements Engineering Techniques (KASRET)</td>
<td>Case study</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
</tbody>
</table>

Looking at Table 12, only five papers of the 11 offer somewhat strong support for technology transfer. That is, the context of the study is described to the point where practitioners can evaluate and compare it to their own environment, the study design is described to a degree at which one can understand how the results have been produced and the validity of the study has been discussed to unveil any circumstances that may affect the results. Four papers are classified as being strong in all of these three properties, and one further paper deviates slightly only in one property. On further investigation of the papers, one of the papers presenting RAM [45] was found to describe a static evaluation and was removed from further investigation. Table 13 provides brief descriptions of the papers identified to support technology transfer fully; for detailed information about the studies, the reader is referred to the original papers.
Table 13: Brief description of papers found to support technology transfer

<table>
<thead>
<tr>
<th>Context</th>
<th># of cases</th>
<th>Brief description of technologies</th>
<th>Results in short</th>
</tr>
</thead>
<tbody>
<tr>
<td>[50] A Danish manufacturer of sound and vibration measurement</td>
<td>1</td>
<td>Scenarios, including description of user tasks and usability tests with daily user tasks based on screen mock-ups</td>
<td>The number of usability problems per screen is reduced by 70% and the number of implementation defects per month is reduced by more than 20%. Sales figures showed that the product sells twice as many units and at twice the unit price.</td>
</tr>
<tr>
<td>[44] A Swedish CASE tool developer. Geographically distributed stakeholders that take part in prioritizing high-level requirements</td>
<td>1</td>
<td>Four different charts for visualizing and interpreting prioritization data</td>
<td>The distributed prioritization is useful and the visualization charts are a valuable decision support.</td>
</tr>
<tr>
<td>[43] A Swedish software and hardware developer of navigation, control, fleet management and service for automated guided vehicles. An international developer of power and automation technologies</td>
<td>2</td>
<td>Requirements Abstraction Model (RAM), a model for working with requirements on several levels of abstraction</td>
<td>Indications of that implementation of RAM has yielded substantial increases in both the accuracy of practices performed in requirements engineering and in requirements quality.</td>
</tr>
<tr>
<td>[56] A company developing real-time supply chain management solutions to optimize power networks</td>
<td>1</td>
<td>Knowledge-based Approach for the Selection of Requirements Engineering Techniques (KASRET), a decision support system for the selection of requirements engineering techniques for a specific software project</td>
<td>The majority of the projects members appreciated the high quality requirements gained from using the recommended techniques. The techniques selected were very helpful in reducing the delay of the project and improving the overall quality of the software specification.</td>
</tr>
</tbody>
</table>

Of the papers in Table 13, one presents more than one case in the evaluation, namely [43], which presents two cases of industry evaluation. However, both [56] and [50] present some information that shows the techniques were used on a wider scale than that presented in the papers. Two of the papers present investigations in a market driven context [43, 44].
6.3.1. RQ3.1: To what degree are the evaluations described?

Table 14 shows the combination of research methods and the degree to which the context, design and validity of the evaluations are presented in all of the papers.

<table>
<thead>
<tr>
<th>Research method</th>
<th>Context described</th>
<th>Study design</th>
<th>Validity discussed</th>
<th>#</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action research</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Action research</td>
<td>Strong</td>
<td>Medium</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Action research</td>
<td>Medium</td>
<td>Strong</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Action research</td>
<td>Weak</td>
<td>Strong</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Action research</td>
<td>Weak</td>
<td>Weak</td>
<td>Medium</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Medium</td>
<td>Weak</td>
<td>Medium</td>
<td>5</td>
<td>5%</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Medium</td>
<td>Weak</td>
<td>Weak</td>
<td>32</td>
<td>33%</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
<td>4</td>
<td>4%</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>Medium</td>
<td>Weak</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Case study</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>2</td>
<td>2%</td>
</tr>
<tr>
<td>Case study</td>
<td>Medium</td>
<td>Weak</td>
<td>Medium</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Case study</td>
<td>Medium</td>
<td>Weak</td>
<td>Medium</td>
<td>3</td>
<td>3%</td>
</tr>
<tr>
<td>Case study</td>
<td>Weak</td>
<td>Medium</td>
<td>Weak</td>
<td>10</td>
<td>10%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>7</td>
<td>7%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
<td>3</td>
<td>3%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Strong</td>
<td>Medium</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Medium</td>
<td>Strong</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Weak</td>
<td>Medium</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (SW)</td>
<td>Strong</td>
<td>Strong</td>
<td>Medium</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Laboratory experiment (SW)</td>
<td>Medium</td>
<td>Strong</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Interview</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td>Interview</td>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
<td>1</td>
<td>1%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td>97</td>
<td>100%</td>
</tr>
</tbody>
</table>

The distribution shown in Table 14 indicates that:

- Most of the papers utilizing conceptual analysis as a research method do not describe how the results are produced or their validity (labeled A in Table 14). This is expected, as there is no real reason to present these aspects in a basic evaluation. This also means that the evaluations offer slim decision support for technology transfer.

- Most of the case studies do not score strongly in all three properties in terms of the way in which the study is described. Only four papers (labeled B in Table 14) of 34 presenting case studies also presented the context, study design and validity of the study.

- Most of the experiments carried out with human subjects score strongly in all three properties for the level on which the study is described (labeled D in Table 14).
Summary and Discussion of RQ3: To summarize the findings, the technologies identified in Section 6.1 as receiving most attention (Language extended lexicon (LEL) [34-37] and Use cases [38-42]), in terms of leading to several publications on the same technology, are not found to be among the four strongest papers in relation to supporting technology transfer. Thus, it cannot be assumed that more attention on the part of researchers automatically translates to stronger technology transfer support in terms of more and better evaluations, and a better description of evaluation design, context and validity discussions. This is surprising, as more papers dealing with the same technology could be assumed to be an evolution in terms of more and better technology transfer support. However, the papers presenting Language extended lexicon (LEL) do provide a gradual refinement in terms of evaluation, as all three type of evaluations (see Figure 3) are covered by the four papers [34-37].

Only four of the 97 articles investigated in this study are found to offer strong evidence that supports technology transfer.

With respect to the way evaluations are presented, almost half do not describe how the evaluation was carried out, the context or the validity of the study. Only about one in ten studies has a clear description of all three aspects. This might not be remarkable in the case of general research papers, but it is important to remember that the papers investigated in this review are selected on the basis of the fact that they explicitly evaluate technologies. Furthermore, they are journal papers.

The research method that scores the highest in all three properties is laboratory experiments with human subjects. This may be because that the research field has some established guidelines for how to conduct and report experiments, e.g. [31] and [57].

6.4. The secondary studies

The classification process used facilitates an identification of all evaluations presented in the papers. The results presented so far have focused on the primary studies in the papers in order to make interpretation of the results easier for the reader. Studies are classified as primary if they are the focus of the paper or, if that cannot be determined, the study presenting the highest level of support for technology transfer is selected as the primary evaluation. Identifying all evaluations reported in the papers is important, as the purpose of the review is to identify support for technology transfer. All evaluations presented are candidates for providing technology transfer support. Thus, for completeness, this section presents the studies classified as not being primary in the review. Twenty-eight secondary evaluations were identified in the review and are presented in Table 15.

Table 15: Classification of secondary studies

<table>
<thead>
<tr>
<th>Research method</th>
<th>Context</th>
<th>Subjects</th>
<th>Scale</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>8</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>1</td>
</tr>
<tr>
<td>Conceptual analysis</td>
<td>Academia</td>
<td>Researcher</td>
<td>down-scaled real example</td>
<td>1</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Researcher</td>
<td>Industrial</td>
<td>2</td>
</tr>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 15 shows the research method, context, subjects and scale of the secondary studies. The studies are reported in 21 different papers. Ten studies are candidates for providing technology transfer support and thus influence the results given in Section 6.3, as the research method used is either case study, action research or lessons learned. However, on closer investigation, only one study would influence the results presented for RQ3, as this was carried out in a real environment with real subjects and an application of industrial scale. This study is the second case study presented in [43], which is noted in Table 13.

### 7. Conclusions

This paper presents a systematic literature review that examines all of the papers published in Requirements Engineering journal that contain any type of technology evaluation. The aim is to gauge the support for technology transfer (see the definitions in Section 2.1), i.e., the degree to which industry practitioners can use the reporting of technology evaluations as decision support for adopting the technologies in industry practice. This is done by evaluating the type of research performed in terms of research methodology, who performs the evaluations, where the evaluations are performed, and to what extent the design, context and validity are described.

The major findings of the review are:

- Only four of the 97 papers investigated provide strong technology transfer support (see Section 6.3). This was rather unexpected, as requirements engineering is, as the name indicates, an engineering field. In addition, the papers investigated are journal publications, which would indicate mature research with a focus on validations of some sort. On a possible positive note, all the papers that provide strong technology transfer support were written in the last seven years, which might indicate a positive trend.

- The majority of evaluations were performed by the researcher him/herself (see Sections 6.2.3 and 6.2.5). This limits the realism provided in evaluations and hence the support offered for technology transfer.

- As concerns the technologies presented, few have a pre-project timeline focus (see Section 6.1.2). Most technologies presented are meant to be used when the understanding of the system under consideration has become rather mature. Technologies used to acquire this understanding seem to be somewhat

<table>
<thead>
<tr>
<th>Case study</th>
<th>Industry</th>
<th>Practitioner</th>
<th>Industrial</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Case study</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
</tr>
<tr>
<td>Case study</td>
<td>Industry</td>
<td>Student</td>
<td>Industrial</td>
<td>1</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>2</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Academia</td>
<td>Student</td>
<td>Toy example</td>
<td>1</td>
</tr>
<tr>
<td>Laboratory experiment (HS)</td>
<td>Academia</td>
<td>Student</td>
<td>Industrial</td>
<td>1</td>
</tr>
<tr>
<td>Laboratory experiment (SW)</td>
<td>Academia</td>
<td>Researcher</td>
<td>Toy example</td>
<td>1</td>
</tr>
<tr>
<td>Interview</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Industrial</td>
<td>2</td>
</tr>
<tr>
<td>Action research</td>
<td>Industry</td>
<td>Researcher</td>
<td>Industrial</td>
<td>1</td>
</tr>
<tr>
<td>Survey</td>
<td>Industry</td>
<td>Practitioner</td>
<td>Not mentioned</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td>28</td>
</tr>
</tbody>
</table>
overlooked, thus potentially limiting the value of the technologies presented. In addition, none of the technologies investigated is meant to be used after a project has concluded (see Section 6.1.2). This may have implications for the progression in research, as there are e.g. few established ways to evaluate requirements engineering performance.

- With regard to the presentation of the studies, experiments with human subjects are on the whole well described (see Section 6.3.1). However, case studies that have the potential to provide more realism in evaluations are rather poorly described. Presenting the context, study design and validity of evaluation is important in case studies, as this aids practitioners in deciding, whether the results are transferable to another setting, and also evaluate the overall validity of any claims and conclusions.

- The scale on which the evaluations were performed is also an important aspect. Forty-one percent were on an industry scale, which would be positive from a technology transfer support perspective. However, this means that the rest (59%) were on a downscaled or toy example level. This can be explained to some extent in terms of an introduction of a new technology perhaps being best illustrated using a simple, small-scale example. Two factors might counter this argument, however. The first is that the papers investigated contained technology evaluations. Papers that purely presented a new technology were not included in the review at all. Second, as only journal publications were reviewed, one might argue that the maturity of the technology presented would have passed the “yet another technology” stage, where toy examples are the norm.

The aspects of technology transfer and decision support for practitioners to adopt technologies that are described and defined in this review are naturally somewhat simplified. Many aspects not covered, such as attaining management support and initiation threshold, are also relevant. However, the perspectives covered in this paper aim to describe a minimum level of support factors. These factors (e.g. description of context, design, validity), and the way in which the evaluations were performed in terms of subjects and scalability, are critical from both an academic and industry perspective. Thus, the findings of this review do have a number of implications for research:

- The major potential for improvement as concerns technology transfer support seems to be the subjects used in the evaluations. Not using industry practitioners as subjects excludes several important aspects of an evaluation. First, successfully convincing industry of participating in an evaluation speaks to the perceived relative advantage of a technology. Second, it also involves training practitioners, which highlights issues having to do with usability. There is an apparent risk of missing addressing these issues if researchers themselves make the evaluations.

- Regarding the time-line focus of the technologies, there is a tendency to focus on technologies to be used when projects have already started (in-project). The implication for research is whether this reflects the actual needs in industry. Looking at the studies identified in this review as providing the strongest technology transfer decision support, half have a more complete view, looking at pre-project RE also. The question is whether the industry commitment for validation and piloting was obtained as industry problems were better addresses by these studies.
The results indicate that empirical evidence from realistic evaluations in industry are scarce, but even more significant is the level of description of study design, context and validity as this is within the control of the researcher and not dependent on hard to obtain external resources. This limits the value of the research from an adoption perspective. Using guidelines such as those presented by Kitchenham [57] could improve the decision support material offered to practitioners. This would also be beneficial from an academic perspective as it improves the credibility of the research, and the ability to gauge the value of the contribution.

The scale of the evaluation is predominantly toy or down-scaled examples. The implication for research is that it is hard to judge whether the technology scales in relation to applicability in practice. Performing evaluations on a more realistic scale could enable better technology decision support material but also mean that unrealistic solutions are discarded earlier [6].

Looking at the evaluations, it is very difficult to attain good support for technology transfer by utilizing only one research method or, for that matter, making only one evaluation. A combination of methods, such as an experiment for control and a case study in industry for scalability and realism, would offer a combination of evaluation results superior to any single evaluation (see Figure 3). This combinatory approach of incremental evaluation and technology transfer would not only increase the support for technology transfer but would also improve the evaluations performed, both from a research and industry adoption perspective.

**Appendix A. Technologies identified in the review**

<table>
<thead>
<tr>
<th>Technology</th>
<th>Technology #</th>
<th>Technology</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHP</td>
<td>2</td>
<td>Problem frame</td>
<td>2</td>
</tr>
<tr>
<td>--Cost-value approach</td>
<td>1</td>
<td>--Alloy</td>
<td>1</td>
</tr>
<tr>
<td>Albert II</td>
<td>1</td>
<td>Promela</td>
<td>2</td>
</tr>
<tr>
<td>Analogy-based domain analysis (ABDA)</td>
<td>1</td>
<td>Rationale</td>
<td>1</td>
</tr>
<tr>
<td>Disagreement chart</td>
<td>1</td>
<td>Relationship analysis</td>
<td>1</td>
</tr>
<tr>
<td>Distribution chart</td>
<td>1</td>
<td>Release planning</td>
<td>N/A</td>
</tr>
<tr>
<td>Domain model</td>
<td>1</td>
<td>--Release Planner Provotype (RPP)</td>
<td>1</td>
</tr>
<tr>
<td>--Multi-Agent Domain Engineering Methodology (MADEM)</td>
<td>1</td>
<td>RENISYS</td>
<td>1</td>
</tr>
<tr>
<td>Domain reduction abstraction</td>
<td>1</td>
<td>Requirements Abstraction Model (RAM)</td>
<td>2</td>
</tr>
<tr>
<td>EPMcreate (EPM Creative Requirements Engineering Technique)</td>
<td>1</td>
<td>Requirements generation model</td>
<td>1</td>
</tr>
<tr>
<td>Entity relationship model</td>
<td>1</td>
<td>Requirements Similarity Analysis</td>
<td>1</td>
</tr>
<tr>
<td>Facilitator</td>
<td>1</td>
<td>Retrenchment</td>
<td>1</td>
</tr>
<tr>
<td>--Co-operative Requirements Capture (CRC)</td>
<td>1</td>
<td>RM-Tool</td>
<td>1</td>
</tr>
<tr>
<td>Fault tree analysis</td>
<td>1</td>
<td>Role activity diagram (RAD)</td>
<td>1</td>
</tr>
<tr>
<td>Feature dependencies</td>
<td>1</td>
<td>SAP Goal Map</td>
<td>1</td>
</tr>
<tr>
<td>Finite state machine (FSM)</td>
<td>1</td>
<td>Satisfaction chart</td>
<td>1</td>
</tr>
<tr>
<td>Fitness relationship</td>
<td>1</td>
<td>Scenarios</td>
<td>2</td>
</tr>
<tr>
<td>Gap typology</td>
<td>1</td>
<td>--Goal-Based Requirements Analysis Method</td>
<td>1</td>
</tr>
<tr>
<td>Generating Natural Language from UML (GeNLang/ML)</td>
<td>1</td>
<td>--Immersive scenario-based requirements engineering (ISRE) method</td>
<td>1</td>
</tr>
<tr>
<td>Goal-orientation</td>
<td>N/A</td>
<td>--Language extended lexicon (LEL)</td>
<td>4</td>
</tr>
<tr>
<td>--Assumption/commitment style reasoning</td>
<td>1</td>
<td>--Requirements Analysis Support System (NDRASS)</td>
<td>1</td>
</tr>
</tbody>
</table>
## Technology Transfer Decision Support in Requirements Engineering Research: A Systematic Review of REj

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Goal Oriented Requirements Analysis (AGORA)</th>
<th>Scenario-Based Requirements Analysis Method</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFR</td>
<td>Grounded Theory</td>
<td>Scenario-Based Requirements Analysis</td>
<td>1</td>
</tr>
<tr>
<td>i²</td>
<td>i² Framework</td>
<td>System Reliability Analyser (SRA)</td>
<td>1</td>
</tr>
<tr>
<td>Analysis</td>
<td>Analysis of Web Application Requirements</td>
<td>User Manual as Requirements Specification</td>
<td>1</td>
</tr>
<tr>
<td>Tropos</td>
<td>--Requirements Engineering Framework (REF)</td>
<td>Set Theory</td>
<td>2</td>
</tr>
<tr>
<td>RESCUE</td>
<td>--Scenario Advisor Tool</td>
<td>Soft Systems Methodology (SSM)</td>
<td>N/A</td>
</tr>
<tr>
<td>TCD</td>
<td>TCD Approach</td>
<td>SPIN</td>
<td>2</td>
</tr>
<tr>
<td>Influence</td>
<td>N/A</td>
<td>State Machine Hazard Analysis</td>
<td>1</td>
</tr>
<tr>
<td>Inspection</td>
<td>N/A</td>
<td>State-Charts</td>
<td>2</td>
</tr>
<tr>
<td>Defect</td>
<td>--Defect-Based Reading</td>
<td>STATEM</td>
<td>1</td>
</tr>
<tr>
<td>N-fold</td>
<td>N-fold Inspection</td>
<td>View-Charts</td>
<td>1</td>
</tr>
<tr>
<td>IBIS</td>
<td>Issue-Based Information Systems (IBIS)</td>
<td>UML</td>
<td>1</td>
</tr>
<tr>
<td>KAOS</td>
<td>N/A</td>
<td>Usability Tests</td>
<td>1</td>
</tr>
<tr>
<td>Obstacle</td>
<td>N/A</td>
<td>Use Cases</td>
<td>5</td>
</tr>
<tr>
<td>ReqMon</td>
<td>N/A</td>
<td>MissUse Case</td>
<td>N/A</td>
</tr>
<tr>
<td>Knowledge-based Approach for the Selection of Requirements Engineering Techniques (KASRET)</td>
<td>Requirements Pattern Language (RPL)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>LOTSOS</td>
<td>LOTSOS</td>
<td>SCORES</td>
<td>1</td>
</tr>
<tr>
<td>MAGERIT</td>
<td>MAGERIT</td>
<td>Use Case Map</td>
<td>1</td>
</tr>
<tr>
<td>MemReq</td>
<td>MemReq</td>
<td>Use Case Statecharts</td>
<td>1</td>
</tr>
<tr>
<td>MetaEdit+</td>
<td>MetaEdit+</td>
<td>User Interface and Requirements</td>
<td>1</td>
</tr>
<tr>
<td>Mismatch</td>
<td>Mismatch Handling for COTS Selection</td>
<td>---Misuse-oriented quality requirements</td>
<td>1</td>
</tr>
<tr>
<td>Handling</td>
<td>Sensitivity Analysis (MiHOS-SA)</td>
<td>engineering (MOQARE)</td>
<td></td>
</tr>
<tr>
<td>OOA</td>
<td>Multi-Area Feature Trees</td>
<td>View-Point</td>
<td>N/A</td>
</tr>
<tr>
<td>Multimedia</td>
<td>Multimedia Specification Process Model</td>
<td>MultiSpec</td>
<td>1</td>
</tr>
<tr>
<td>NFR</td>
<td>NFR-Classifier</td>
<td>--RECOCASE</td>
<td>1</td>
</tr>
<tr>
<td>Object</td>
<td>Object-Oriented Analysis</td>
<td>Viewpoint-Oriented Requirements</td>
<td>1</td>
</tr>
<tr>
<td>oriented</td>
<td>--Object-Oriented Method (FOOM)</td>
<td>Definition Method (VORD)</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>N/A</td>
<td>WinWin Negotiation</td>
<td>N/A</td>
</tr>
<tr>
<td>--Problem</td>
<td>--Problem Dependency Map</td>
<td>EasyWinWin Negotiation</td>
<td>N/A</td>
</tr>
<tr>
<td>--Conflict</td>
<td>--Conflict</td>
<td>Negotiation Model Checking</td>
<td>1</td>
</tr>
<tr>
<td>--Formal</td>
<td>--Formal Object-Oriented Method (FOOM)</td>
<td>CADiZ</td>
<td></td>
</tr>
</tbody>
</table>

## References


Chapter IV

A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations

Accepted with revisions to

Empirical Software Engineering.
A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations

M. Ivarsson¹, T. Gorschek²

¹Department of Computer Science and Engineering
Chalmers University of Technology
martin.ivarsson@chalmers.se

²School of Engineering, Blekinge Institute of Technology
tony.gorschek@bth.se

Abstract

One of the main goals of an applied research field such as software engineering is the transfer and widespread use of research results in industry. To impact industry, researchers developing technologies in academia need to provide tangible evidence of the advantages of using them. This can be done through step-wise validation, enabling researchers to gradually test and evaluate technologies to finally try them in real settings with real users and applications. The evidence obtained, together with detailed information on how the validation was conducted, offers rich decision support material for industry practitioners seeking to adopt new technologies and researchers looking for an empirical basis on which to build new or refined technologies.

This paper presents a model for evaluating the rigor and industrial relevance of technology evaluations in software engineering. The model is applied and validated in a comprehensive systematic literature review of evaluations of requirements engineering technologies published in software engineering journals. The aim is to show the applicability of the model and to characterize how evaluations are carried out and reported to evaluate the state-of-research.

The review shows that the model can be applied to characterize evaluations in requirements engineering. The findings from applying the model also show that the majority of technology evaluations in requirements engineering lack both industrial relevance and rigor. In addition, the research field does not show any improvements in terms of industrial relevance over time.

1. Introduction

The state of empirical evaluations performed in software engineering research has been criticized over the years [1-3]. The main arguments center around a lack of empirical evaluations [4, 5], poor study execution [3, 6] and a lack of realism when performed [7]. This may hamper progress in the field, as practitioners looking to adopt new technologies developed in academia are offered scarce decision support [8]. In an applied research field such as software engineering, the transfer and widespread use of research results in industry ultimately determine the relevance and success [7, 9]. Researchers should thus not only develop new technologies and advocate potential benefits but also base the information on the extent to which the technology has been...
validated (tested). A lack of realistic evaluations also limits the potential for practice to influence research, as evaluations carried out in industry provide feedback on what works in practice. In this context, a technology can be seen as any method, technique, procedure or model (or combination thereof) used in software development and maintenance [8, 10, 11].

Practitioners looking for new technologies must be able to evaluate the potential benefits and risks prior to adopting them. For the evidence to be convincing to industry, evaluations must be performed in a realistic setting, using realistic subjects and tasks, not using only toy examples, as practitioners value evaluations in settings comparable to their own environments [12]. The realism of evaluations thus determines the potential relevance of research results for industry. The trustworthiness of results must be considered in addition to relevance. The rigor of an evaluation and the way evaluations are presented influence the amount of trust placed in the evidence. Usefulness is limited without a description of how the results are obtained or in what context they are valid.

This paper presents a model for evaluating rigor and the industrial relevance of technology evaluations in software engineering. Traditionally, the relevance or impact of research results is associated with, for example, the number of citations [13]. However, the impact on research in the form of the number of publications or citations does not necessarily reflect success in the field. The model presented in this paper aims to balance this view of the relevance of research by evaluating the potential for impacting industry. The potential for impacting industry is evaluated by considering how the results are obtained and the realism of the setting [7, 14]. Practitioners value results obtained in settings similar to their own with research methods relevant in their environment [12]. In addition to relevance, the trustworthiness of the results presented is also considered. Rigor is evaluated indirectly by gauging to what extent and detail the context of a technology evaluation is presented by researchers. Failing to report aspects related to rigor makes it difficult to understand and use the results and limits the possibility for other researchers to replicate or reproduce the study [15].

The model can be used in several ways.

- It can be used to characterize and evaluate the level of rigor and relevance in a research field and how it has progressed over time.
- Practitioners looking for new technologies can use the results of the model to gauge the relevance of available evidence in a field and pinpoint studies potentially relevant for them.
- Researchers looking for an empirical basis on which to build new or refined technologies or studies to synthesize or replicate can use the results to find well described studies.
- The results of applying the model also highlight gaps or imbalances in research and can motivate the directions of future research.
- The model emphasizes aspects that are important in planning and reporting evaluations in software engineering and can thus be used preemptively when planning and reporting studies. For research to influence practice, the studies that are performed must produce results that are relevant to industry.

To make a pilot examination and provide an initial validation of the model, it is used as a basis for a systematic literature review [16] of technology evaluations in
requirements engineering. The review has two purposes and contributions. First, it is used to test and refine the model itself that is aimed at classification of rigor and relevance of technology evaluations. Second, the results that the model produces indicate the level of rigor and relevance of technology evaluations in requirements engineering, and how these have changed over time.

The paper is structured as follows. Section 2 introduces relevance and rigor in software engineering and presents related work. The model for evaluating rigor and relevance of technology evaluations is presented in Section 3. The design of the systematic review and the research questions addressed are given in Section 4, and threats to validity are discussed in Section 5. The results of the review are presented in Section 6, and the validation of the model is discussed in Section 7. Future work is presented in Section 8. Section 9 gives conclusions.

2. Background and related work

This section introduces relevance and rigor in software engineering in Sections 2.1 and 2.2, respectively, and explains the use of these terms in this paper. Section 2.3 presents related work on rigor and relevance.

A wide gap has been identified between what is practiced in industry and what is proposed by research [1, 17, 18]. At the heart of this gap is the transfer of research results into use in industry. This paper considers one potential factor governing this transfer by focusing on how technologies are evaluated in research. It also considers how these evaluations can be used by other researchers and, equally important, by industry professionals to gauge the relevance and validity of the evaluations. In short, can researchers get valid results they can build on and/or do industry professionals get enough decision support to dare to try out research results?

Different aspects of technology evaluations are given in Figure 1. Rigor refers to both how an evaluation is performed and how it is reported. If the study is not adequately described, the rigor of the evaluation cannot be evaluated by reviewers and other researchers. A background to and details regarding the rigor of software engineering research is given in Section 2.2. Relevance on the other hand refers to the potential impact the research has on both academia and industry. Academic relevance, or impact, is shown through the ability to publish papers and through citations by other researchers. Industrial relevance concerns the evaluation’s value for practitioners seeking to adopt technologies. To aid the transfer of technologies from academia to industry, research needs to provide evidence of the benefit of adopting research results (technologies). Here, evaluations can provide an incentive for practitioners to try new technologies. A background to relevance in software engineering is given in Section 2.1. Work related to rigor and relevance in software engineering is presented in Section 2.3.
2.1. Relevance

Several different measures have been used to evaluate academic relevance or impact in research, including the impact of individual researchers based on the number of publications [19] and different indexes (h-index [20], w-index [21] etc.) related to the number and distribution of citations of researchers or individual articles [13]. This indicates one aspect of the impact on research. However, quantity is only one aspect of impact and critics have argued that it should not be emphasized over quality or the value of research [22]. Assessing research quality or the value of research introduces subjective bias in both what to assess, e.g. what constitutes value and quality, and reviewers’ competence to assess it [23].

The impact on industry or the industrial relevance of research is even harder to gauge than academic relevance. One decisive factor for the relevance of research results is the actual topic studied [14]. Studying relevant topics increases the likelihood of the results being relevant to industry. However, the time perspective needs to be considered. Redwine and Riddle found that it takes in the order of 15-20 years to mature a technology to the stage that it can be disseminated to the technical community [24]. This makes it infeasible to use the actual uptake of research results in industry as the only evaluation tool. It also makes it difficult to distinguish relevant topics from irrelevant ones.

The industrial relevance of software engineering research has been questioned over the years [1, 2]. The ultimate goal of an applied research field such as software engineering is the transfer and use of results in industry. The actual impact on industry is not evaluated in this paper but rather the potential for impact. Zelkowitz [12] found that practitioners value research methods that are most relevant to their own environment, i.e. research methods that concern the performance of technology in live situations. Given the present state-of-research, research methods such as case studies and lessons learned have a greater potential to provide results compelling for industry since experiments are often conducted in unrealistic environments [25]. This means that evaluations carried out using these research methods should have a greater potential to provide compelling evidence to practitioners. However, the research method is just one aspect that influences relevance. It has also been argued that an
increase in the realism of evaluation is central to producing results that support a transfer to industry [7, 14]. Evaluations being representative of the target to which to generalize to support the transfer of results [14]. Realism can be improved by making evaluations that more closely resemble industrial settings. Aspects that must be considered in order to increase realism are the scale, context and subjects used in the evaluation. Scale refers to the time scale and size of the application/use/test used in the evaluation. Using small examples over a short period of time might not show how the technology under evaluation would scale to an industrial setting [14]. Context refers to the environment in which the evaluation is carried out. In order for practitioners to evaluate the suitability of a particular technology, the context in which it has been used/tested must be described [8, 10]. This enables an understanding of whether the technology would be suitable for use in another context. If evaluations are made in an artificial setting, e.g. the classroom, it is difficult to value the results [14] and the environments in which they are valid. An artificial setting is also likely to influence the motivation and goals of the subjects involved in the evaluation, thus introducing threats to construct validity [15]. Subjects used in the evaluation have a potential impact on the results obtained as they might not behave in the way professionals would and thus not give results that are valid in industry [26, 27].

2.2. Rigor
Rigor in research often refers to the precision or exactness of the research method used; e.g. a controlled experiment often enables greater control over variables than a case study [15]. This is one way to view rigor, the precision of the research approach utilized. Rigor can also mean the correct use of any method for its intended purpose [28, 29], implying that there is a context or application in which certain methods are appropriate or applicable. Several studies have evaluated the rigor used in software engineering research, of which a selection is presented in Section 2.3. The aim of this paper is not to argue for a specific research methodology but rather to emphasize that the methodology used should be carried out in accordance with corresponding best practices. As seen in Figure 1, rigor consists of two components, the rigor of the study and the way it is presented. Reference literature for research methods common in software engineering is given in Table 1. These research methods all have different traits and procedures for collecting and analyzing data, and thus different criteria for what constitutes the rigor of a study. Experimentation and case study research in software engineering have been extensively discussed while surveys and action research in software engineering have received less attention. However, action research has been given much attention in the related research field of information systems research.

<table>
<thead>
<tr>
<th>Research method</th>
<th>Reference literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimentation</td>
<td>[15, 30, 31]</td>
</tr>
<tr>
<td>Survey</td>
<td>[32]</td>
</tr>
<tr>
<td>Case study</td>
<td>[33-35]</td>
</tr>
<tr>
<td>Action research</td>
<td>[36]</td>
</tr>
</tbody>
</table>

In addition to the references in Table 1, guidelines for how to carry out and report empirical work in software engineering can also be found in [3]. However, rigor not only relates to the way in which a study is carried out. If a study is not presented adequately, a reviewer or other reader cannot determine whether the study has been carried out in a rigorous way, and the rigor of the study thus becomes irrelevant as it is
not presented. Failing to report the study in an adequate way also limits the opportunity for other researchers to understand and replicate or reproduce the study [15]. This means that presenting studies in an adequate way is a common denominator for all research methods. A compilation of aspects that should be included in presentations of evaluative research is given in Table 2.

Table 2: Study presentation aspects

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related work, research goal, problem statement, hypothesis</td>
<td>Defines the problem/goal of the evaluation and how it relates to previous research</td>
<td>[3, 15, 33, 36, 37]</td>
</tr>
<tr>
<td>Context</td>
<td>Information about the context in which the evaluation is performed including e.g. the experience of the staff, development process used etc.</td>
<td>[3, 15, 33, 36, 37]</td>
</tr>
<tr>
<td>Study design</td>
<td>Describes the products, resources and process used in the evaluation e.g. population, sampling etc.</td>
<td>[3, 15, 33, 36, 37]</td>
</tr>
<tr>
<td>Validity, limitations</td>
<td>Discusses any limitations or threats to the validity of the evaluation including measures taken to limit these</td>
<td>[3, 15, 33, 37]</td>
</tr>
<tr>
<td>Study execution</td>
<td>Describes the execution of the evaluation including data collection, preparations etc. Any sidesteps from the design should also be described. The execution should describe aspects that ease replication.</td>
<td>[3, 15, 33, 36]</td>
</tr>
<tr>
<td>Analysis</td>
<td>A presentation of the analysis where the analysis model and tools are described, as are assumptions for these. In addition, information about significance levels and the applicability of tests should be discussed.</td>
<td>[3, 15, 33, 36, 37]</td>
</tr>
<tr>
<td>Presentation and interpretation of results</td>
<td>An interpretation of the analysis in relation to the hypothesis/problems addressed by the evaluation</td>
<td>[3, 15, 33, 36, 37]</td>
</tr>
</tbody>
</table>

Reference literature shows a high degree of coherence as to what aspects that are important to report in evaluative research. Guidelines for action research in [36] do not explicitly state the validity or limitations of a study, but several elements in the guidelines relate to similar concepts such as the credibility of data, degree of openness etc. [36]. Thus there seems to be a consensus as to what aspects it is important to address when research studies are presented.

**Related work**

Several studies have investigated how software engineering research is carried out with respect to relevance and rigor. A selection is shown in Table 3.
Table 3: Literature studies related to relevance and rigor in software engineering

<table>
<thead>
<tr>
<th>Scope</th>
<th>Selected software engineering journals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICSE, TSE, IEEE SW, ICSE [4]</td>
</tr>
<tr>
<td></td>
<td>Requir. Engineering Journal [38, 39]</td>
</tr>
<tr>
<td></td>
<td>Journal of Empirical Software Engineering [40]</td>
</tr>
<tr>
<td></td>
<td>Controlled experiments from selected journals and conferences [41]</td>
</tr>
</tbody>
</table>

**Relevance**

- Research approach and method used in software engineering are quite narrow.
- Illustration of the use in examples is the most prevalent evaluation method.
- The amount of validation is increasing.
- An increase in the number of empirical evaluations over time.
- Experience reports and pseudo controlled studies are the most common study types.
- Use of examples commonly employed as validations.
- Half of the evaluations use professionals as subjects.
- Evaluations of requirements technologies offer scarce support for practitioners looking to adopt technologies.
- Most evaluations are performed by the researcher himself/herself.
- The most used evaluation method is to apply a technology to an example.
- Toy examples are used in half of the evaluations.
- A majority of the evaluations are carried out in academia.
- Professionals are used frequently in evaluations.
- Long-term studies are missing.
- Narrow focus of topics empirically investigated.
- The majority of subjects used in experiments are students.
- The majority of applications used in the experiments are constructed for the purpose of the experiment or constitute student projects.
- The duration of tasks in the experiments are short.

**Rigor**

- The most common type of sampling used in evaluations is investigator sampling.
- No significant illegal use of analysis on scales of measurement.
- Great misuse of the term "case study".
- Self-confirmatory studies.
- The soundness of studies does not improve over time.
- The type of study performed is only defined in half of the cases.
- There is a lack of replicated studies.
- Experiments using human subjects are on average well described, i.e. present context, study design and the validity of the study.
- Except for when experiments are performed, few studies describe the context, study design and validity of the evaluations.
- Internal and external validity is often reported.
- Relatively low and arbitrary reporting of context variables.
- Reporting of validity is often vague and unsystematic.
- Most experiments do not relate to theory.
- The level of statistical power falls substantially below accepted norms.
To some extent, previous studies give mixed messages about the state of rigor and relevance in software engineering. In terms of relevance, the research method used has been investigated in several literature reviews. Glass et al. [4] point out that applying the technology to an example is the dominant research method. However, both Zelkowitz [38] and Zannier et al. [39] show that the number of validations, in contrast to pure theoretical or advocating research, has increased over time, indicating an improvement of the state of evaluations. The research method used also seems to depend on the publication venues included in the review. For example, in the Empirical Software Engineering (EMSE) journal, it is reported that experiments and case studies are the most often used research methods [40] while results in the Requirements Engineering journal (REj) show that illustrating technologies in an example is the most commonly used research method [8]. The results are again different when the International Conference on Software Engineering (ICSE) is considered, where experience reports and pseudo controlled evaluations are in the majority [39].

Regarding the subjects used in evaluations, and thus to some extent the realism of evaluations, it seems that the results depend on the research method used and the publication venue. For example, Höfer and Tichy [40] report that the majority of case studies uses professionals as subjects while experiments often utilize students. Sjöberg et al. [25] also found that subjects are often used in experiments in software engineering. Looking at the publication venue, most of the evaluations in REj are carried out by the researcher him/herself [8] while half of the evaluations presented in ICSE use professionals as their subjects.

Scale, which is also an aspect of relevance, has been found to be unrealistic. For example, considering experiments in software engineering, the applications used are often constructed for the purpose of the experiment and the duration of tasks performed in the experiments is short [25]. In addition, more than half of the evaluations reported in REj is performed in toy or down-scaled examples [8].

The rigor of evaluations in software engineering research has also been investigated. First, several studies have found that presentations of research show room for improvement [8, 41]. This includes failing to report both certain aspects, such as validity, study design etc., and the structure and contents of the presentation [8, 25, 39]. For example, Zannier et al. [39] found that it is common for researchers not to report the type of study that is carried out, and the REj review showed that aspects related to study design, validity and context are seldom reported except for in experiments.

Detailed aspects concerning the rigor of experiments carried out in software engineering have also been investigated. The actual rigor of experiments reported in software engineering seem to be lacking, as statistical power falls below accepted norms in related fields [6]. Only 6% of the studies investigated in [6] had a power of 0.80 or more. Achieving only a weak statistical power makes it impossible for researchers to draw conclusions from experiments.
3. A model for evaluating rigor and industrial relevance in technology evaluations

A model that captures rigor and the relevance of technology evaluations in software engineering research was developed to evaluate these dimensions. This section gives an overview of the model and details its constituents. The aim of the model is to enable a classification of individual evaluations in order to characterize research carried out in a field. Thus the model needs to be applicable to many types of evaluations, i.e. research methods. The way in which research is classified is described in Section 3.1. The resulting classifications are quantified and visualized to be able to characterize and understand technology evaluations in a given field of research. The quantification into variables for rigor and relevance is described in Section 3.2 and the visualizations employed are given in Section 3.3. Limitations of the model are given in Section 3.4.

3.1. Classifying research

To analyze the state of technology evaluations, studies are classified from the perspectives of relevance and rigor. The purpose is not to achieve an exact classification of each individual study but rather to give an approximate overview of the state and progress of research in order to identify patterns. Thus, a detailed in-depth analysis of studies does not necessarily add any benefit over a more simplified one. For example, several aspects might be considered when classifying the extent to which the context is reported, i.e., product, process, practices, tools, techniques, people, organization and market [42]. Classifying the reporting of context can then be done by considering the number of aspects reported in each paper. Using this detailed classification would lead to a precision in the results, i.e. a classification that can pinpoint what aspects are included or missing from reports. A high degree of precision in classification is not necessarily sought, however, as the goal of the model is to provide an overview of the state-of-research. A detailed classification would also lead to few (if any) papers being classified at the highest level (containing all aspects) [42]. To characterize state-of-research, the classification should relate to how research is currently performed and reported, considering only aspects that are often reported. If no papers include a particular aspect, the ability of that aspect to describe the state-of-research is limited. The evolution of the model to improve the ability to capture the state-of-research is described in Section 7.1. Instead of classifying details, a simplified way of classifying relevance and rigor using scoring rubrics is used in the model. Rubrics are one way to formalize the evaluation into different criteria and their levels [43]. They have been previously proposed and successfully used in Software Engineering for the evaluation of Master Theses [44]. The scoring rubrics used to evaluate rigor and relevance are presented in Sections 3.1.1 and 3.1.2 respectively.

3.1.1. Rigor

It is not the actual rigor of studies, e.g. use of a correct analysis method on the scales of measurement, appropriate sampling type etc., that is considered in this model but rather the extent to which aspects related to rigor are presented. Failing to report aspects related to rigor makes evaluation by reviewers or readers difficult. The focus is on the way in which aspects are reported enabling the classification of diverse study types; e.g. characteristics of rigor for an action research study are not the same as an experiment. Table 3 shows the aspects considered for evaluating rigor of studies and the scoring rubrics that was developed to guide the evaluation. Three aspects are considered in scoring rigor; the extent to which context, study design and validity are...
described (see Table 4). All these aspects are scored with the same three score levels; “weak”, “medium” and “strong” description. The scoring of each aspect is described in detail in Table 4.

Table 4: Scoring rubric for evaluating rigor

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Strong description (1)</th>
<th>Medium description (0.5)</th>
<th>Weak description (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context described</td>
<td>The context is described to the degree where a reader can understand and compare it to another context. This involves description of development mode, e.g., contract driven, market driven etc., development speed, e.g., short time-to-market, company maturity, e.g., start-up, market leader etc.</td>
<td>The context in which the study is performed is mentioned or presented in brief but not described to the degree to which a reader can understand and compare it to another context.</td>
<td>There appears to be no description of the context in which the evaluation is performed.</td>
</tr>
<tr>
<td>Study design described</td>
<td>The study design is described to the degree where a reader can understand, e.g., the variables measured, the control used, the treatments, the selection/sampling used etc.</td>
<td>The study design is briefly described, e.g., “ten students did step 1, step 2 and step 3”</td>
<td>There appears to be no description of the design of the presented evaluation.</td>
</tr>
<tr>
<td>Validity discussed</td>
<td>The validity of the evaluation is discussed in detail where threats are described and measures to limit them are detailed. This also includes presenting different types of threats to validity, e.g., conclusion, internal, external and construct.</td>
<td>The validity of the study is mentioned but not described in detail.</td>
<td>There appears to be no description of any threats to validity of the evaluation.</td>
</tr>
</tbody>
</table>

3.1.2. Relevance

The industrial relevance of an evaluation consists of two parts in the model. First, the realism of the environment in which the results are obtained influence the relevance of the evaluation. Three aspects of evaluations are considered in evaluating the realism of evaluations: subjects, scale and context. The scoring rubrics used to assess these are given in Table 5. The expertise or skill of subjects used in the evaluation is also likely to influence the results [14, 15, 26, 27]. As the reporting of background and the skill level of subjects used in evaluations varies substantially [25]; the only aspect included in the model differentiates whether subjects are practitioners, students or researchers. To address aspects such as the scalability and usefulness of technologies under evaluation, applications used in the evaluation must have a realistic scale [7, 14]. The scale aspect in the model refers to the type of application used in the evaluation and ranges from toy examples to industrial scale applications.
Finally, the context in which results are obtained determines the type of study. For example, a case study in industry is likely an on-line evaluation in a realistic setting, while a similar evaluation performed in academia is probably not.

Second, the research method used to produce the results influence the relevance of the evaluation. A diverse set of research methods is included in the model to cover a wide range of activities from application (test/illustration) of a technology to experiments and any sort of empirical evaluation. The simplest form of evaluation is chosen to be a technology applied to an example. The reason for including application examples as a type of evaluation in the classification is to provide a point of comparison in relation to other research methods. Excluding application examples would remove the ability to appreciate the proportions between different types of evaluations used in software engineering. In addition, researchers in software engineering often use example applications to “validate” or evaluate technologies. The scoring discerns research methods that contribute to relevance from ones that do not. This valuation is likely to evolve over time to reflect how research methods are valued by practitioners, for example, practitioners value studies carried out in the form of case studies or lessons learned. This can depend on the state of validation in software engineering, as experiments are often carried out using students experimenting with toy examples. If experiments more closely resembled the environment to which the results are generalized (often industry), this view might change, i.e. practitioners’ valuation of experiments could improve if the realism of experiments improved. Thus, scoring should change over time to correspond to how research is carried out. The scoring rubric used to classify research method is detailed in Table 5.
Table 5: Scoring rubric for evaluating relevance

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Contribute to relevance (1)</th>
<th>Do not contribute to relevance (0)</th>
</tr>
</thead>
</table>
| Subjects        | The subjects used in the evaluation are representative of the intended users of the technology, i.e., industry professionals. | The subjects used in the evaluation are not representative of the envisioned users of the technology (practitioners). Subjects included on this level is given below:  
  - Students  
  - Researchers  
  - Subject not mentioned |
| Context         | The evaluation is performed in a setting representative of the intended usage setting, i.e., industrial setting. | The evaluation is performed in a laboratory situation or other setting not representative of a real usage situation. |
| Scale           | The scale of the applications used in the evaluation is of realistic size, i.e., the applications are of industrial scale. | The evaluation is performed using applications of unrealistic size. Applications considered on this level is:  
  - Down-scaled industrial  
  - Toy example |
| Research method | The research method mentioned to be used in the evaluation is one that facilitates investigating real situations and that is relevant for practitioners. Research methods that are classified as contributing to relevance are listed below:  
  - Action research  
  - Lessons learned  
  - Case study  
  - Field study  
  - Interview  
  - Descriptive/exploratory survey | The research method mentioned to be used in the evaluation does not lend itself to investigate real situations. Research methods classified as not contributing to relevance are listed below:  
  - Conceptual analysis  
  - Conceptual analysis/mathematical  
  - Laboratory experiment (human subject)  
  - Laboratory experiment (software)  
  - Other  
  - N/A |

3.2. Quantification and measurement conversion

To analyze the resulting classifications in order to present an abstract view of the state of the technology evaluation, the classification is converted into numerical values. This conversion (from a nominal to an ordinal scale) adds information by placing a value on different classifications. Conversion from a weaker to a stronger measurement scale is usually not permissible. However, in this case, the goal is to add information to discern studies that have low rigor/relevance from studies that have higher levels. For example, using professionals in studies is a more realistic solution than using students [7, 14], and thus a higher value is assigned to studies that use professionals.

The value of the constituents is summed up to form variables for rigor and relevance that can be analyzed, as can be seen in Table 6 and Table 7. Generally, an addition of measures on the ordinal scale makes little sense. However, the variables should be interpreted as how many aspects contribute to industrial relevance for relevance (see Table 7) and how many aspects are described for rigor (see Table 6). For example, a study that is assigned a relevance value of “2” has two aspects that are classified as contributing to industrial relevance. This does not imply however that a study that has twice the value for relevance is twice as likely to influence industry. It only provides an approximation of how many aspects contributing towards relevance a study has.
The ability of this quantification to reflect the actual relevance might be argued. For example, this would rate the relevance of an experiment using students as subjects experimenting with a toy example in academia lower than an application example on an industrial scale carried out by a researcher in academia (as can be seen in Table 7). In this case, the experiment that uses students might produce more relevant results as it does not employ an ad hoc research method and is, for example, more likely to control for researcher vested interest. The variable for relevance only illustrates how many aspects of the study are realistic and provides an approximation of the relevance of studies carried out in software engineering.

Table 6: Quantification of rigor

<table>
<thead>
<tr>
<th>Rigor, Rigor = C + S + V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context described (C)</td>
</tr>
<tr>
<td>Study design described (S)</td>
</tr>
<tr>
<td>Validity discussed (V)</td>
</tr>
<tr>
<td>Each aspect is scored according to the following scheme</td>
</tr>
<tr>
<td>Weak presentation</td>
</tr>
<tr>
<td>Medium presentation</td>
</tr>
<tr>
<td>Strong presentation</td>
</tr>
</tbody>
</table>

Table 7: Quantification of relevance

<table>
<thead>
<tr>
<th>Relevance, Relevance = C + RM + U + S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context (C)</td>
</tr>
<tr>
<td>Research method (RM)</td>
</tr>
<tr>
<td>User/Subject (U)</td>
</tr>
<tr>
<td>Scale (S)</td>
</tr>
<tr>
<td>Aspects are scored as 1 if contributing to relevance, 0 otherwise.</td>
</tr>
</tbody>
</table>

The average of these variables is also used in the analysis. The average of relevance should be interpreted as how many aspects contributing to relevance are included on average in the studies. The average for rigor should be interpreted as the average number of aspects that are described in the studies. For example, if the average for rigor is “1”, this means that, on average, one aspect is fully described or two aspects are mentioned in the studies.

### 3.3. Visualization

Three different charts are used to visualize the resulting variables, where a bubble chart depicts the combination of rigor and relevance to characterize the type of research carried out in a research field. Bubble charts can be very effective in giving visual indications of how much research is carried out and categorizing research type [45, 46]. Figure 2 gives an example of this kind of chart, and indicates the amount of evaluations that end up with a specific combination of rigor and relevance. It can be seen, for example, that most studies (ten) in Figure 2 have zero rigor and relevance and two studies show the highest level of rigor and relevance.
A line chart showing how the average rigor and relevance have changed over time is used to analyze the evolution of research in a field. If relevance and rigor in a research field do not change over time, they exhibit a maturity flat line (labeled B in Figure 3). If they improve or deteriorate, they exhibit maturity growth (labeled C in Figure 3) or maturity degradation (labeled A in Figure 3). Signs of health in a research field would be a flat line of high maturity or a growth in maturity.

Finally, to investigate the influence of the publication venue on the rigor and relevance of the evaluations presented, the average rigor and relevance for the publication venue is considered. An example of this can be seen in Figure 4, where Journal 2 has high levels of relevance and rigor and Journal 3 has low levels. Researchers looking for relevant studies that are well described might then prefer Journal 2 over Journal 3.
3.4. Limitations

The goal of the model is to describe state-of-research and to identify studies with high levels of rigor and industrial relevance. Classifying research presentations is inherently an approximation as the classification is of the presentation as opposed to the actual research [4]. This introduces a number of limitations with respect to using the model:

- The results from the model do not consider the actual results of the evaluated studies. Studies ending up as having high levels of rigor or relevance might describe negative results. The results from using the model only present the level of rigor and relevance of studies included.
- The model does not consider the alignment of the technology being evaluated with the context in which it is being evaluated. For example, a technology developed to be used in large teams can be evaluated in small teams. It is up to the user of the results to value the contribution of the individual papers. The model only provides a way of discerning studies with different levels of rigor and relevance.
- Evaluations of technologies on industry data by the researcher him/herself is classified as having “researcher” as subject. In some cases this mean that the study get a low relevance score relative to its actual value for practitioners. This is a trade-off made in the model to avoid introducing subjective bias in the scoring and to keep the focus on identifying evaluations that take practitioners’ valuation of usability and usefulness into account.

4. Model Validation – Systematic Review Design

To validate the model presented in Section 3, it is applied in a full systematic literature review of technology evaluations in requirements engineering. This section gives a detailed account of the design of the systematic literature review. The review has two purposes. First, it illustrates how the model can be used to evaluate the rigor and relevance of technology evaluations in the field of requirements engineering (see e.g. [47]). Second, the results of the review give an overview of the state of rigor and relevance of technology evaluation in requirements engineering (see e.g. [48]). The review follows the guidelines proposed by Kitchenham and Charters [16]. The main deviation from the proposed procedures is the lack of study quality assessment [16], as all studies presenting a technology evaluation are included in the review. Thus, the
quality assessment is part of the inclusion criteria and scoping, i.e., the assessment of
the quality of the studies included is part of the data extraction procedure.

4.1. Research questions

The research questions addressed in this review are derived from the model presented
in Section 3 and are:

RQ1. What is the state-of-practice in relation to rigor and relevance of
technology evaluations in requirements engineering?
RQ2. Have rigor and relevance changed over time?
RQ3. Does publication venue influence rigor and relevance?

The first research question concerns characterizing the type of evidence produced in
evaluations in requirements engineering research (relevance) and how it is presented
(rigor). Relevance is characterized by the realism of evaluations. The research method
used, as well as subjects, context and scale of evaluations, determine the realism of the
evaluations. The presentation of evaluations as regards study design, validity and
context determines the level of rigor.

The second research question seeks to investigate whether there have been any
improvements with respect to relevance and rigor over time. This could indicate that
technologies and research in the area are maturing to the point where industry trials
and applications are more dominant. The last research question investigates the
influence of publication venue on relevance and rigor.

4.2. Inclusion criteria

This review aims to investigate the rigor and relevance of technology evaluations in
requirements engineering. The principal criterion for including a paper is thus that it
should present an evaluation of a requirements engineering technology. The set of
inclusion criteria is detailed below:

- The paper should be in the scope of requirements engineering. A requirement
describes a condition or capability needed by the user to solve a problem or
achieve an objective [49]. There is vagueness between requirements and
design, making it hard to define a clear inclusion criterion. However,
requirements primarily concerns what the systems should do,, not how to do it
[50]. Papers that focus on what the system should do are thus included, even if
they contain elements of design.
- The paper should present a technology. In this context a technology can be
anything from methods, techniques and procedures to models and tools (or
combinations thereof) [8, 10, 11].
- The paper should present an evaluation. Evaluation is here defined to cover a
wide range of activities from application (test/illustration) of a technology in a
toy example invented by the researchers themselves to experiments and any
sort of empirical evaluation.

These broad inclusion criteria make for an “include heavy” selection, thus avoiding
dismissing papers that have some sort of evaluation of a requirements technology.
4.3. Identification of articles

A database search was made to identify papers that should be included in the review. The search term was devised to find requirements engineering technologies that have been evaluated and has been used in a previous review [8] (see Table 8).

Table 8: Search term

<table>
<thead>
<tr>
<th>Population: requirement*, specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intervention: empiric* OR experience* OR &quot;lesson learned&quot; OR &quot;lesson learnt&quot; OR &quot;lessons learned&quot; OR &quot;lessons learnt&quot; OR evaluat* OR validation* OR experiment* OR stud* OR case* OR example* OR survey OR analys* OR investig* OR demonstrate*</td>
</tr>
</tbody>
</table>

The search term was applied to all journals classified as software engineering by ISI (listed in Table 9). The database search was done using Inspec and applied only to the title, abstract and keywords, as a full text search would cover too many irrelevant results [51]. The search was performed on October 16, 2008, and identified 3593 papers. To identify and extract papers relevant for the review, the title and abstract were read and compared to the inclusion criteria. This identified 455 papers that were included for data extraction. On the basis of reading the full paper, 349 papers were included and classified in the review. The number of papers in each journal that was included is given in Table 9.

Table 9: Journals included in the review

<table>
<thead>
<tr>
<th>Journal</th>
<th>Abbreviation</th>
<th>Papers included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications of the ACM</td>
<td>COMACM</td>
<td>4</td>
</tr>
<tr>
<td>Computer</td>
<td>COMPUTER</td>
<td>4</td>
</tr>
<tr>
<td>Computer Journal</td>
<td>COMPUTERJOURNAL</td>
<td>9</td>
</tr>
<tr>
<td>Empirical Software Engineering</td>
<td>EMSE</td>
<td>15</td>
</tr>
<tr>
<td>IBM Systems Journal</td>
<td>IBMJOURNAL</td>
<td>1</td>
</tr>
<tr>
<td>IEEE Software</td>
<td>IEEESOFTWARE</td>
<td>40</td>
</tr>
<tr>
<td>IEE Proceedings-Software</td>
<td>IEEPROC</td>
<td>12</td>
</tr>
<tr>
<td>International Journal of Software Engineering and Knowledge Engineering</td>
<td>ISEKE</td>
<td>19</td>
</tr>
<tr>
<td>Information and Software Technology</td>
<td>IST</td>
<td>37</td>
</tr>
<tr>
<td>Journal of Research and Practice in Information Technology</td>
<td>JRPIT</td>
<td>5</td>
</tr>
<tr>
<td>Journal of Systems and Software</td>
<td>JSS</td>
<td>32</td>
</tr>
<tr>
<td>Requirements Engineering journal</td>
<td>RE</td>
<td>97</td>
</tr>
<tr>
<td>Software Practice and Experience</td>
<td>SPE</td>
<td>7</td>
</tr>
<tr>
<td>Software Quality Journal</td>
<td>SQJ</td>
<td>6</td>
</tr>
<tr>
<td>ACM Transactions on Software Engineering and Methodology</td>
<td>TOSEM</td>
<td>8</td>
</tr>
<tr>
<td>IEEE Transaction on Software Engineering</td>
<td>TSE</td>
<td>53</td>
</tr>
<tr>
<td>Journal of the ACM</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Journal of Software Maintenance and evolution-Research and Practice</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>IEEE Transactions on Dependable and Secure Computing</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td>349</td>
<td></td>
</tr>
</tbody>
</table>
4.4. **Data collection**  
The papers included in the review were analyzed and classified according to the aspects and scoring rubrics in the model presented in Section 3. All aspects are classified according to what is mentioned in the articles that were reviewed. If the aspects are not explicitly mentioned in the paper, it is mapped according to the reviewer’s understanding. An article can describe several studies in which case the primary study, as portrayed by the paper, is classified. If that cannot be determined, the study most advantageous with respect to subsequent analysis is chosen. The data extracted were stored in a MySQL database for later analysis.

5. **Threats to validity**  
The main threats to validity in this study are publication and selection bias, and data extraction, each detailed below.

5.1. **Publication and selection bias**  
Including only papers from journals classified as by ISI as belonging to software engineering limits the possibility to generalize the results to other forums in which requirements engineering technologies are published. This also introduces the risk of missing technologies and evaluations published in conference proceedings, technical reports, workshops etc. However, as the major journals in the software engineering field are included in the review, this threat should be limited.

The selection of papers from journals is also a threat to validity. First, the search procedure used introduces a threat as it could miss papers relevant for inclusion in the review. However, the search terms used did not miss relevant papers when applied to REj [8]. Even if the search missed papers, it should not introduce any systematic bias with respect to the results. Second, the inclusion criterion is first applied by reading the abstract of the papers. This introduces a threat as the abstract does not necessarily reflect what is actually presented in the papers. This threat was investigated in REj and found to be limited [8].

5.2. **Data extraction**  
A potential threat to validity is the subjective judgment used to include/exclude papers and to extract data from the papers that were included. To limit this threat, a pilot trial was carried out with the classification scheme and inclusion criteria, and these were changed prior to use as described in Section 7.1. In addition, the aspects and scoring rubrics used for data extraction are derived from the research questions. With respect to the actual data extraction, the scoring of aspects is subject to subjective variations. For instance, there may be several plausible classifications for one paper. To limit this threat, papers were classified giving researchers the benefit of the doubt, i.e. papers were classified in accordance with what is mentioned in the papers. In addition, when several classifications are possible, the one most beneficial for the subsequent analysis was chosen. This means that the results presented in this review are in some sense a best case scenario, e.g. example applications by the researcher himself/herself are sometimes reported as “case studies”, giving a higher relevance score.
6. Results from using the model

This section presents the findings of the review and is arranged according to the research questions presented in Section 4.1. The quantification and visualization techniques from the model described in Section 3 are used to analyze the data from the review.

6.1. RQ1: What is the state-of-practice in relation to rigor and relevance of technology evaluations in requirements engineering?

The variables for rigor and relevance found in the articles included in the review are given in Figure 5. The size of the bubbles indicates the number of papers in each class. The maximum value for rigor a paper can have is three, while relevance has a maximum of four. The figure shows that the majority of evaluations end up in the lower left quadrant of the bubble chart, indicating a lack of both rigor and relevance. 116 articles have zero rigor and relevance. This means that about one third of all the evaluations included in this review are experiments in which aspects related to rigor are not described or are examples of application of a technology done by either students or researchers in academia in toy examples. This is disappointing from an academic perspective as it is difficult to synthesize and gain a better understanding of how technologies actually perform if there is no actual evaluation or reporting of evaluations. Example applications often do not evaluate the usefulness or usability of a technology but rather simply illustrate that it can be applied to an example. The concentration of evaluations ending up in the left half of the chart is also disappointing from a technology transfer perspective, as these evaluations have less potential for actually influencing practice. Furthermore, few evaluations landing in the upper part of the chart illustrate that aspects needed to evaluate the quality (rigor) of evaluation are often lacking in the reporting. This can hamper the progress of research as, even if interesting results are published, the possibility to further investigate the issue through replication or reproduction can be limited.
6.2. **RQ2: Have rigor and relevance changed over time?**

To analyze the evolution of research, Figure 6 shows how the variables of rigor and relevance have changed over time together with the number of papers included each year. Rigor and relevance are scored on the left y-axis while the number of papers is given on the right y-axis. Rigor has a maximum value of three, and relevance can be at most four. The results indicate an improvement in both rigor and relevance. Rigor matures from a rating of around 0.5 during most of the 1990s to about one in the 21st century. This is a doubling over ten years, which is an improvement, even though the average rigor is still not high (on average, only one of the three aspects is described).
With regard to relevance, the increasing number of papers that present technology evaluations supports the results given in [38], which shows increasing levels of validation. However, the industrial relevance or realism of evaluations does not seem to show any noticeable improvement over time (even though the most recent years indicate a slight improvement). Thus, while the results show an increasing number of potentially relevant studies, the average relevance of research papers remains about the same.

6.3. **RQ3: Does publication venue influence rigor and relevance?**

To investigate the influence that publication venue has on articles, the average rigor and relevance of each journal are calculated. This is illustrated in Figure 7. The number of papers included in each journal is indicated on the right y-axis. What is apparent from the results is that the rigor and relevance of articles varies between different publication venues. The Empirical Software Engineering (EMSE) journal stands out in terms of both relevance and rigor. No other journal comes near the same levels of both relevance and rigor. The Software Quality journal (SQJ) has a maximal relevance score but a low score for rigor. On the other side of the spectrum, it can be seen that the flagship journal in software engineering, Transactions on Software Engineering (TSE), surprisingly has both low relevance and rigor. This may depend on TSE incorporating work that is more theoretical. However, pure theoretical work is not included in this review as the simplest form of evaluation is chosen to be a technology applied to an example. Furthermore, journals with page limitations, e.g. IEEE Software and Computer, show low levels of rigor. When the number of pages is limited, the first
thing to go seems to be describing aspects related to rigor. This may be good enough for these journals if the results are more thoroughly described elsewhere.

![Figure 7: Average rigor and relevance of each journal](image)

7. Validation of the model

The review presented in Sections 4 and 6 provides a case for applicability of the model to technology evaluations in requirements engineering. This section gives an account of how the model was constructed and validated and how the results of applying the model can be used. Section 7.1 describes how in the review the model evolved through usage. Section 7.2 discusses how the results of the systematic review, and thus the model, can be used.

7.1. Model evolution

The systematic review acted as a validation of the model, showing that the model could be applied, and also doubled as a case on which the model was refined. The model thus evolved based on lessons learned from the systematic review to improve its ability to characterize research. The interesting thing is that the evolution of the model actually consisted of reducing the model (making it more basic and forgiving). This was necessary, as the state of the technology evaluations (studies) were lacking in any advanced characterizations other than the simple ones seen in the systematic review in Section 6. The model should relate to state-of-research and only include aspects that several studies/papers mention to avoid ending up with many detailed aspects applicable to only a very few studies (if any). Thus, the model used in the systematic review is actually the evolved (simplified) version of the initially designed model. The simple model came out of the necessity to offer a more forgiving and less extensive and detailed evaluation of present research. This is confirmed if we look at the results of the systematic review carried out with the simplified model. Looking at the
classifications uncovered in the review (see Figure 5), about 10% of the papers have the highest level of rigor and 15% have the highest level of relevance. This means that the quite basic model relates rather well to state-of-research. Having more detailed aspects would only offer even fewer (if any) papers classified at higher levels. The original model (prior to adapting it to reality and simplifying it) included additional aspects. These can be seen in Table 10.

Table 10: Aspect excluded from the model

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source of problem statement</td>
<td>What is the source of the problem addressed by the technology? If the technology is developed to address needs in industry it is more likely to be relevant for industry.</td>
<td>[52]</td>
</tr>
<tr>
<td>Expertise and skill of subjects</td>
<td>What is the skill and expertise of subjects used in the evaluation in relation to the technology being evaluated?</td>
<td>[25]</td>
</tr>
<tr>
<td>Task duration</td>
<td>What is the duration of the evaluation?</td>
<td>[25]</td>
</tr>
<tr>
<td>Dependant variables considered (efficiency, effectiveness, quality)</td>
<td>Are dynamic properties of technology usage measured or is it is only a case of application?</td>
<td></td>
</tr>
<tr>
<td>Rigor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conclusion validity/Reliability</td>
<td>Is the conclusion validity or reliability of the study discussed?</td>
<td>[15]</td>
</tr>
<tr>
<td>External validity</td>
<td>Is the external validity discussed?</td>
<td>[15]</td>
</tr>
<tr>
<td>Internal validity</td>
<td>Is the internal validity discussed?</td>
<td>[15]</td>
</tr>
<tr>
<td>Construct validity</td>
<td>Is the construct validity discussed?</td>
<td>[15]</td>
</tr>
</tbody>
</table>

Source of problem statement was considered for inclusion in the classification. If the problem tackled in an evaluation originates from industry, the results are presumably more relevant than if an academic problem is investigated. This aspect was removed, however, since very few papers mentioned having an empirical basis or industry need that motivated the research that was carried out. In addition, not reporting industrial motivation might be a style of writing, i.e. research is motivated in relation to previous research.

Describing the expertise and skill of subjects in relation to the technology under evaluation is important in terms of indicating the population for which the results holds true [14]. Alas, the expertise of subjects used in evaluations is seldom mentioned in papers. Instead, a simplified classification of subjects is included in the classification, identifying only whether the subjects are researchers, students or professionals. Different aspects can be considered in classifying the scale of evaluations, including task duration and the scale of the application. The duration of tasks performed in evaluations is seldom mentioned in papers. Instead, only the scale of the application is included in the model, as this is often reported.

The way in which the study is carried out in terms of what dependant variables are measured is not considered in the final model. The variables measured should be relevant to practitioners for the results to be relevant for industry [53]. This aspect was excluded from the classification as it is difficult to incorporate the aspect into an overall scoring of rigor and relevance. It is also difficult to discern what constitutes a relevant dependant variable in requirements engineering [54], and the relevance of variables measured is likely also dependant on the technology investigated.
A decision was also made to exclude more fine-grained aspects concerning the validity of the evaluation, as validity is seldom reported in papers. Thus, it was decided to include only one aspect, comprising all parts, for classifying validity.

With respect to the scoring rubrics used to classify studies and the quantification of the classification into variables, it is important that the valuation actually approximates the relevance and rigor of studies. Rigor is scored to reflect the extent to which aspects related to rigor are reported. The scoring should enable locating studies described to an extent that facilitates replication, reproducing and synthesis of evidence. In this respect, the quantification is straightforward in assigning higher values to studies that described aspects to a greater extent and should thus provide a good approximation of what studies lend themselves to replication, reproduction and synthesis.

The scoring for relevance should reflect the potential for evidence to impact industry. Evidence obtained in realistic settings is more likely to facilitate generalization to the target environment (industry). Thus, when valuing context, subjects and scale, a higher score is assigned when the aspects are classified as being realistic. The research method used in studies also influences how compelling evidence is to practitioners. Zelkowitz [12] found that research methods relevant for practitioners’ environments, e.g. case studies, are valued higher than, for example, experiments. This can in part depend on these research methods lending themselves to investigations of real world situations. However, it can also depend on the state-of-research, i.e. the realism of experiments is usually not high [25]. Table 11 shows the distribution of context, scale and subjects for different research methods as found in the systematic review presented in Sections 4 and 6. Research methods that are scored as contributing to relevance in the model presented in Section 3 often have context, subjects and scale that are realistic. The majority of studies reported as lessons learned, field studies, surveys and interviews have realistic scale, subjects and context. Action research studies are also often conducted in realistic settings but use researchers as subjects. This can be expected, however, as, in action research studies, researchers sometimes act as experts/investigators that drive change [55]. However, case studies show a mix of realistic and unrealistic traits. It is often emphasized in the literature that case studies per definition are conducted in real world settings [33, 56]. This is not reflected in the results of the review and is most likely due to a large misuse of the term “case study” in the papers reviewed. This ambiguous use of the term “case study” introduces a problem in evaluating research methods since it is likely that the type of case study that is valued by practitioners [12] is not the type of case study often presented in requirements engineering. This makes interpreting the resulting relevance score for studies more uncertain. The problem could have been solved when classifying the studies. Instead of giving researchers the benefit of the doubt when the research method was classified, i.e. placing trusting in the reporting in the papers, the research method could be mapped according to investigator understanding. However, this risk of introducing subjective bias in classifying research method as aspects related to rigor is seldom reported. Ideally, researchers and reviewers should try to minimize the use of incorrect terminology in published work.

Looking at experiments using both software and human subjects, these are seldom carried out in a realistic setting. This may be one reason for practitioners to value case studies and lessons over experiments. Considering the state-of-research presented in
Table 11, the valuation of research methods used in the model presented in Section 3 seems to approximate the actual relevance. Given the distribution, studies ending up with a relevance score of three or four should be interpreted as relevant, as this includes action research studies performed with researchers as subjects and experiments carried out in a realistic setting.

Table 11: Context, subjects and scale used with different research methods

<table>
<thead>
<tr>
<th>Research method</th>
<th>Context</th>
<th>Subjects</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academia</td>
<td>1 8 3</td>
<td>1 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Industry</td>
<td>10 14</td>
<td>28 4 4</td>
<td>3 3 0</td>
</tr>
<tr>
<td>Practitioner</td>
<td>20 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Researcher</td>
<td>30 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Student</td>
<td>108 14</td>
<td>10 1 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Not mentioned</td>
<td>4 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Toy</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Downscaled</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Experiment (HS)</td>
<td>36 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Example application</td>
<td>8 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Lessons learned</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Case study</td>
<td>4 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Field study</td>
<td>0 4 4</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Experiment (SW)</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Survey</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Interview</td>
<td>0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>
7.2. Using the model

Researchers can use the 10% of studies with the highest level of rigor (see Figure 5) to synthesize evidence in the field. This is possible because the studies are described in such a degree that the evidence can be scrutinized and judged. These studies are also prime candidates for replication or reproduction. Researchers looking for motivation on the part of industry for studies or practitioners searching for evidence of usability and usefulness of technologies in an industrial setting can use the 15% of studies rated as having the highest industrial relevance (see Figure 5).

However, the results given in Section 6 show that the majority of evaluations lack both rigor and relevance. To increase the number of studies that can be used for the above mentioned objectives, the results presented in Section 6 also emphasize ways to improve the state-of-research in terms of rigor and relevance. Figure 8 illustrates three different alternatives to increasing rigor and relevance of evaluations in Figure 5. In planning evaluations, researchers can use these alternatives to ensure an adequate rigor and relevance of studies. Each ways are discussed below.

![Figure 8: Improving state-of-research](image)

**A: Increasing rigor of articles with low rigor and industrial relevance:**

It has been argued that rigor is irrelevant until relevance has been established [57], suggesting that the most important way to improve evaluations is to focus on Arrow B in Figure 8. This is not the exact standpoint taken in this paper. There is still a necessity to experiment and investigate the underlying assumptions of technologies. The main thing is to keep evaluations relevant. This means that two issues must be considered: the topic investigated and how evaluations are performed. Distinguishing between relevant and non-relevant evaluations is not easy, as there is no way of knowing this up front. There is also no guarantee that academic relevance overlaps industrial relevance. Earlier literature reviews have found that topics studied empirically are quite narrow [40]. To improve the situation, studies with less industrial relevance should offer academic value in terms of rigor by shifting the focus from application examples performed by the researcher him/herself to actual evaluations using experiments, for example. Experiments have a higher potential to enable an understanding of the technology under evaluation and facilitate synthesizing evidence. Further, conducting more experiments would potentially overcome the problem of a narrow focus of empirical research. If more experiments are performed, it is likely that the scope of what is investigated will broaden.
B: Increasing the relevance of articles:
The most significant way to aid technology transfer seems to be to promote research relevant to industry. This not only improves the technology transfer decision support offered to practitioners but also enables feedback from industry to academia, i.e. evaluations in industry can provide feedback to academia on what works in practice. This constitutes one way of improving the relevance of technologies developed in academia.

The first thing to do to increase the relevance of technology evaluations is to perform them in settings similar to ones in which they are intended to be used. This implies having a realistic scale, subjects and context in evaluations. In addition to the setting, the way in which evaluations are performed influences their realism. Using off-line experiments, it is difficult to emulate real usage with respect to scale and the duration of tasks. This calls for dynamic evaluations where technologies are used in real software development in which they are monitored and evaluated. However, a prerequisite for performing dynamic evaluations in industry is that the basics of the technology are known. Dynamic evaluations are meant to address scaling issues and sort out teething problems, e.g. tailoring technology for practical use. Thus, failing to emphasize an understanding of the basics of a technology beforehand could render dynamic evaluations useless. Researchers need to pursue an iteratively improved understanding of the technologies under evaluation (illustrated in Figure 9). Going directly for dynamic evaluations in industry is risky as there might be numerous issues that are not understood that would jeopardize the effort. Static evaluations, e.g. experiments, in either academia or industry, are needed to understand the basics of the technology, e.g. the efficiency and effectiveness of a technology. Pilot evaluations in industry are used to tailor and collect initial feedback on practical issues in the particular technology using limited resources. Then, when the technology is understood and tailored for practical use, dynamic evaluations utilizing case studies, action research etc. can be pursued. This also enables continuous feedback to technology development in academia, as industry feedback adds valuable information beyond pure academic feedback.

In addition, getting companies to commit to evaluations alone indicates the relevance of technologies. It is unlikely that commitment will be gotten to an idea that is not perceived as important. If research is pursued in this way, bad ideas are likely to be
discarded more quickly [2]. This also limits the risk of discarding good ideas that are simply not yet mature as they are iteratively tested and refined.

**C: Increasing rigor of relevant articles**

In the case of studies that have a high potential for impacting industry, e.g. studies that show improvements in industry, failing to report adequately lessens their value. From an academic perspective, failing to report study design and validity makes it difficult to evaluate the quality of the evaluation. Empirical work is not perfect in contrast to theoretical work, and failing to report aspects of the study makes evaluating them a question of guesswork. It also limits the possibilities to replicate or reproduce the study and thus strengthen the evidence. From an industrial standpoint, failing to report the context of the evaluation makes it hard for practitioners to understand whether the results are valid for them.

**8. Future work**

This review considers only evaluative research in requirements engineering with a focus on work published in the included journals. In order to compare and contrast different sub-fields in e.g. software engineering, we encourage other researchers to conduct similar reviews with a broader or different scope. The classification scheme used in this review might need to be changed to reflect specific fields, e.g. there may be a need to include data source (benchmarking suite) to evaluate the testing field.

While the model can and should be adapted to fit specific subject areas, it should also evolve towards a more precise and demanding model as research design, execution and reporting improve.

**9. Conclusions**

This paper presents a model for evaluating the rigor and industrial relevance of technology evaluations, and introduces the use of candidate rubrics for this purpose. The model can be used by practitioners to gauge the relevance and validity of available evidence, and by researchers to point out evidence that can be synthesized, or give information that would enable replication or reproduction. Researchers get valid results they can build on, and industry professionals get decision support in terms of giving motivation to try out research results.

The model is validated through a systematic literature review that investigates the rigor and industrial relevance of technology evaluations in requirements engineering. The aim is to show the applicability of the model and to characterize how evaluations are carried out and reported to evaluate the state-of-research. The major findings of the review are:

- The number of evaluations presented in requirements engineering is increasing.
- The majority of technology evaluations in requirements engineering lacks both industrial relevance and rigor. This means that most evaluations are performed in unrealistic environments and that the reporting lacks descriptions of study design, context and validity.
- A very small fraction of evaluations (six out of 349) exhibits the highest level of industrial relevance and rigor.
- The research field does not show any improvement in terms of average industrial relevance over time. Even though some excellent evaluations are carried out, the
field as such seems to maintain a *status quo* in terms of relevance. While the average rigor of evaluations has doubled in the last ten years, it is still at a very low level. However, as evaluations are becoming more commonplace, more relevant and rigorous evaluations are being performed.

- There is a significant difference between publication venues when it comes to industrial relevance and rigor. This might indicate that reviewing policies and/or differences in quality demands impact research reporting.

The results presented pertain only to technology evaluations in requirements engineering and especially the journals included in this review. However, as the journals are the premier publication venues for researchers in requirements engineering, they are likely to contain the most mature research in the field.

Regarding the relevance of research carried out, different indexes related to number of publications and citations are currently often used to assess the relevance or impact of research. The results of this review stress the need for a more balanced view of relevance in software engineering research. Number of publications and citations does not consider the potential for research to impact industry, which is the most important factor determining success in an applied research field such as software engineering [7]. In this respect, the model presented in this paper can provide a more balanced view of relevance of research. The model can also be used preemptively in order to improve state-of-research. The model emphasizes aspects that are important in planning and reporting research and can thus be used by researchers when they plan and report studies.

**References**


Chapter V

Practice Selection Framework

Submitted to

International Journal of Software Engineering and Knowledge Engineering
Practice Selection Framework

M. Ivarsson¹, T. Gorschek²

¹Department of Computer Science and Engineering
Chalmers University of Technology
martin.ivarsson@chalmers.se

²School of Computing
Blekinge Institute of Technology
tony.gorschek@bth.se

Abstract

Knowledge management (KM) in software engineering and software process improvement (SPI) are challenging. Most existing KM and SPI frameworks are too expensive to deploy or do not take an organization’s specific needs or knowledge into consideration. There is thus a need for scalable improvement approaches that leverage knowledge already residing in the organizations.

This paper presents the Practice Selection Framework (PSF), an Experience Factory approach, enabling lightweight experience capture and use by utilizing postmortem reviews. Experiences gathered concern performance and applicability of practices used in the organization, gained from concluded projects. Project managers use these as decision support for selecting practices to use in future projects, enabling explicit knowledge transfer across projects and the development organization as a whole. Process managers use the experiences to determine if there is potential for improvement of practices used in the organization. This framework was developed and subsequently validated in industry to get feedback on usability and usefulness from practitioners. The validation consisted of tailoring and testing the framework using real data from the organization and comparing it to current practices used in the organization to ensure that the approach meets industry needs. The results from the validation are encouraging and the participants’ assessment of PSF and particularly the tailoring developed was positive.

Keywords: Postmortem review; Knowledge management; Software process improvement; Software engineering

1. Introduction

To maintain and increase competitive advantages, software organizations must continually strive to maximize the utilization of the knowledge and experience internal to the organization, refining and improving their development practices. Process assessment and improvement are often the means by which development organizations assure and improve their processes and tools to support their specific needs. For improvement, several well known software process improvement (SPI) frameworks exist, both prescriptive (see e.g. CMMI [1] and SPICE [2]) and inductive [3-5]. Prescriptive, or model based, frameworks take an approach based on a set of best practices that have proven successful in other organizations. The improvements are
implemented through the benchmarking of the organization to the predefined one-size-fits-most best practices. Inductive methods take their starting point in a thorough understanding of the current situation, basing improvement efforts on the issues most critical to the specific organization.

Independent of the framework chosen, there is a risk that several things will be omitted when an organization turns directly to process improvement as a way to assure and improve its processes. First, process improvement often happens as a reaction to a problem, and an SPI effort can take up to 24 months or more to realize [6]. Second, cost in terms of initiation threshold (training, tools, doing it the first time and so on) and potential risks associated with unproven practices can also be important factors [7-9]. Third, management and practitioner commitment is crucial, and change might be resisted if not very well motivated and supported by both practitioners and managers [10-15]. Fourth, but not least, looking directly towards the new, e.g. introducing new practices such as use-cases, modeling practices and so on, can result in knowledge and experience already owned by the organization being lost or overlooked [16].

The Experience Factory (EF) [17] approach have been central in leveraging knowledge internal to software organizations and to learn from past success and failures in order to improve development processes. EF is an infrastructure for sharing and reusing experiences gathered in projects carried out in an organization. Traditionally, objective metrics have been used to evaluate projects and then to package and disseminate these experiences in the organization [15]. In comparison to subjective metrics, objective metrics have the advantage of not being influenced by judgment of an expert. Still, commitment and intimation threshold remains a problem as EF is a major undertaking for an organization [15, 18].

An alternative is to use postmortem reviews to gather subjective metrics in EF to lower cost and circumvent the need for only relying on objective metrics. Postmortems can have a learning effect on an individual level, team-level and organizational level. However, few studies have focused on the organizational level [19].

Postmortems introduces a new problem as even if experiences are captured and stored, they are not always reusable or even usable by their intended audience [20]. This becomes evident when looking at state-of-practice as reports of software engineers not learning from past mistakes [21, 22] and experiences owned by the organization being overlooked [22]. In addition, the effort for analyzing postmortem data to enable learning on the organizational level can be high [23].

This paper proposes an approach towards quality assurance and improvement of processes, utilizing the traditional SPI mindset, but building upon it and changing the focus to a bottom-up approach that starts with both organizational needs and organizational knowledge. The approach is presented as a framework, namely the Practice Selection Framework (PSF). PSF is an Experience Factory approach that utilizes postmortem reviews for capturing experience in order to lower the initiation threshold for improvement and learning.

PSF is a framework for sharing experiences regarding practices used in an organization and was developed in close cooperation and collaboration with a large multinational company developing software intensive products, namely Volvo Car Corporation. In
PSF, experiences are stored and structured in the practice repository that contains the practices of an organization, and they are tagged with meta-data, primarily collected from the organization itself. The meta-data include information such as applicability, experiences, tips, and pros and cons associated with the practice in question. One of the overall goals is to leverage the inherent knowledge in the repository to equip projects with practices and support project managers when starting projects.

A crucial part of PSF is enabling the feedback loop. Practitioners perform a sort of mandatory postmortem, adding meta-data (or even practices) to the repository after every project. The idea is that the best and worst experiences regarding practice usage are collected in one place, improving the practice selection activity continuously as it offers iteratively improved decision support material to practitioners searching the repository for practices appropriate for their specific situation. This enables sharing practices that have been shown to lead to success in the entire organization and providing continuous evaluation of practice performance. Keeping track of the performance of practices used provides a basis for (and can be seen as) continuous process improvement.

Major benefits of this way of working are that good experiences are shared across projects and organizational specific meta-data are collected with regard to each individual practice instead of only having general one-size-fits-most information prescribed by e.g. an SPI framework. This is not to say that traditional SPI is not relevant; rather it is an integral part of PSF. As practice experiences from projects are reported back, the usefulness and fit of practices can be judged. If the judgment in any instance is that the present practices are inadequate, a traditional SPI effort can and should be initiated but should be aimed largely at PSF, i.e. improving/adding to the collection of practices and the meta-data in the repository.

This paper presents the PSF framework and its constituents as well as a static validation [24] performed in industry. The validation take the form of a workshop aimed at tailoring and validating the concepts used in PSF, as well as collecting feedback on the framework itself. The results show that PSF is a promising approach for sharing and improving practices used in an organization.

The paper is structured as follows. Section 2 provides an overview of knowledge management for process improvement in software engineering and discusses their characteristics and possible limitations. In Section 3, an overview of PSF is given and illustrated with generic examples. Section 4 presents the design of the static validation, including research questions to be answered, the context in which it was performed, and the actual design of the validation. The results from the validation is presented in Section 5 and discussed in Section 6. Finally, conclusions are presented in Section 7.

2. Background and related work

There are many methods and frameworks that take organizational learning into account. Knowledge Management in general, but also in relation to software engineering in particular, has discussed the importance of managing and utilizing knowledge and of spreading tacit and explicit knowledge in an organization [25, 26]. The recognition that practitioners themselves are the main company assets makes this even more evident. The challenges identified point towards the difficulty in valuing
tacit knowledge and deciding what knowledge should be made explicit. The cost of making knowledge explicit and the use of training as one major vessel for knowledge transfer are high-cost, and accuracy can be hard to gauge: i.e. what knowledge should be made explicit, packaged and transferred? The actual transference is also a challenge. For a project manager initiating a project, the choice of what practices to use is many times based on a “gut feeling” derived from experience and tacit knowledge [26]. Using e.g. predictive or process models that use large amounts of project data as input or using simulation can support practitioners [27], but transparency and, ultimately, trust can be an issue. In addition, the cost of keeping high-quality/high-accuracy data for every project up-to-date, making it suitable for input to the prediction simulation, again becomes a problem, as high initiation and maintenance costs increase the initiation threshold of any process improvement activity [7-9].

The concept of performing postmortems upon development project conclusion is closely related to the concept of supporting organizational learning, making experiences explicit, and can be seen as vital for process improvement activities [28-30]. Performing postmortems is both practical and low cost, offering clear benefits, while it also has a low initiation threshold, i.e. suitable for organizations of all sizes. The main problem is that organizations seldom perform postmortems, even if they are in the official process charter, as pressures to start the next project overshadow the good intentions of learning. Verner and Evanco report that, of 42 projects studied, only 33% had postmortem reviews [29], even though performing postmortems was associated with the production of high quality development artifacts and the ability to manage risk more efficiently throughout the development process [29]. As a result, the same mistakes propagated across projects over time [29].

Another problem is that, even if postmortems are performed, information is seldom shared across project boundaries [21, 31, 32], resulting in situation that the only vessel for knowledge transfer is the individual practitioner moving between projects. Methods for analyzing postmortem data for use on an organizational level exist but carry high costs [23]. The project managers, who are the driving force behind selecting the practices to be used in a specific project, might or might not be involved in the learning experience of another project. This problem is compounded by the fact that few organizations have procedures or structures in place to enable sharing of information [28, 29, 31]. This can be devastating, as good practices are highly dependent on project characteristics, i.e. what works for one project might not be suitable for another [33-37]. Relying on only word-of-mouth rumors, good practices conveyed out of context with little or no additional information can create more problems than they solve. Looking at SPI frameworks, such as CMMI, most of the data gathered during process assessment is project based, i.e. the realization phase is often the subject of study. One potential problem is that most SPI frameworks adopt a one-size-fits-all view, not only across companies but also across projects.

Experience Factory (EF) [15, 38] is very closely related to the same principles, i.e. learning and knowledge transfer. EF is an infrastructure for reusing life cycle experiences and products for software development. Experiences are collected from development projects and are packaged and stored in an experience base. Packaging entail generalizing, formalizing and tailoring the information collected to be easy to reuse. The idea is that software development projects can improve their performance
through the utilization of experiences from previous projects. The classical usage of EF utilizes the Quality Improvement Paradigm (QIP) [17] for software process improvement supported by the Goal Question Metric (GQM) [17] for establishing project and corporate measurements. The instantiation of EF requires the creation of a new experience factory organization to be responsible for the analysis and packaging of the experience collected. Packaging is a difficult task in itself [18], and the cost of the EF organization is about 10% of the total organizational project budget [15]. This is not to say that EF has negative return on investment but rather that the commitment and thus the initiation threshold for such an undertaking is considerable, especially for small and medium sized organizations (SMEs). The most prominent use of EF as described above is the NASA SEL [15] (for a review of research on EF see [39]).

PSF was strongly inspired by the idea of postmortems, while aiming to remedy some of the presented problems by combining it with an EF like approach for supporting organizational learning. Building on state-of-practice, the realization that gut feeling was dominant in practice selection, PSF is intended to allow practitioners to share information and experiences associated with a given practice. In essence, by using the PSF repository, it is possible to “browse” through other colleagues’ gut feelings and experiences, while maintaining the context of a specific project. The context of the experiences gathered in the repository is maintained as project characteristics and saved in relation to the practices and experiences. Traceability to originator is also saved as it enables face-to-face communication transferring tacit knowledge, which is crucial for keeping PSF light-weight, making it accessible to SMEs as well as larger organizations [22, 40]. As the focus of PSF is the practices, everything associated is seen as meta-data, and the main work is connected to the gathering of this meta-data from practitioners as projects are concluded.

PSF builds on the concepts of EF and related research. The main differences from the classic EF are highlighted below:

- The classic usage of EF builds on that measurements are in place to evaluate the technologies used while PSF focuses on qualitative feedback through postmortems. Using qualitative measurements enables more organizations to utilize EF as few have large measurement programs, as the instantiation can be very costly [15]. Still, using postmortems integrates feedback into the engineering work and thus assures that experiences are collected.

- In EF, experiences, processes and technologies are packaged, i.e., generalized, formalized and tailored to be easy to reuse. Packaging is a difficult task [18] and in PSF this is circumvented by using already existing packages, practices. PSF builds on the current situation, the practices used in the organization, and uses experiences to gradually refine these and transfer successful ones to other projects. In contract to the classic use of EF, PSF stores experiences that are already analyzed in a postmortem analysis meeting. This means that there are no explicit packaging and generalization in PSF, but rather that experiences are pre-packaged as they concern some already existing practice, which it is connected to.

- EF has a separate organization for processing the information collected from the development organization. This organization is concerned with establishing measurement goals, analyzing and packing the information collected. Since there are no or few quantitative measurements and no explicit packaging in PSF, this burden is limited and there is not necessarily a need for a separate
organization. Instead, PSF can be seen as a tool for process and project managers to use for choosing among and improving the practices used in the organization. This means that the use of PSF is distributed over already existing roles.

- As there is no packaging of experiences and the information in PSF are in most cases subjective (not as accurate as quantitative measurements) the focus shifts towards enabling sharing tacit knowledge to overcome the ambiguities, e.g., enabling locating persons to get more information from.

In comparison to postmortem reports, PSF adds structure to experiences captured. This is achieved by focusing the type of experiences captured, i.e., experiences related to practice performance. In addition, to facilitate locating relevant information captured, practices are arranged according to the process model used [18, 41].

3. Practice Selection Framework – an Overview

This section gives an overview of PSF and a detailed account on how to use PSF to share practices, coupled with experiences within an organization. An overview of PSF can be found in Figure 1. PSF is an approach for leveraging the knowledge inherent to an organization regarding practices used. People in projects often already know what practices work and what does not [5, 42, 43].

In PSF, practices are the primary driver for transferring experiences in an organization. This is motivated by the fact that practices are often pre-packaged in the form of process or method descriptions, often including templates and guidelines, which are already available. This natural state should decrease the overhead of documenting experiences. In addition, practices are a primary candidate for conveying knowledge, as most process improvement efforts revolve around introducing new practices to tackle problems divulged during process assessment. Thus, practitioners are used to practices being the main artifact of a change.

Practices together with meta-data collected from the organization are stored in a practice repository (labeled A in Figure 1), which is shared across the organization. Meta-data include attributes describing practices (labeled A1 in Figure 1), experiences from using the practice (labeled A2 in Figure 1) and the characteristics of projects in which the practice have been used (labeled A3 in Figure 1). The practice repository and its constituents are described in Section 3.1 and the structure used to organize practices is discussed in Section 3.1.4.

Practices as well as meta-data are collected through postmortem reviews when projects conclude (labeled B in Figure 1) which is detailed in Section 3.2. The meta-data include information such as applicability, experiences, tips, and pros and cons associated with the practice in question. One of the overall goals is to leverage the inherent experience stored in the repository to equip projects with practices and support project managers when starting projects (labeled C in Figure 1). Experiences collected regarding practice performance aid other parts of the organization i.e. other projects to make informed decisions regarding adopting practices already used in the organization (see Section 3.3). This enables projects learning from each other in addition to continuous process assessment as experiences can indicate if there are practices that need to be improved. How to use PSF to assess practices used in the
organization and support software process improvement (labeled D in Figure 1) is described in Section 3.4.

Finally, different organizations looking to adopt PSF will most likely have different needs and constraints. Acknowledging that there is not a one-size that fits all organizations Section 3.5 discusses how PSF can be tailored and implemented.

![Figure 1: Overview of the Practice Selection Framework.](image)

### 3.1. A - Practice repository

Practices and meta-data collected from the organization are stored in the practice repository (Experience base in EF). As the focus of PSF is the practices, everything associated is seen as meta-data. Meta-data take the form of characteristics of projects where practices have been used, experiences from using practices as well as attributes of the practice itself e.g. pre-requisites for usage (see Figure 2). The characteristics of projects set the context for the experiences. This means that project managers that use the information in the repository can evaluate how relevant the experience is for his/hers project [17]. It is up to the organization adopting PSF to decide how the entities are tailored with respect to what information should be collected for each.

![Figure 2: Entities in the practice repository.](image)

The next sub-sections give an account of information that can be used for each of the entities in the practice repository. In addition, to facilitate easier finding practices used in the organization, the repository needs to be structured. How the repository can be structured is discussed in Section 3.1.4.
3.1.1. A1 - Practice

Keeping descriptions of practices is necessary to be able to facilitate their adoption in other projects. To achieve this, the definition of a practice must be specific to the organization, e.g. if use cases [44] are a practice used in the organization, it is the organization’s definition of a use case and its experience related to it that are important to share within the organization. The definition of the practice in the literature is irrelevant as it is not the organization’s own experience. Guiding practice documentation by stipulating what attributes are important ensures a minimum level of information completeness and uniformity in the repository, making it easier to use. A number of attributes suitable for describing practices are summarized in Table 1.

<table>
<thead>
<tr>
<th>Practice Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A descriptive title for the practice.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Purpose should be a concisely capture what the practice is used for. This attribute is used to screen practices quickly.</td>
</tr>
<tr>
<td>Description</td>
<td>The description should convey how the practice is used. In addition, other sources can be used to facilitate usage such as: Templates, Examples, Checklists</td>
</tr>
<tr>
<td>Process connection</td>
<td>Process connection traces the practice to the governing process. The process connection shows where in the development the practice is meant to be used.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Dependencies captures what other practice are needed to make this practice work.</td>
</tr>
<tr>
<td>Pre-requisites</td>
<td>Pre-requisites describes all other things that need to be in place to use the practice. This includes roles, tools etc.</td>
</tr>
</tbody>
</table>

Practices are often described in the form of process descriptions, templates used, examples, checklists and guidelines. In addition to describing the practice, certain attributes have been included to support practitioners in selecting practices from the repository. Purpose provides an initial screening of practices when selecting from the repository through abstraction of what the practice is used for. If the practice’s purpose does not fit the needs of the project, the search can quickly move on to the next practice while, if the purpose fits, a more thorough evaluation of the practice can be made. Practice dependencies capture what is needed in terms of other practices to take advantage of a practice at hand, e.g. capitalizing on the use of an object oriented programming language is dependent on having an object oriented design in place. Process connection indicates in what part of the development process the practice is meant to be used (see Section 3.1.4). Finally, prerequisites are used to describe what in addition to other practices is needed. This includes roles, life cycle models etc. that have to be in place to make use of a practice. An example of a prerequisite is the need of having the role tester present if test-case driven inspections are to be used [45].

3.1.2. A2 - Experience

Postmortem analysis is used to assess practices when projects are concluded (see Section 3.2). The experiences collected from the analysis are documented in PSF to
enable other projects to learn from projects’ failures and successes and thus pick up practices that have been shown to be successful elsewhere in the organization. To control what information is collected, attributes in PSF are tailored to fit the information prerequisites of the organization. In addition, rules control the extent to which experiences are collected. The rules stipulate what attributes is mandatory for projects to supply and the required documentation. Ideally, experiences are rooted in quantitative measures, e.g. effectiveness and efficiency of the practice in the project. However, as measurement programs are expensive to set up and sustain, the attributes presented here focuses on a few qualitative measures presented in Table 2.

PSF supports both codification and personalization strategy of knowledge management. Codification is achieved by describing practices, projects and experiences and storing them in the practice repository. In addition, PSF enable personalization by traceability to the person that has experience from using the practice as an experience (named “Responsible” in Table 2). This is important as the information related to practices often is incomplete or does not reflect their actual use. This is in part because extensive documentation is considered too resource demanding or that the knowledge is hard to make explicit. To manage this, organizations can opt for a lightweight level of documentation and instead use personalization of the knowledge. This can be made possible by providing traceability to experts owning tacit knowledge, e.g. “Charlie knows use-cases, talk to him”. This makes it possible for practices and experiences to be personalized, and PSF can be used as a catalog with links to experts. This aids in transferring tacit knowledge, which compensates for shortcomings in documentation. In addition, this can be used to keep track of experts or employees with particular experiences in the organization, which can be used when staffing projects or identifying resources to be used in tutoring and training new employees or when a project adopts new practices.

Table 2: Experience attributes.

<table>
<thead>
<tr>
<th>Experience</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>Traceability to project</td>
<td>Provides a link to the project in which the experience has been captured.</td>
</tr>
<tr>
<td>Traceability to practice</td>
<td>Provides a link to the practice that the experience relates to.</td>
</tr>
<tr>
<td>Responsible</td>
<td>Provides traceability to the one responsible for the practice in this particular project. This attribute can be used to keep track of persons with experience from using the practice within the organization.</td>
</tr>
<tr>
<td>Performance</td>
<td>Performance reflects how the practice have performed in this project. Performance can be actual measures of effectiveness or efficiency or a subjective measure of how well project participants think that the practice have fulfilled its purpose e.g. on a 100 point scale.</td>
</tr>
<tr>
<td>Cost/bang</td>
<td>Essentially bang-for-buck or return on investment for the practice. This reflects how the participants in the project perceive the benefits from the practice weights against the cost of using it.</td>
</tr>
<tr>
<td>Recommendations</td>
<td>Recommendations are a way to share additional experiences from using the practice. Recommendations can for example be concrete advice on how to improve the practice.</td>
</tr>
</tbody>
</table>
All the attributes in Table 2 focus on capturing subjective experiences of practice use and initiation threshold. Collecting experiences of the initiation threshold, such as risk and initiation cost, alleviate adopting practices in other projects. Attributes related to practice use include performance, who is responsible, cost/bang, pros and cons. Performance consists of an evaluation made by practitioners of how the practice has performed, which is used by others as an indication of whether it is worth adopting. Even though a practice is perceived as being advantageous, certain aspects of the practice might bring negative effects or warrant changes in the practice. The recommendations attribute are a way for practitioners to convey these aspects of practice usage experienced. Cost/bang is a way to capture the relative cost effectiveness of a practice. If for example a requirements specification practice is to be evaluated, the cost/bang captures how expensive each requirement was to specify on average using the practice in question.

3.1.3. A3 - Project

The context in which the experience is acquired is saved to enable project managers to gauge the relevance of the experience in relation to their own challenges and projects. For experiences to be usable in a particular project they need to be valid in that context, e.g. an experience derived in a small research project might not be applicable in a large production project. To evaluate the transferability of experiences, the characteristics of the project in which the experience was derived are stored in connection with the experiences. Project characteristics are also used to catalyze the diffusion of practices in the organization, as other projects can see what practices are used in other projects together with how those projects compare to the one at hand. A summary of possible project characteristics is given in Table 3.

Table 3: Project attributes.

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead-time</td>
<td>Captures the lead-time for the project.</td>
</tr>
<tr>
<td>Personnel</td>
<td>The number of people involved in the project.</td>
</tr>
<tr>
<td>Project size in man-hours</td>
<td>The resources in man-hours used in the project.</td>
</tr>
<tr>
<td>Requirements stability</td>
<td>The volatility of the requirements. Captures some of the conditions (level of change) under which the project was performed.</td>
</tr>
<tr>
<td>Development mode</td>
<td>The development mode of the project including for example customer contract, innovative, market driven etc.</td>
</tr>
<tr>
<td>Development type</td>
<td>The project type including for example new product development, maintenance, evolution of existing product etc.</td>
</tr>
</tbody>
</table>

Project characteristics should be tailored to the organization at hand to avoid keeping characteristics that are not considered important for the organization or ones which are redundant in nature, e.g. keeping development mode as a attribute when all projects executed in the organization are bespoken projects. This builds a frame of reference for project managers reasoning regarding the suitability of a practice for projects. Keeping experiences and context information in the form of project characteristics is important, as practitioners consider case studies, field studies and legacy data as most valuable when selecting practices [46], where it is vital to be able to evaluate whether the context of the experience at hand is applicable in a future project.
3.1.4. A4 - Support structure of repository

Structuring the repository by decomposing the practices into several levels of abstraction helps to keep an inventory of what practices are present in different parts of the process. In addition, this eases the effort required to search for and increases the probability of successful identification of practices when selecting from the repository [18, 41]. Practices are decomposed to structure the information into several process abstraction levels instead of flattening the information by only providing one level in the structure of the repository. In Figure 3, a decomposition based on CMMI [1] comprising four levels of abstraction is shown. The top three levels structuring the process consist of Process area, Specific goals and Specific practice. These levels are used to structure the practices used in the organization, which are the bottom level of abstraction. The top level, the Process area, divides the process into clusters of related practices. CMMI comprises 22 process areas of which three are shown in the top part of Figure 3. A project manager searching for a practice related to requirements engineering would only need to consider the practices found under the respective abstraction. The next abstraction level further refine the division into Specific goals needed to satisfy the process area. Figure 3 shows the goals needed to satisfy the Requirements development process area, which include Develop customer requirements, Develop product requirements, and Analyze and validate requirements. The next abstraction level encompasses practices used to achieve the goals on the level above. Figure 3 shows an excerpt of practices used to fulfill the goals for the Requirements development process area. The last level holds the concrete practices used in the organization that constitute one way of performing these practices. Developing customer requirements can for instance be performed with a wide array of practices for specifying requirements (see Figure 3) e.g. Use cases and State-charts. How the decomposition is arranged in terms of how many levels and what these consist of is specific to the organization and needs to be tailored when instantiating PSF. Organizations that already have process descriptions or guides can use these to connect practices to, or connect experience attributes to practices already documented. Structuring the repository according to a process has shown to be beneficial in other knowledge management efforts. The tailoring of abstraction levels can be done in a way similar to that when working with requirements on different abstraction levels [47, 48].

<table>
<thead>
<tr>
<th>Process area</th>
<th>Requirements development</th>
<th>Project planning</th>
<th>Configuration management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific goals</td>
<td>SG1 Develop customer requirements</td>
<td>SG2 Develop product requirements</td>
<td>SG3 Analyze and validate requirements</td>
</tr>
<tr>
<td>Specific practice</td>
<td>SP1.1 Elicit needs</td>
<td>SP1.2 Develop the customer requirements</td>
<td>SP2.1 Establish Product and Product Component Requirements</td>
</tr>
<tr>
<td>Organization specific practice</td>
<td>Interview</td>
<td>Use cases</td>
<td>State charts</td>
</tr>
</tbody>
</table>

Figure 3: Example of practice decomposition.

3.2. B - Feedback

Postmortems reviews are a practical method for learning from concluded software projects [49]. Postmortems enable projects reflecting on lessons learnt, what worked and what can be improved in the future. In PSF, a postmortem is made when a project
has concluded to assess the performance and uncover what practices were used successfully and which constitute an improvement potential. The results are documented in the practice repository according to the rules and attributes established. This makes experiences regarding practice performance transcend project borders as the repository is shared within the organization. It is up to the organization how to organize the postmortem regarding only supplying experiences for the most successful and unsuccessful practices to all used in the project. Only providing experiences of practices featuring some particular characteristic offers a less resource intensive option to documenting experiences related to all practices in the project. Using postmortems for feedback of experiences integrates knowledge management into the engineering work and enables management to assure that a minimum level of experiences is collected. This is important, as it often is hard to get workers to share their experience [50]. An example of feeding back experiences is given in Figure 4 where two practices, use cases and a cost estimation practice has been used in a maintenance project. In the example it can be seen that use cases have been perceived as working well while the cost estimation practice could need improvement.

3.3. C - Selection of practices

Selection involves utilizing the information stored in the practice repository to equip a project with practices. This is usually carried out at project start-up but can also be used when a project needs new practices, e.g. mid-project. In an organization where projects and processes are diverse, the project manager that has knowledge about the project in question preferably performs this task. To support project managers in selecting practices, meta-data including experience and the context in which they were derived are used as decision support. PSF offers different types of decision support material for project managers looking to adopt practices. First, the practice itself together with its meta-data e.g. purpose, pre-requisites etc. offer an initial description of the practice itself. Second, meta-data concerning projects can show if and in what projects the practice have been used before and the characteristics of these. Last, experience collected from these projects indicate how the practice have performed. To get more information, experiences offer traceability to persons that have experience from using the practice. Example 1 illustrates how practice selection can be performed.
Example 1: An example of practice selection.

Previous projects that the project manager has participated in have all received a complete requirements specification from the client. Now, a new project includes finding out and specifying the requirements for the client. The project manager thus need to, among other things, adopt a new practice for writing down the requirements for the system.

<table>
<thead>
<tr>
<th>Process area</th>
<th>Requirements development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific goals</td>
<td>SG1 Develop customer requirements</td>
</tr>
<tr>
<td>Specific practice</td>
<td>SP1.2 Develop the customer requirements</td>
</tr>
<tr>
<td>Organization specific practice</td>
<td>Use cases State charts</td>
</tr>
</tbody>
</table>

Figure 5: Except of the practice repository.

Browsing through the repository he finds the *Develop customer requirements* practice which essentially entail writing down requirements, something that the organization in this example has experience from using Use cases and State charts. The project manager thus has the choice of trying one or both of these practices or try something not previously used in the organization. In order to make an informed decision he examines the experiences collected from previous projects shown in Figure 6. Here he can see that Use cases have received mixed opinions from projects while State charts have mostly received favorable opinions. To make the decision he compares the projects that have previously used the practices to the one at hand and sees that projects X, O and U are comparable to the project at hand i.e. similar project characteristics and similar products. As State charts seems to have perform better than Use cases for similar projects he decides that State charts are a candidate to use in the new project and proceeds by talking to persons that have hands-on experience with using the practice, which is also stored in the repository. He might also use this information to try to staff the project with staff that have experience from specifying requirements using the practices chosen.

<table>
<thead>
<tr>
<th>Experience</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project X</td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>Stina Svensson</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Could be improved</td>
<td></td>
</tr>
</tbody>
</table>

Use cases

<table>
<thead>
<tr>
<th>Experience</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project Y</td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>Ulf Nilsson</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td></td>
</tr>
</tbody>
</table>

State charts

<table>
<thead>
<tr>
<th>Experience</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project O</td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>Sten Stensson</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Neutral</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project Z</td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>Kalle Karlsson</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Project U</td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>Kalle Karlsson</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Experiences stored in the practice repository.
3.4. D - Identification of improvement potential

In addition to alleviating sharing practices within the organization, PSF also support process improvement as the repository provides a snapshot of how the organizations practices performs. This can be used to assess the need for improvement in the organization. The issues in Table 4 summarize how PSF is used to unveil a need for improvement. An improvement effort is initiated as a response to a careful consideration of current practices and associated experiences. The initiation of an external SPI activity can be mapped to three activities in PSF. Selection of practices from the repository not only supports project managers choosing working practices for a project but also doubles as an inventory of practices and experiences owned by the organization. A need for improvement is detected either as a lack of practices suiting the need of the project or as suitable practices having received mostly negative feedback. Should an adequate practice (or solution to a problem) be missing from the repository, an external process improvement activity should be initiated. The selection of an SPI framework to use for improvement is not relevant; rather, it is up to the organization to decide and a task to be solved. The important thing is that the SPI effort be focused on the practice repository. This implies that the process assessment originates with the practices in the repository, and any change or addition to the process (for example adding a new practice) is mapped and included into the repository, which in effect causes the improvement effort to transcend project boundaries making the results accessible to the whole organization.

Table 4: Process issues with relation to PSF.

<table>
<thead>
<tr>
<th>Relation to PSF</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing practice</td>
<td>There are no practices present in the repository matching the needs of the project.</td>
<td>Use SPI efforts to identify and include new practices in the repository.</td>
</tr>
<tr>
<td>Practice selection failure</td>
<td>The selection of practices performed has resulted in choosing the “wrong” practice for the project at hand. Projects abandon the practice.</td>
<td>Even though the practice has not been used in the project, the negative experience should be fed back into the practice repository to warn others making the same selection.</td>
</tr>
<tr>
<td>Practice deficient</td>
<td>A practice has received negative feedback indicating that there is a potential for improvement in selecting another practice for the task.</td>
<td>Inventory the repository for other suitable practices. If none is found, initiate an SPI effort to rectify the deficiency.</td>
</tr>
<tr>
<td>External issues</td>
<td>Issues negatively influencing project performance that do not relate directly to PSF or practices contained therein.</td>
<td>Initiate an SPI effort to identify and rectify the problem. If the improvement involves new practices, include them in the repository.</td>
</tr>
</tbody>
</table>

However, relying only on the experience originating from inside the organization can mean in the long run that the organization becomes increasingly good at using obsolete practices [16]. To limit this risk, and for the repository to be up-to-date and optimized, regular SPI efforts should also be performed. These can be seen as maintenance overhauls and can be triggered by, for example, practices receiving mostly negative experiences or by a general conception that the process could be improved. In an optimal case this is based on quantitative data obtained from an extensive measurement program collecting metrics on for example defects (quality) and time and cost per production unit (efficiency). In reality very few organizations have extensive measurement programs as the cost of keeping and updating detailed records and
collecting metrics for analysis is very high [15, 51]. In practice, maintenance can be initiated when the need arises, but it could be prudent to consider it as ongoing quality assurance.

3.5. Tailoring and implementing PSF
PSF is a framework that needs tailoring to fit an organization’s needs, conditions and process. This means that the parts presented thus far need to be adapted to reflect needs in the organization. This includes choosing what attributes to use in the practice repository to describe practices, experiences and projects. In addition, how selection of practices, feedback of experience and assessment of practices used need to be established and responsibilities assigned to roles in the organization. An initial tailoring can be performed as described in the static validation presented Section 4.

When tailored, PSF is introduced in the organization using an appropriate process assessment methodology such as iFLAP [4, 5] or SCAMPI [52]. During an initial process assessment of the organization, practices already in use are documented and discussed. This is the initial documentation of the repository. In addition, it is important to collect meta-data in relation to the practices discussed, collecting experiences and, especially, noting inadequacies. Initially seeding the repository is important [18] as a repository that have no usable information is likely to be abandoned and provides no incentive to supply new information.

4. Static validation of PSF in Industry
Static validation of PSF was performed at Volvo Car Corporation to get initial feedback on how the framework addresses needs in industry. This section presents the design of the static validation. First, the research methodology is introduced in Section 4.1. The context in which the study is conducted and the participants are presented in sections 4.2 and 4.3 respectively. Section 4.4 describes the workshop used to tailor and introduce the participants to PSF while Section 4.4.1 introduces the questionnaire used to measure the participants view of how PSF would work at VCC. Finally, the research questions aimed to be answered in the validation are presented in Section 4.5.

4.1. Research methodology
The study presented in this paper is part of a technology transfer cycle consisting of innovation, static validation, dynamic validation and finally release of the technology for wider use. This general model for technology transfer developed for software engineering is shown in Figure 7 (see Gorschek et al. [53, 54]).

The innovation step is where the idea is born and developed into a technology. The technology is then tested and validated in different settings, from laboratory to industry, utilizing different types of research methods as needed. Static validation of technologies often involves experimentation in academia to investigate the basic concepts before the technology is tested in industry. Static validations in industry sort out teething problems related to transferring a technology to industry and provide a way to get initial feedback on scalability and usability without incurring the risks of testing it in production projects. Given the results of the static validation, one can either move back to the innovation step and refine the idea or move on to dynamic validation to test the idea in a real life setting. Dynamic validation is carried out in case studies in either “real” software projects or a smaller pilot project aimed at evaluating
the technology in question. The last step in the technology transfer process is to release the technology for wider use when it has been shown to be useful and usable. 

This paper presents a static validation of PSF in industry. The validation is set up as a workshop (described in Section 4.4) aimed at tailoring PSF to fit the organization, while also validating the concepts used in PSF. During the workshop PSF is tailored to fit the organization and usage of PSF is simulated by applying it to examples from the organization. Validation is achieved through collecting the workshop participants’ opinion on how this tailoring would work in a real setting. This gives initial feedback on what concepts work and what needs to be changed before actual piloting and thus build commitment to the solution.

**Figure 7: Technology transfer process (adapted from [53])**

### 4.2. Research context

The evaluation presented was conducted at the Swedish automotive manufacturer Volvo Car Corporation (VCC) currently owned by Ford Motor Company. Process descriptions at VCC are documented in a Business Management System (BMS). Instructions in the BMS guides what to do in projects and technologies are thus already chosen when commencing a project. Differences between projects stem from differences in system characteristics, i.e., safety critical systems need additional practices. Concerning knowledge management and retaining experiences gained from projects, postmortems are carried out on project conclusion and experiences documented in white books and provide feedback to instructions in the BMS. Experiences from using practices are also disseminated by face-to-face communication and rotation of workforce. In addition, cross-function teams and reviews are also used to utilize the knowledge and experiences. Process documentation and knowledge management activities at VCC are established and successful albeit a potential for improvement was uncovered in the questionnaire used during the workshop in this evaluation. The improvement potential relate to how to effectively communicate and use the experiences gained from concluded projects. For more background information on the case setting see [55, 56].
4.3. Participants
The workshop was run with four participants selected based on expert knowledge in relation to the organization and working practices. Three are employed at VCC and the fourth participant is a PhD student tied to VCC with extensive knowledge from research projects carried out in the organization.

4.4. Workshop description
The static validation is organized as a workshop with participants all actively participating. During the workshop, PSF was tailored to fit the organization, and real examples of practices, projects and experiences from the organization were used as the main input. In essence, this was equivalent to actual use of PSF.

At the end of the workshop, a questionnaire designed to measure the participants view of PSF was filled in (see Section 4.4.1). The workshop consists of six steps described below. All the steps are driven by the participants knowledge of how work is carried out in the organization. This means that all examples used during the workshop comes from real projects, experience and practices.

1. Overview of PSF
First a general introduction to PSF describing the goals and purpose of the framework is given.

2. Definition of practice and relation to used processes
The second step in the workshop aims at establishing an organization specific definition of a practice. This is important so that all participants have the same terminology and perception of what is being discussed.

3. Choosing what meta-data to collect
What attributes to collect for the entities in the practice repository is then established. To avoid missing important attributes or choosing ones that are redundant or not relevant, these steps are example-driven. For each of the entities, real examples from the organization are used to test the selected attributes. This involves specifying practices, experiences and projects according to the attributes chosen.
   3.1. Practice
   3.2. Project
   3.3. Experience

4. Using PSF for selection and improvement
When all attributes are established, more examples, including all entities are developed. During this step, the participants are asked to reflect and discuss how the tailoring would perform with regards to usage i.e. finding improvement potential and supporting practice selection. The tailoring is also refined during this step to reflect issues uncovered.

5. PSF support structure
The roles and responsibilities as well as initial ideas on how to organize the feedback of experiences are discussed on the fifth step of the workshop. In addition, the structure used to organize practices in the repository as well as how to connect it to processes already used in the organization is established.
6. Questionnaire
The last step of the workshop consists of a questionnaire aimed to measure the participants view of PSF in this particular organization. The questionnaire is described in Section 4.4.1.

4.4.1. Questionnaire
A questionnaire is used at the end of the workshop to measure the participants views on the tailoring of PSF developed. A summary of the questionnaire can be found in Appendix A. The questionnaire is divided into two parts. The first part asks for information on the current situation relevant to PSF. This is used as a benchmark for the answers given in respect to PSF. The next part, collect opinions on how the participants think that this situation would change if PSF, as tailored during the workshop, were implemented all out. Most questions are designed to be answered on the scales shown in Figure 8. The figure shows how the views are converted into numbers. For example, if the subject thinks that there will be “less” effort this answer is given the value -2. The same goes for the other scales, if a subject thinks that the quality will be “better” the answer is converted into a 2, and 4 for “much better”, and so on. For the first part of the questionnaire, the participants are asked to give point estimates while on the second part they are asked to provide three estimates, worst case, likely case and best case scenarios. The worst and best case are meant to capture the real case with 90% confidence. This makes it possible to assess the level of uncertainty the subjects have on the particular question.

![Conversion of scales used in questionnaire](image)

4.5. Research questions
This section presents the research questions that the validation addresses. Each research question is listed in Table 5 with a description/motivation. The research questions drive the design of the validation and a mapping of the research questions to steps in the workshop (see Section 4.4) and questions in the questionnaire (see Section 4.4.1) are given in Table 5. For example, it can be seen in Table 5 that in order to answer Research Question 2, regarding the quality of documentation, three steps in the workshop are performed, Step 3.1, 3.2 and 3.3, all related to tailoring the practice
repository. In addition, the questionnaire is set up to inquire about quality of the current documentation, questions 7, 8, 11 and 12, and how this is expected to change when introducing PSF, questions 2.2, 3.2 and 4.2.

Table 5: Research questions.

<table>
<thead>
<tr>
<th>Research question</th>
<th>Description</th>
<th>Mapping to WS</th>
<th>Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 1) How can PSF be tailored to fit an organization.</td>
<td>To enable successful technology transfer and knowledge management the organization need the be able to tailor PSF to fit their needs and constraints.</td>
<td>All</td>
<td>11, 12</td>
</tr>
<tr>
<td>RQ 2) Does using PSF improve the quality of the documentation of practices, projects and experiences?</td>
<td>PSF supports both codification and personalization. For successful codification the quality of the resulting documentation need to good-enough to use for sharing and improving practices.</td>
<td>3.1, 3.2, 3.3</td>
<td>7, 8, 11, 12, 2.2, 3.2, 4.2</td>
</tr>
<tr>
<td>RQ 3) Is PSF easy and intuitive to use?</td>
<td>This research question concerns the usability of PSF. If PSF is not useable, there is a significant risk of abandoning the approach. Different aspects of usability are detailed in the sub-research questions.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RQ 3.1) Does PSF improve on practices related to finding out and evaluating practices used in the organization?</td>
<td>To enable sharing and improving practices used in the organization, practitioners need to be able to find suitable practices and evaluate these.</td>
<td>-</td>
<td>9, 10, 13, 14, 15, 16, 18, 5.2, 6.2, 7.2</td>
</tr>
<tr>
<td>RQ 3.2) What decision support material in PSF is most important?</td>
<td>PSF provides different decision support material used in improvement and evaluation. This research question concerns what aspects of PSF are most valuable for supporting decisions.</td>
<td>3.1, 3.2, 3.3</td>
<td>-</td>
</tr>
<tr>
<td>RQ 3.3) How is the effort for using PSF perceived?</td>
<td>For PSF to be easy to use, the effort for using it cannot be deterring for users.</td>
<td>4</td>
<td>2.1, 3.1, 4.1, 5.1, 6.1, 7.1</td>
</tr>
<tr>
<td>RQ 3.4) How is the initiation threshold to using PSF perceived?</td>
<td>To successfully implement PSF, the cost of implementation and for starting using the approach cannot be perceived as a hinder. This would risk failing the implementation.</td>
<td>All</td>
<td>-</td>
</tr>
<tr>
<td>RQ 4) Is the decision material provided by PSF, experiences and project information collected, suitable for decision making regarding evaluation and adoption?</td>
<td>For PSF to be suitable for making decisions regarding adopting and improving practices used in the organization, the benefits of using the approach need to outweigh the cost of using it. This research question thus essentially concerns the usefulness, bang-for-buck, of using PSF. Different aspects of bang-for-buck for PSF is detailed in the sub-research questions.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RQ 4.1) Is gathering and structuring practices in one place, a repository, useful for finding out what practices are used in the organization?</td>
<td>Does the benefit of being able to find practice used in the organization outweigh the cost of establishing and maintaining a practice repository?</td>
<td>5</td>
<td>9, 10</td>
</tr>
<tr>
<td>RQ 4.2) Are experiences collected from a project useful material to evaluate the project?</td>
<td>For experiences to be useful as decision support material for evaluating a project, the costs of collecting and analyzing them need to be lower than the expected benefits.</td>
<td>3.2</td>
<td>15, 16</td>
</tr>
<tr>
<td>RQ 4.3) Are experiences and project information collected from the organization useful material for evaluating practices used in the organization to identify potential improvements?</td>
<td>For experiences and project information to be useful as decision support material for identifying improvements, the costs of collecting and analyzing them need to be lower than the expected benefits.</td>
<td>3.1, 3.2, 3.3, 4</td>
<td>13, 14, 15, 16</td>
</tr>
<tr>
<td>RQ 4.4) Are experiences and project information collected useful material for evaluating practices used in other project to decide on adoption?</td>
<td>For experiences and project information to be useful as decision support material for evaluating practices for adoption, the costs of collecting and analyzing them need to be lower than the expected benefits.</td>
<td>3.1, 3.2, 3.3, 4</td>
<td>9, 10, 13, 14, 15, 16, 17, 18</td>
</tr>
</tbody>
</table>

4.6. Validity

This section discusses the threats to the validity of the study described in this paper. As described by Wohlin et al. [57] validity can be discussed in terms of construct, internal, external and conclusion validity.

4.6.1. External validity

The resulting tailoring developed during the workshop cannot be generalized to other environments. Other organizations may face different needs and have other constraints. However, this is not a threat to the validity as generalizing the actual tailoring is not the objective of the study. The generalization that is important to be able to make here is for PSF and the workshop used for the initial tailoring of PSF. As the study only includes one organization and a few participants the ability to generalize the results are limited. However, this is an initial study and the results warrants further investigation to confirm or dismiss the results presented.

4.6.2. Conclusion validity

The largest threat to conclusion validity in this study is to ensure reliability of the data gathered. For the workshop, the threat concern eliciting all participants views instead of an obfuscated view or only the view of one strong person. During the workshop, every participant actively participated in all steps and discussions. Even though there was not always an immediate consensus on issues, discussions usually resolved these and thus this threat is considered limited.

The questionnaire used to measure the participants’ view of PSF also poses a threat to the reliability of measurements. To limit this threat the questionnaire was extensively
reviewed prior to use to ensure that there were no erroneous or leading questions. The questions used are also derived from the research questions, which should also limit this threat.

The participants in the workshop pose a threat to validity as they are few and might only represent one view of PSF. However, the participants from VCC are all experts with good knowledge of how development is carried out and the state of knowledge management within the organization and the PhD student have extensive knowledge of working practices used and the organization. This threat is thus limited but the number of participants is still too small to draw any general conclusions.

4.6.3. Internal validity
Participants in the workshop not expressing their real opinions, because they feel restricted by the workshop being audio taped and the tailoring and examples used in the workshop photo documented, is a threat to the internal validity. The participants were assured anonymity and the researcher assured not to divulge any sensitive information in publications, which limits this threat.

4.6.4. Construct validity
Only relying on one single data source is a threat to construct validity. The results presented for the PSF validation relies on both data gathered during the workshop and from the questionnaire. Relying on several data sources should limit this threat to validity.

5. Results
The workshop ran for an entire working day. This section presents the results from the workshop answers the research questions posed in Section 4.5. First, the tailoring developed at VCC is presented and exemplified in Section 5.1 and then answers to the research questions are given in the subsequent sections structured according to the presentation in Section 4.5.

5.1. Tailoring of the practice repository at VCC
Tailoring is important for knowledge management and process improvement [20] as there is not a one-size that fits needs and restrictions of all organizations. The PSF workshop tailoring entails choosing what information (meta-data) to collect, how to collect it and how to use it. To try the tailoring, examples of practices, projects and experiences from the organization is used to illustrate and try out actual use of PSF. Deciding what meta-data to collect at VCC during the workshop often started with an initial proposal, later refined as examples were used. The next sections presents the tailoring of the resulting practice repository developed during the workshop. Section 5.1.1 presents the tailoring of attributes used to describe practices, Section 5.1.2 projects and finally Section 5.1.3 describes the attributes chosen to capture experiences. In addition, Section 5.1.4 presents an example from the workshop, where the tailoring has been used to describe a practice and an experience according to the tailoring developed.

5.1.1. Tailoring of practices
At the beginning of the workshop, it was recognized that practices, as described in the generic version of PSF, correspond to instructions in the Business Management
System (BMS) used at VCC to some degree. The difference is that instructions are more generic, a description of what needs to be done, while practices are more concrete i.e. how things are done. It was decided that keeping traceability to the BMS and adding some practice specific description would suffice for using PSF. An example of this from the workshop is the “MISRA C compliance” practice which is an activity performed to check if C code adhere to MISRA C [58]. This can be done manually or by using a tool, which means that there are two different practices that can be used for this activity. The practices thus have different values for the attributes. Using a tool scales well and thus have high return on investment, however, the initiation threshold is higher than using a manual check if you do not already have the tool (procurement time, license cost, training, tool setup etc.). Table 6 gives an overview of attributes considered important when describing practices at VCC.

**Table 6: Practice attributes.**

<table>
<thead>
<tr>
<th>Practice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Title</td>
<td>A descriptive title for the practice.</td>
</tr>
<tr>
<td>Purpose</td>
<td>A short concise description of the purpose of the practice.</td>
</tr>
<tr>
<td>Short description</td>
<td>A short description of the practice. The full description can be found in the BMS. Additionally, specifics for the practice not described in the BMS are described here.</td>
</tr>
<tr>
<td>Source reference</td>
<td>Source reference provides traceability where more information can be found about the practice. This includes traceability to the original description in the BMS, literature and courses.</td>
</tr>
<tr>
<td>BMS instruction</td>
<td>Traceability to the BMS instruction describing the practice.</td>
</tr>
<tr>
<td>Literature</td>
<td>Traceability to selected literature concerning the practice.</td>
</tr>
<tr>
<td>Course</td>
<td>A list of internal and external courses that are given concerning this practice.</td>
</tr>
<tr>
<td>Process connection</td>
<td>Process connection indicates where in the development process the practice is meant to be used.</td>
</tr>
<tr>
<td>Owner</td>
<td>Owner is the person responsible for maintaining the practice.</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>Prerequisites lists everything that need to be in place for using the practice.</td>
</tr>
<tr>
<td>Output</td>
<td>Output indicates what kind of result is produced by using the practice. Output can be a decision, analysis or document.</td>
</tr>
<tr>
<td>Required competence</td>
<td>Required competence lists the needed competence for successfully using the practice. Competence can be experience, knowledge, courses or certificates.</td>
</tr>
<tr>
<td>Tool support</td>
<td>This attribute indicates the tool support the organization has available for this practice.</td>
</tr>
<tr>
<td>Initiation threshold</td>
<td>Initiation threshold essentially captures the time to return on investment. This is given on a subjective scale ranging from very low to very high. This attribute indicates how easy it is to start using the practice.</td>
</tr>
<tr>
<td>Estimated return on investment</td>
<td>Estimated return on investment gives an estimate of the relationship between benefit and cost for the practice. This is given on a subjective scale ranging from very low to very high.</td>
</tr>
<tr>
<td>Pro</td>
<td>Pros indicates the generally accepted positive aspects of the practice in the organization.</td>
</tr>
<tr>
<td>Cons</td>
<td>Cons captures the generally accepted negative aspects of the practice in the organization.</td>
</tr>
</tbody>
</table>
5.1.2. Tailoring of projects

To characterize projects, attributes describing the size, as well as the level of change, in the project were chosen. The size is described using the number of personnel working in the project, the man hours used in the project as well as the lead time of the project. Level of change captures what type of project is carried out. Low level of change implies a maintenance project with little added functionality while high level of change implies new development projects. To characterize the outcome of projects four attributes where chosen, cost, time and quality deviation from plan and content fulfillment. To characterize the changes in pre-requisites during the project it was decided to add an attribute to capture the requirements stability. An overview of attributes to characterize projects at VCC is given in Table 7.

Table 7: Project attributes.

<table>
<thead>
<tr>
<th>Project Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The project name.</td>
</tr>
<tr>
<td>Description</td>
<td>A short description of the project.</td>
</tr>
<tr>
<td>Lead time</td>
<td>The projects actual lead time.</td>
</tr>
<tr>
<td>Staffing</td>
<td>How the project was staffed. This attribute is dived into the two attributes given below.</td>
</tr>
<tr>
<td>Personnel size</td>
<td>The number of personnel that was involved in the project.</td>
</tr>
<tr>
<td>Man hours</td>
<td>The number of man hours actually used in the project.</td>
</tr>
<tr>
<td>Level of change</td>
<td>Level of change indicates the type of project. A level of change of 1 indicates a maintenance project while a level of change of 6 indicates that the project was concerned with only new development.</td>
</tr>
<tr>
<td>White book</td>
<td>Provides traceability to white books documenting experiences and outcomes for the project. The attribute is optional and only used if a white book exists.</td>
</tr>
<tr>
<td>Supplier</td>
<td>Indicates if and what supplier was used in the project.</td>
</tr>
<tr>
<td>HW/SW or mix</td>
<td>Indicates if the product developed in the project is pure software, hardware or a mix of both.</td>
</tr>
<tr>
<td>Cost deviation</td>
<td>Capture the deviation between planned and actual cost for the project.</td>
</tr>
<tr>
<td>Time deviation</td>
<td>Capture the deviation between planned and actual time for the project.</td>
</tr>
<tr>
<td>Content fulfillment</td>
<td>Capture the deviation between planned and actual content for the project.</td>
</tr>
<tr>
<td>Quality deviation</td>
<td>Capture the deviation between planned and actual quality for the project.</td>
</tr>
<tr>
<td>Requirements stability</td>
<td>Captures the volatility of requirements in the project.</td>
</tr>
<tr>
<td>Comment</td>
<td>The comment field is used to supply additional details and explanations to why the classification looks like it does.</td>
</tr>
</tbody>
</table>

5.1.3. Tailoring of experiences

In PSF, project postmortems are performed on project conclusion to evaluate the performance of practices used in the project. The results are documented in the practice repository as experiences. An overview of attributes capturing experiences in at VCC is shown in Table 8. Experiences in PSF are used to inform other projects of
how a specific practice (or combination) has performed. The goal is to aid project stakeholders in deciding what practices to adopt to facilitate improvement. Improvement can come in the form of refining the practice to better suit the project, refine the practice description to better reflect actual use and/or uncovering practices that perform poorly so that replacement practices can be found. During the workshop it was decided to use experiences to keep traceability between projects and practices to be able to distinguish what practices are used in what projects. Traceability to the person responsible for the practice in the particular project is also stored to facilitate face-to-face communication. Keeping track of people with experience can alleviate project staffing and gives an account of which people to contact if one considers adopting practices.

To document practice performance evaluations three attributes were established during the workshop; return on investment, performance and resource use. Return on investment is an estimate of how much benefit, for the project, a practice brings in relation to resources and time invested. Performance reflects how well the practice fulfilled its purpose. Resource use is divided in two parts, first cost which is all fixed costs that are incurred on the project for using the practice, and then the variable cost in time. Experiences are not mandatory to document for all practices. Only practices that perform well or have a potential for improvement must be documented. To capture concrete suggestions for improvements, three attributes were added. These include pros and cons as perceived in the project, as well as recommendations. Recommendations are concrete remarks of what could be done to improve on the situation.

Table 8: Experience attributes.

<table>
<thead>
<tr>
<th>Experience</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>Traceability to practice</td>
<td>Traceability to the practice which the experience concern.</td>
</tr>
<tr>
<td>Traceability to project</td>
<td>Traceability to the project in which the experience was collected.</td>
</tr>
<tr>
<td>Person responsible</td>
<td>The person in the project responsible for the practice.</td>
</tr>
<tr>
<td>Organizational belonging</td>
<td>The organizational unit the person responsible belong to.</td>
</tr>
<tr>
<td>Return on investment</td>
<td>The relationship between benefit and cost as perceived in the project. The attribute is scored on a subjective scale ranging from very high to very low.</td>
</tr>
<tr>
<td>Performance</td>
<td>How the practice has performed in the project. The performance attribute is divided into the three attributes presented below.</td>
</tr>
<tr>
<td>Success</td>
<td>Success of the practice unveil how well the practice has fulfilled its purpose. The success is scored on a 100 point subjective scale where 50 indicate average performance.</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost is all fixed costs associated with using the practice in this project. This include license costs etc incurred on the project.</td>
</tr>
<tr>
<td>Time</td>
<td>How many hours have been devoted to using the practice in this project.</td>
</tr>
<tr>
<td>Pro</td>
<td>The advantages from using the practice as experienced in this project.</td>
</tr>
<tr>
<td>Cons</td>
<td>The disadvantages from using the practice as experienced in this project.</td>
</tr>
<tr>
<td>Recommendations</td>
<td>Recommendations are a way to share additional experiences from using the practice. Recommendations can for example be concrete advice on how to improve the practice.</td>
</tr>
<tr>
<td>Comment</td>
<td>The comment field is used to supply additional details and explanations to why the classification looks like it does.</td>
</tr>
</tbody>
</table>
Most of the attributes presented for experiences, projects and practices are open ended in nature. This was decided as there are few quantitative measurements in place that directly map to the attributes presented. However, if measurements are in place, they can be used for experiences or projects, and these should be used to complement open ended attributes or estimates based on expert opinion in order to increase the reliability.

5.1.4. Examples of use

To illustrate the attributes chosen, an example used during the workshop is given below. The example is from the use of a review practice for System Requirements Descriptions (SRD). The description of the practice is given in Table 9. The description shows that the review is generally considered to have a low initiation threshold, e.g., there is little time and cost for starting to use it. It is also considered to give high return on investment. These attributes are estimates from how the practice generally performs. Ideally these attributes in the practice description change over time to correspond to what the projects actually feedback from using the practice in projects.

Table 9: SRD review practice.

<table>
<thead>
<tr>
<th>Practice</th>
<th>Review of SRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source reference</td>
<td>BMS instruction</td>
</tr>
<tr>
<td>Literature</td>
<td>-</td>
</tr>
<tr>
<td>Course</td>
<td>-</td>
</tr>
<tr>
<td>Process connection</td>
<td>Create system solution</td>
</tr>
<tr>
<td>Owner</td>
<td>QR responsible</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>SRD released for review</td>
</tr>
<tr>
<td>Output</td>
<td>Review protocol and decision</td>
</tr>
<tr>
<td>Required competence</td>
<td>Some knowledge about reviews</td>
</tr>
<tr>
<td></td>
<td>Knowledge about the system construction</td>
</tr>
<tr>
<td></td>
<td>Needed expert knowledge</td>
</tr>
<tr>
<td>Tool support</td>
<td>-</td>
</tr>
<tr>
<td>Initiation threshold</td>
<td>Given competence</td>
</tr>
<tr>
<td>Estimated resource consumption</td>
<td>Low</td>
</tr>
<tr>
<td>Estimated return on investment</td>
<td>High</td>
</tr>
<tr>
<td>Short description</td>
<td>Review document from different perspectives</td>
</tr>
<tr>
<td>Purpose</td>
<td>Increase product quality</td>
</tr>
<tr>
<td></td>
<td>Increase document quality</td>
</tr>
<tr>
<td>Pro</td>
<td>No investments</td>
</tr>
<tr>
<td>Cons</td>
<td>Very dependent on participants</td>
</tr>
</tbody>
</table>

This review practice has been used in project Alpha (obfuscated for reasons of anonymity) and the experience from that project is shown in Table 10. The use of the review was estimated to have high return on investment but was still not considered to be highly successful. The reason for this was that the use of reviews was considered rather low cost even if they did not uncover many defects. The reason for this shortcoming is given in the comment field. To remedy this and improve the practice for future projects, the recommendation given was to use reviewers with different backgrounds in relation to the document authors. Another possibility was to use a different reading technique, e.g. perspective based reading.
It should be observed, that a seemingly trivial observation formulated through PSF can have larger implications. A seemingly low-cost, but low yield, practice like the reviews used can be formally recommended for update/improvement and chances increase as the next project reads experiences of the per-default used review practice.

Table 10: Experience from SRD review in Project Alpha.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traceability to practice</td>
<td>Review of SRD</td>
</tr>
<tr>
<td>Traceability to project</td>
<td>Project Alpha</td>
</tr>
<tr>
<td>Person responsible</td>
<td>John Doe</td>
</tr>
<tr>
<td>Organizational belonging</td>
<td>92000</td>
</tr>
<tr>
<td>Return on investment</td>
<td>High</td>
</tr>
<tr>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>55</td>
</tr>
<tr>
<td>Cost</td>
<td>0</td>
</tr>
<tr>
<td>Time</td>
<td>48+40 (meeting and preparation time)</td>
</tr>
<tr>
<td>Pro</td>
<td></td>
</tr>
<tr>
<td>Cons</td>
<td></td>
</tr>
<tr>
<td>Recommendations</td>
<td>Used reviewers that were too similar to the document writers. Should be more careful when staffing review teams.</td>
</tr>
<tr>
<td>Comment</td>
<td>Didn’t find many crucial comments about the solution</td>
</tr>
</tbody>
</table>

5.2. **RQ1: How can PSF be tailored to fit an organization?**

The tailoring presented in Section 5.1 shows that PSF can be tailored to reflect the needs of the organization and that data from the organization offer valuable testing during the workshop. The company data uncovered what is realistic to collect and how useful the information is. Often it was noted that having simple numbered attributes could not capture the finer details and it was decided to add comment fields that can be used to enrich the descriptions where appropriate. However, there is no guarantee that the initial tailoring of PSF will fit future needs. This means that the information in the repository should be continuously scrutinized to find ways to refine what meta-data to collect.

5.3. **RQ2: Does using PSF improve the quality of the documentation of practices, projects and experiences?**

At the time of the workshop, VCC had a defined and documented process in the BMS, which acted as an electronic process guide with all instructions and activity descriptions accessible through the intranet. During the workshop it was argued that using PSF would provide a feedback mechanism that enables the BMS to be updated and refined to better reflect what is actually done in projects and to more closely correspond to developers’ needs. Figure 9 shows how PSF was estimated to influence quality of the various kinds of documentation. The bars are summarized for all participants in the workshop. The black bar shows the most likely outcome while the grey and white bars show the high and low estimates. The high and low estimates
intend to capture the actual outcome with 90% certainty. From Figure 9 it can be seen that the quality of practices, experiences and projects is estimated to be better using PSF. The high and low estimates show that there is some uncertainty in the estimates; yet, quality is expected to improve even in the worst case. The hesitation regarding quality of practice documentation is explained by how PSF would be used. The prime way of using PSF was discussed to be gathering feedback from projects and letting staff close to projects document or change practices. If process documentation is still isolated from projects, there is not a huge advantage of using PSF. Quality of experience documentation, on the other hand, was expected to improve more than process documentation. At the time of the workshop, experiences from practices used in projects were seldom documented, and the use of attribute driven experience documentation provided in PSF would improve experience documentation. Using the attributes gives both structure aiding the documentation and assure a minimum level needed from the project. Rules can be set up enforcing that at least some experiences are captured from a project e.g. a project should document three good experiences and three that warrants improvement. Project documentation would also improve using PSF, much due to a lack of a structured documentation at the time of the workshop.

![Figure 9: Quality of documentation.](image)

### 5.4. RQ3: Is PSF easy and intuitive to use?

Using PSF includes documenting projects, experiences and practices, and later using this information for transferring practices between projects, learning from concluded projects and identifying improvements. To be easy to use, PSF should not incur too much added effort while still providing improved decision support. During the workshop, the PSF documentation effort was estimated, as can be seen in Figure 10. The effort for documenting experiences, projects and practices were expected to increase slightly (black bars in Figure 10) as there were seldom documentation of these entities at the time of the workshop. The figures reflect the expected frequency of documentation; a project is documented once per project while several experiences are
collected and documented each project. The estimation of effort for documenting practices show the highest uncertainty (white and grey bars in Figure 10) due to uncertainties in how PSF would actually be used. If experiences are gathered and practices are refined to reflect how practices are actually used in projects the effort was expected to be low or even less then at the time of the workshop due to that actual working practices do not change that much. If practices are continuously refined to reflect each new experience, then the effort was expected to be higher.

Figure 10: Documentation effort.

Regarding using PSF for locating and evaluating practices for adoption, and evaluating projects, the results from the workshop are inconclusive (see Figure 11 and Figure 12). The white bars in Figure 11 shows that the effort for using PSF for evaluating practices and projects was estimated to stay the same while the effort for finding practices used in the organization was expected to decrease slightly. However, the uncertainties in the estimates are high as shown by the black and grey bars showing the high and low estimates. The black bars in Figure 12 shows that the precision in evaluating projects and finding practices was estimated to improve using PSF while the precision for evaluating practices was estimated to stay the same as at the time of the workshop. The white and grey bars shows that the uncertainty in the estimate is high. During the workshop, it was discussed that the effort and precision of using PSF is dependent on how the process around PSF is set up and managed. Using a new practice always account for an initial cost and the change also needs to be compliant with internal and external process standards. This means that people with sufficient knowledge about these constraints have to make decisions about what practices to use. In addition, at the time of the workshop there were few occasions where project managers could choose what practices to use at VCC. This was stipulated by the process documentation in the
BMS. Thus, using PSF at VCC would be focused on improvement rather than selection i.e. using experiences gathered to improve existing practices and to identify practices that need replacements. Other organizations might have other constraints and less defined processes, thus enable more focus on using PSF for selecting practices, and thus transfer practices, together with experience, between projects.

Regarding the effort needed to implement and start running PSF the opinions during the workshop differed. Half of the participants considered implementing PSF a risky endeavor while half thought it reasonable to implement. Part of this ambivalence is rooted in that the intended scope of PSF at VCC was not thoroughly established during the workshop. Implementing PSF at VCC would essentially involve creating all practice entities and linking these to the correct BMS instruction. The number of BMS instructions included in the initial PSF effort (the number of processes included) determine the implementation cost. In recognition of this, it is prudent to start small with a limited scope and try PSF before extending the effort. This should limit the risks associated with implementing PSF and minimize the risk of failing before starting to collect experiences from development projects.

![Figure 11: Effort of use.](image)
Figure 12: Precision of use.

The estimates for how PSF would influence precision in decisions was mostly positive (see Figure 12). The uncertainty in the estimates for evaluating practices is rooted in that quality of decisions depends on the person making them. As with documenting practices, it was noted during the workshop that practitioners close to projects should evaluate practices for improvement to better correspond to needs in projects. This was explained by the fact that project staff, in contrast to line or managerial staff, has recent hands on experience and tacit knowledge from using practices. Finding suitable practices and practices that need improvement was estimated to improve by organizing the information in a few attributes that can be compared and structured. Organizing all practices in one place and connecting experiences to them was valued during the workshop as the most important decision support included in PSF. However, this also depend on appropriate tools that support comparing attributes between experiences and sorting out experiences that have interesting traits.

5.5. RQ4: Is the decision material provided by PSF suitable for decision making regarding evaluation and adoption?

Thus far, the results have shown that PSF was estimated to add some cost from collecting experience, refining practices and documenting projects. In return, PSF was estimated to improve quality of documentation and could improve the precision in decisions regarding what practices to improve and transfer between projects. To provide a reality check of these costs and benefits, a valuation of the relation between these, essentially return on investment of using PSF, summarized the questionnaire. This valuation reveals if PSF is actually worth using, i.e., if the benefits are expected to outweigh the costs. The results are shown in Figure 13. The black bars shows that all activities was estimated to provide benefits that outweigh the costs using them. The grey and white bars in Figure 13 unveils that using PSF for identifying improvements to practices is the aspect that was estimated as most likely to provide high value. Other
aspects show higher uncertainty. This is expected, as noted earlier, since selection is hard to implement at VCC, as there is little room for alternating practices used in projects. Still, selection of practices was estimated as having high potential, which was recognized during the workshop, and the difficulties discussed. To enable successful selection at VCC, strict rules for what can be selected and who makes these decisions needs to be established and managed. Improvement was discussed to be easier as PSF only high-light practices that need to be improved or changed. This is less intrusive and requires less management as it is only an assessment and do not directly influence on project performance. The effect comes later as a change to practices.

![Figure 13: Perceived value for the different aspects of PSF.](image)

Gathering all practices in one place and adding attributes to them was estimated to be beneficial. However, the estimate has a high degree of uncertainty. The positive parts have been mentioned before (see Section 5.4) as enabling finding practices easier. The possible downside of gathering all practices in one place is that it might lead to more formalized routines and that local practices needed to cope with local problems becomes overlooked [59]. This is not something inherent in PSF but rather an implementation issue and something that needs to be acknowledged. PSF aims at allowing for some room for projects to work as best suited, but can also be used to make practices more rigid. The latter case was valued less during the workshop.

Finally, the least valued aspect, still expected to provide positive return on investment, is using PSF for evaluating projects. Evaluating projects are an added benefit from using PSF in addition to selection and assessment of practices. The main advantage
comes from providing different pieces of information in one place through traceability e.g. white books, characteristics etc. However, the information collected regarding projects are first and foremost meant to be used to distinguish the context in which the experiences have been collected.

6. Discussion
This initial validation in industry showed that PSF has the potential for providing real benefits. The validation at VCC has naturally been focused on the parts of PSF that best suit the organization in question, which proved to be using it for refining practice descriptions and for identifying improvement potential. This means that the validation and results is mostly generalizable to organizations with similar disposition.

The inclusion of a “recommendation” attribute in the experience entity was valued during the workshop as a good way of capturing concrete advice for how to refine practices and to add descriptions to practices. One can browse the repository looking at the recommendations collected for each. This also provides a foundation for small step-wise refinements of practices.

Even though it was thought that selection and transfer of practices between projects was hard to achieve at VCC, the potential benefit of this way of working was acknowledged. All projects are to some part different in that it develops different products, has different staff, lead-times etc. This means that using different practices in different projects and using PSF to share experiences is possible but not as easy to achieve as using it for improvement. Selection of practices is a change that impact project management to a larger extent and might be an option in a later stage of implementation.

The results also indicate that for successful knowledge management it is not only a matter of choosing what information to capture and disseminate. The process and organizational issues impact the endeavor to a large extent. At VCC, this includes roles and responsibilities for documenting practices and selection of practices as well as for whom the information in the repository is aimed. This includes if the information, practice descriptions, are going to be used as an instrument for control or for documenting and disseminating how work is actually performed.

6.1. Further work
Even though this validation is limited, the results show potential and warrant further investigation. A larger dynamic validation, pilot projects, is needed to further understand how PSF would work in industry over time and how to overcome the difficulties uncovered in this initial validation.

7. Conclusions
This paper presents PSF, an Experience Factory approach, enabling lightweight experience capture and use by utilizing postmortem reviews. Experiences gathered concern performance and applicability of practices used in the organization gained from concluded projects. Experiences are used by project managers to determine if practices used in other projects are useful for them. Process managers use the experiences to determine if there are a potential for improvement of practices used in
the organization. This way of working effectively lowers the initiation threshold for working with an Experience Factory approach:

- Data collection with postmortem reviews are a practical approach for gathering experiences [49] and incur relatively low costs on the development organization.
- The effort for analysis and packaging experiences are limited by using data that is already analyzed and packaged (connected to specific practices).
- PSF builds on state-of-practice in practice selection. Professionals are likely to rely on evidence from trusted sources (colleagues) and from settings similar to their own over that of researchers or consultants [46].

PSF was developed in collaboration with industry and part of testing and transferring PSF into industry use is validation to get feedback from practitioners. Validation was performed to tailor and assure that the approach meets industry needs. The validation was set up as a full day workshop where PSF was used on real data from the organization. The use of PSF was then compared to current practices used in the organization, implying that PSF is validated in relation to successful state-of-practice approaches for knowledge management and process improvement. Overall, the results from the workshop were encouraging and the participants’ assessment of PSF and particularly the tailoring developed during the workshop was positive. Using PSF to select what practices to use in projects also shows promise. However, selection introduces risks in projects as what practices are used needs to be managed. Introducing new practices also introduces risks. The most promising aspect was using PSF to uncover improvement potential and grounding this in the development organization.

Acknowledgements
This research has been partly funded by the Swedish industry and government joint research program IVSS – Intelligent Vehicle Safety Systems – through the CEDES project. We would also like to thank the participants in the workshop.

Appendix A

<table>
<thead>
<tr>
<th>PART 1. Background information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Name</td>
</tr>
<tr>
<td>2. Role</td>
</tr>
<tr>
<td>3. Department</td>
</tr>
<tr>
<td>4. Project</td>
</tr>
<tr>
<td>5. Experience</td>
</tr>
<tr>
<td>6. What practices do you have experiences from using?</td>
</tr>
<tr>
<td>7. How do you document practices today?</td>
</tr>
<tr>
<td>8. How do you value the quality of the practice documentation today?</td>
</tr>
<tr>
<td>9. How do you communicate knowledge about practices today?</td>
</tr>
<tr>
<td>10. How do you value the quality of practice communication today?</td>
</tr>
<tr>
<td>11. How do you document knowledge about concluded projects today?</td>
</tr>
<tr>
<td>12. How do you value the quality of the documentation of concluded projects today?</td>
</tr>
<tr>
<td>13. How do you communicate knowledge about concluded projects today?</td>
</tr>
<tr>
<td>14. How do you value the quality of the communication of concluded projects today?</td>
</tr>
<tr>
<td>15. How do you use knowledge gained from concluded projects today?</td>
</tr>
<tr>
<td>16. How do you value the use of knowledge from concluded projects today?</td>
</tr>
<tr>
<td>17. Do you look at previous projects as input to how to do your next project?</td>
</tr>
<tr>
<td>18. How do you select practices to use today? What aspects are important?</td>
</tr>
</tbody>
</table>
19. How would you value the success of the project on these attributes where 3 is a total success and -3 is a total failure.
20. If you feel that there are important aspects missed in the previous questions, please provide them here.

**PART 2: PSF estimates**

**PART 2.1 Effort and resources**

1. How do you perceive the effort and resources required for setting up TSF?
2. Assume that practices are documented in the way established during the workshop. How do you value documenting practices in TSF on the aspects given below?
   2.1. The effort need for documenting practices compared to today?
   2.2. The quality of the resulting documentation of the practices compared to today?
3. Assume feedback of project information as established in the workshop. How do you value documenting projects this way on the aspects given below?
   3.1. The effort need for documenting projects compared to today?
   3.2. The quality of the resulting documentation of projects compared to today?
4. Assume feedback of experiences from using practices in projects as established in the workshop. How do you value documenting experience this way on the aspects given below?
   4.1. The effort need for documenting experiences compared to today?
   4.2. The quality of the resulting documentation of experiences compared to today?

**PART 2.2. Using TSF**

5. How do you perceive that finding out what practices are used in the organization would change on the aspects given below?
   5.1. The effort needed for finding out what practices are used in the organization compared to today?
   5.2. The precision of finding out what practices are used in the organization compared to today?
6. How do you perceive that evaluating your project would change on the aspects given below?
   6.1. The effort needed for evaluating a project compared to today?
   6.2. The precision in evaluating the project compared to today?
7. How do perceive that evaluating practices used in the organization would change on the aspects given below? Evaluating practices used in the organization is either carried out to distinguish practices that can be adopted to a project or to find improvement potentials in the organization.
   7.1. The effort needed for evaluating the practices in the organization compared to today?
   7.2. The precision in evaluating the organization’s practices compared to today?
8. What aspects are most important when evaluating practices in TSF?

**PART 2.3. The value of using TSF**

9. This part of the questionnaire concerns the value of using TSF. The value in essence concerns the return on investment or bang for the buck of using TSF. That is, the value gained from using TSF compared to the added overhead. How do you value the activities given below?
   9.1. Gathering practices in one place to facilitate finding them easily?
   9.2. Evaluating a project?
   9.3. Evaluating practices used in the organization in order to identify ones that can be used in a new project?
   9.4. Evaluating practices used in the organization to identify areas that potentially could be improved?
10. Would enabling evaluating practices used in other projects affect your decisions regarding what practices to use in your project?

**References**


Chapter V


Tool support for disseminating and improving development practices

Submitted to

Software: Practice and Experience
Tool support for disseminating and improving development practices

M. Ivarsson\(^1\), T. Gorschek\(^2\)

\(^1\)Department of Computer Science and Engineering Chalmers University of Technology
martin.ivarsson@chalmers.se

\(^2\)School of Computing
Blekinge Institute of Technology
tony.gorschek@bth.se

Abstract

Knowledge management (KM) in software engineering and software process improvement (SPI) activities pose challenges as initiatives are deployed. Most existing approaches are either too expensive to deploy, or do not take an organizations’ specific needs into consideration. There is thus a need for scalable improvement approaches that leverage knowledge already residing in the organizations.

This paper presents tool support for an Experience Factory approach for disseminating and improving practices used in an organization. Experiences from using practices in development projects are captured in postmortems and provide iteratively improved decision support for identifying what practices work well as well as improvement needs. An initial evaluation of using the tool for organizational improvement has been performed utilizing both academia and industry. The results from the evaluation indicate that organizational characteristics influence how practices and experiences can be used. Experiences collected in postmortems are estimated to have little effect on improvements to practices used throughout the organization. However, in organizations where different practices are used in different parts of the organization, making practices available together with experiences from use, as well as having context information, can influence decisions on what practices to use in projects.

1. Introduction

Software organizations must continually strive to maximize the utilization of the knowledge and experience internal to the organization for refining and improving their development practices. Development practices are ultimately based on the knowledge and experiences of the organization’s staff. The challenge for leveraging experiences regarding practices is that they are often localized to the individual or project in which they are gained. The challenge is amplified by trends to distribute software organizations and development [1] as experiences are more likely to be shared within groups of people that interact with each other. To overcome this challenge, knowledge management in software engineering has focused on documentation and use of experiences to generalize and distill best practices [2-5]. A potential problem with this formalization and generalization is a reduction of local practices needed to cope with uncertainty and local problems [6].

Performing postmortems upon project conclusion is closely related to learning from project experiences and supporting organizational knowledge by making experiences
explicit [7-9]. Postmortems can have a learning effect on an individual level, team-level and organizational level [10]. However, few studies have focused on the organizational level [11]. Even if experiences are captured and stored, they are not always reusable or even usable by their intended audience [12]. This becomes evident when looking at state-of-practice as reports of software engineers not learning from past mistakes [13, 14] and experiences owned by the organization being overlooked [14].

This paper introduces tool support for utilizing postmortems in an Experience factory approach [2] for improving and diffusing practices in an organization. Practices are in this context any method, technique, procedure, tool or model used for software development. Postmortem reviews are used to evaluate used development practices and the results from the evaluation are stored as experiences in the tool, which makes them available for other parts of the organization. Other projects can use the tool to find out what practices are used in the organization and how they have performed in other projects. This can be used as input to deciding on what practices to use in projects thus aid in diffusing practices that have worked well and also in transferring knowledge embedded in the practice. The process improvement organization can use the evaluations to find areas in which practices need improvement.

This paper presents an initial evaluation of the tool in academia and at three different companies, all with different characteristics. Participants in the evaluation use the tool in a number of scenarios related to practice improvement and diffusion and rate how it would change the current situation. The results indicate that in organizations where practice documentation reflect actual practice and projects are in a position to learn from each other, diffusion of practices can take place. However, using postmortem data to affect organizational wide practices are not seen as likely. In addition, the results highlight the need for trust and resources for enabling learning from experience. Trust relates to how different parts of the organization value experiences of others. If there is little trust in others experiences, learning will not happen. In addition, without dedicated resources for change available, even the best effort for identifying what needs to change will not lead to realization.

The paper is structured as follows. Section 2 provides an overview of research on the Experience Factory and postmortems and discusses their characteristics and possible limitations. In Section 3, an overview of the Practice Selection Framework (PSF) and associated tool support is introduced. Section 4 presents the design of the evaluation, including research questions to be answered and the actual design of the evaluation. The results from the evaluation is presented in Section 5. Finally, conclusions are presented in Section 6.

2. Background and related work

There are many methods and frameworks that take organizational learning into account. Knowledge Management in general, but also in relation to software engineering in particular, has discussed the importance of managing and utilizing knowledge and of spreading tacit and explicit knowledge in an organization [15, 16]. The recognition that practitioners themselves are the main company assets makes this even more evident. In software engineering, Experience Factory (EF) [2, 17] has been central for reusing life cycle experiences and products for software development. Experiences are collected from development projects and are packaged and stored in
an experience base. Packaging entail generalizing, formalizing and tailoring the information collected to be easy to reuse. The idea is that software development projects can improve their performance through the utilization of experiences from previous projects. The classical usage of EF utilizes the Quality Improvement Paradigm (QIP) [17] for software process improvement (SPI) supported by the Goal Question Metric (GQM) [17] for establishing project and corporate measurements. The instantiation of EF requires the creation of a new experience factory organization to be responsible for the analysis and packaging of the experience collected. Packaging is a difficult task in itself [18], and the cost of the EF organization is about 10% of the total organizational project budget [19]. This is not to say that EF has negative return on investment but rather that the commitment and thus the initiation threshold for such an undertaking is considerable. The most prominent use of EF as described above is the NASA SEL [19] (for a review of research on EF see [20]). However, even though the EF at NASA may have had a positive return on investment, few or organizations have the resources to implement and run a similar effort. To overcome this problem, the Knowledge Dust to Pearls approach [3, 4, 21, 22] capitalize on both analyzed (Pearls) and unanalyzed experiences (Dust). Unanalyzed experiences satisfies the short-term needs for experience sharing and are later analyzed and packaged into best practices.

The challenges with managing experiences point towards the difficulty in valuing tacit knowledge and deciding what knowledge should be made explicit. The cost of making knowledge explicit and the use of training as one major vessel for knowledge transfer are high-cost, and accuracy can be hard to gauge: i.e. what knowledge should be made explicit, packaged and transferred? The actual transference is also a challenge. For a project manager initiating a project, the choice of what practices to use is many times based on a “gut feeling” derived from experience and tacit knowledge [16]. Using e.g. predictive or process models that use large amounts of project data as input or using simulation can support practitioners [23], but transparency and, ultimately, trust can be an issue. In addition, the cost of keeping high-quality/high-accuracy data for every project up-to-date, making it suitable for input to the prediction simulation, again becomes a problem, as high initiation and maintenance costs increase the initiation threshold of any process improvement activity [24-26].

The concept of performing postmortems upon development project conclusion is closely related to the concept of supporting organizational learning, making experiences explicit, and can be seen as vital for process improvement activities [7-9]. Performing postmortems is both practical and low cost, offering clear benefits, while it also has a low initiation threshold, i.e. suitable for organizations of all sizes. The main problem is that organizations seldom perform postmortems, even if they are in the official process charter, as pressures to start the next project overshadow the good intentions of learning. Verner and Evanco report that, of 42 projects studied, only 33% had postmortem reviews [9], even though performing postmortems was associated with the production of high quality development artifacts and the ability to manage risk more efficiently throughout the development process [9]. As a result, the same mistakes propagated across projects over time.

Another problem is that, even if postmortems are performed, information is seldom shared across project boundaries [13, 27, 28], resulting in that the only vessel for knowledge transfer is the individual practitioner moving between projects. Methods for analyzing postmortem data for use on an organizational level exist but carry high costs [29, 30]. The project managers, who are the driving force behind selecting the
practices to be used in a specific project, might or might not be involved in the learning experience of another project. This problem is compounded by the fact that few organizations have procedures or structures in place to enable sharing of information [7, 9, 28]. This can be devastating, as good practices are highly dependent on project characteristics, i.e. what works for one project might not be suitable for another [31, 32]. Relying on only word-of-mouth rumors, good practices conveyed out of context with little or no additional information can create more problems than they solve.

Tools for experience storage and retrieval in software engineering have been developed and experiences from using these have been published. Related to this paper, experiences have been connected to electronic process guides (EPGs) [33-36]. The type of information kept is mostly development artifacts such as project documentation, templates and checklists. The advantage of using development artifacts already present is that it offers a low-cost way of collecting experiences which are readily available for ad-hoc reuse. The Visual query interface (VQI) [22] also provide structure and tools for organizing and searching for experiences. Experiences from using VQI have also mostly focused on reusing development artifacts [22]. Both VQI and Well of Experience (WoX) have been used for supporting developers in their tasks e.g. by documenting solutions to reoccurring problems [22, 37]. Other tools such as Skill manager [37] have been used for keeping track of competence and experienced staff in the organizations.

3. Practice Selection Framework

The tool support presented in this paper implements the Practice Selection Framework which is a tailored Experience factory approach in which postmortems are used to evaluate practices used in development projects. This Section introduces PSF and the tool developed to evaluate PSF. The main motivation for using postmortems to capture experiences is that few organizations have precise quantitative measurement programs to discern practice performance. In addition, the dedicated resources needed for running an experience factory is high [2]. Lowering the resource demand could enable more organizations to capitalize on experiences and knowledge inherent to the organization, for improvement. Using practices already available in the organization to transfer knowledge between projects limits the initiation threshold. Practices used in the organization can be stored in PSF, immediately making them available throughout the organization. Experiences collected then provide iteratively improved decision support for deciding what practices to use.

PSF (illustrated in Figure 1) is meant to provide decision support to utilize experiences captured in postmortems to enable organizational improvement. This is achieved by using part of the postmortem to evaluate the practices used in the development project and store these experiences in PSF to provide decision support for project and process managers.
Improvement often involves introducing a new method, tool, technique etc. Thus, practices are a good candidate for conveying experiences. The use of PSF depends on organizational characteristics. If the organization prescribes what practices to use in projects, PSF can be used provide feedback on organizational practices and bridge a potential gap between project and line organization (process improvement organization). Feedback of experiences can be used by project managers to improve current development practices. In addition, evaluations of prescribed practices can identify where variations in used practices are needed i.e. circumstances when prescribed ones do not perform well. If projects in the organization are free to select what practices to use, PSF can be used to aid diffusion of practices that have worked well to other projects. Projects are given access to all practices used in the organization and how these have performed in previous projects and can use this as input when deciding what practices to use in their project. In comparison to EF, the goal of PSF is not necessarily to generalize practices but rather to promote using local practices to cope with local problems [38].

PSF collects experiences from several development projects enabling prioritizing improvement needs on the organizational level, i.e., identifying what is most important to improve first given the organization’s goals instead of individual project’s goals [38]. Improvement in projects often happens as a reaction to a problem and might thus address issues not beneficial to the organization. For example, extensively adding features to an architecture in one project might render it hard to use in other projects. PSF is meant to relate to state-of-practice for documenting practices in the organization and not to introduce new requirements for practice descriptions. Experiences are then related to existing practices by traceability. PSF is described in more detail in the next Section where prototype tool support is introduced.
3.1. Tool support
To evaluate using postmortems to capture experiences as in PSF, prototype tool support has been developed. The purpose of the tool is to collect feedback on PSF and to identify functionality needed to use PSF efficiently.

The data-model used in PSF is the same as in EF [2] (illustrated in Figure 2). Experiences are traced to both the project in which it was captured and to the practice it concern. Keeping traceability to the context in which the experience have been captured enables evaluation of how practice performance is affected by context and also enables users to evaluate if the experience is valid in their own context.

To structure and assure a minimum level of data collected, documentation of practices, experiences and project are attribute-driven. The attributes for each entity should be tailored to fit the needs of the organization. Collecting subjective data place high importance on the definition of the attributes and that all users feeding back and using the information have the same understanding of the attributes. Some of the attributes used to capture experiences during the evaluation presented in this paper are given in Figure 3. Experiences are traced to both projects and practices. In addition, traceability to the persons responsible for using the practice in the project is kept. This enables users of PSF to locate knowledgeable people in the organization from which they can get additional information about the experience and the practice.
To distinguish how practices have performed in projects and thus identify practices that have worked well or warrant improvement, practice performance is evaluated from different perspectives. Examples of attributes used to document practice performance in the evaluation are given in Figure 3. Effectiveness concerns the degree to which the practice achieves its purpose. For example, for a review practice, effectiveness concerns the share of defects found in the artifact reviewed and efficiency concerns the rate at which defects are found, i.e., defects per hour. Evaluating these perspectives with expert opinion means that the precision in the estimate is rather low. To improve this, objective measurements should be used as input to the estimation if available. However, the important precision to have is whether the practice has performed well or not. This enables identifying good practices and ones needing improvement. Deciding on what practices to use in a project is a complex task that need to consider project and product factors, knowledge of staff etc. PSF is meant to provide additional input to this decision process by providing alternatives that are already used in the organization.

The tool currently implements two ways of finding relevant practices and experiences, through a process and project view [5, 33]. The process view is shown in Figure 4. The idea is to have practices connected to the process to enable users locating what practices the organization have experiences from using for specific parts of the process. In the evaluation, the process from CMMI [39] is used. If used in an organization, the process view should ideally be the process that is used in the organization.

The users can select any part of the process to get an overview of all practices available for it e.g. selecting the process “Elicit needs” gives all practices used in the organization to elicit needs. For each process, a ranking of practices are given provided how projects have evaluated them. This enables users to get an overview of what practices have been estimated to perform best and worst for the process and each performance attribute. In addition, each practice is presented with a summary of the performance attributes collected for it as shown in Figure 5.
To get a better understanding of how relevant the experience is and in what projects the practice have been used, users can access all experiences collected for the specific practice. An example of an experience used in the evaluation is given in Figure 6. To each experience, a short description of the project in which it has been captured is given. The full description of the project can also be accessed if needed.

The other way to access experiences are to access specific projects carried out in the organization and view experiences collected from each. This can be used to identify good and bad experiences from projects similar to the present one.
4. Research methodology

Practice Selection Framework has been developed in several steps. After the first concept was developed, it was tailored and evaluated in a workshop in industry. The results from the evaluation was overall positive and improvement proposals from the workshop was used to refine the concept. To improve understanding of using postmortems for organizational improvement, tool support for PSF was developed to enable further evaluation. Two different evaluations are presented in this paper, first an evaluation in academia and then one in industry. The research questions addressed in the evaluation are given in Section 4.1 and the design of the study is given in Section 4.2.

4.1. Research questions

This section presents the research questions that the evaluation addresses. Each research question is given in Table 1 with a description/motivation.

Table 1: Research questions.

<table>
<thead>
<tr>
<th>Id</th>
<th>Research question</th>
<th>Motivation/description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 1)</td>
<td>Is PSF usable?</td>
<td>This research question concerns the usability of PSF and the PSF tool. If PSF is not useable, there is a significant risk of abandoning the approach.</td>
</tr>
<tr>
<td>RQ 2)</td>
<td>Is PSF useful for improving practices related to diffusing good practices and identifying improvements?</td>
<td>For PSF to be useful for making decisions regarding adopting and improving practices used in the organization, the benefits of using the approach need to outweigh the cost of using it. This research question thus essentially concerns the usefulness of using PSF.</td>
</tr>
<tr>
<td>RQ 2.1)</td>
<td>Is PSF useful for identifying practices for adoption?</td>
<td>For experiences to be useful as decision support for adopting practices, the costs of collecting and analyzing them need to be lower than the expected benefits.</td>
</tr>
<tr>
<td>RQ 2.2)</td>
<td>Is PSF useful for identifying improvement potential?</td>
<td>For experiences to be useful as decision support for identifying improvements, the costs of collecting and analyzing them need to be lower than the expected benefits.</td>
</tr>
<tr>
<td>RQ 2.3)</td>
<td>Is PSF useful for identifying practices that have been used in similar projects?</td>
<td>For PSF to be useful to find practices used in similar projects, the costs of collecting and analyzing experiences need to be lower than the expected benefits.</td>
</tr>
<tr>
<td>RQ 2.4)</td>
<td>Is PSF useful for finding knowledgeable persons in an organization?</td>
<td>For PSF to be useful to identify knowledgeable persons, the costs of keeping track of experiences need to be lower than the expected benefits.</td>
</tr>
<tr>
<td>RQ 3)</td>
<td>What are the challenges for using PSF?</td>
<td>To understand the risks involved with implementing PSF, challenges to PSF usage needs to be understood.</td>
</tr>
</tbody>
</table>

4.2. Study design

The steps in designing the evaluations are given in Figure 7. First, the evaluation is planned, i.e., how to perform the evaluation. Planning for the evaluations is presented in Section 4.2.1. The evaluation aims to collect feedback on PSF usage. To enable
usage, tool support was developed to enable potential users to use PSF prior to implementing it. Tool support is presented in Section 3.1. The material used in evaluating the tool is presented in Section 4.2.2. The evaluation is first performed in academia to initially evaluate PSF and refine the tool prior to commencing industry evaluations. Starting with an evaluation in academia is a low-cost way to test and improve PSF before starting evaluations in industry. In addition, the academic evaluation also doubles as a test of the evaluation material and the PSF tool prior to industry evaluation. The results from the academic evaluation are presented in Section 5.1. The evaluation performed in industry is presented in Sections 5.2 and 5.3. Section 4.2.3 provides details on the operation of the evaluation.

![Figure 7: Study design.](image)

### 4.2.1. Plan for evaluation

The study presented in this paper intends to collect initial feedback on using postmortems for organizational improvement. The evaluation involves letting participants use the PSF tool to be introduced to using postmortem data for diffusing and improving practices used in the organization. The specific tasks performed by the participants as part of the evaluation are:

- Find useful practices to use for a specific task
- Find practices used in specific projects
- Find practices with improvement potential
- Find knowledgeable persons in the organization

The tool is filled with examples of practices, projects and experiences prior to the evaluation, of which some are presented in Section 3.1. All examples are from requirements engineering (RE) practices. After having completed the tasks, a questionnaire aimed at collecting the participant’s views on the PSF tool and how the PSF concept would work in their organization is completed. The questionnaire is presented in Section 4.2.2. After the questionnaire, an interview is held to capture additional information regarding how knowledge management and process improvement is currently carried out at the companies and to elicit improvement proposals for PSF. During the interview, the major challenges to using PSF are also elicited.
4.2.2. Evaluation material

To collect the participants’ view of how PSF would perform in their organization a questionnaire is used. The questions used in the questionnaire are derived to answer the research questions presented in Section 4.1. The performance of each task performed in the evaluation is estimated (the scales are given in Figure 8). For each task, different perspectives are estimated: the effort for performing the task, how PSF affect the precision or ability of performing the task, and the value of using PSF for performing the task. The value of using PSF for performing a task is an estimation of the tradeoff between the added effort needed to enable the task and the value that comes from performing it. Estimating the value for using PSF provides a reality check of the estimations, i.e., even if the precision improve there is no guarantee that it will provide benefit to the organization. These perspective have previously been used successfully in software engineering technology evaluations [40]. For identifying good practices and improvements, an additional perspective is added, the amount of trust placed in the data that is used for making decisions. PSF relies heavily on subjective estimations of practice performance. Using subjective estimations introduces uncertainties in the data gathered. Participants are asked to estimate how much trust they would place in the subjective estimations in PSF for making decisions regarding adopting good practices and identifying improvements.

Figure 8 shows how the perspectives are converted into numbers. For example, if the subject thinks that there will be “less” effort this answer is given the value -2. The same goes for the other scales, if a subject thinks that the value will be “better” the answer is converted into a 2, and 4 for “much better”, and so on. For the results presented in Sections 5.1 and 5.2, each perspective are summarized for each group in the evaluation, one for the academic evaluation and one for each company.

![Figure 8: Scales used in the questionnaire.](image)

4.2.3. Operation

The evaluation is performed individually and starts with a walkthrough of the tool and an introduction to the PSF concept. Then, the participants uses the tool and estimates
how PSF would affect state-of-practice. Table 2 shows the time used for each of the activities in the evaluation.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walkthrough</td>
<td>30 min</td>
</tr>
<tr>
<td>Usage and questionnaire</td>
<td>60-90 min</td>
</tr>
<tr>
<td>Interview</td>
<td>30-60 min</td>
</tr>
</tbody>
</table>

Due to time constraints, two of the industry participants could not take part in the full evaluation. Instead, a 1.5 hours walkthrough and interview was held with these.

5. Results

This section presents the results from the evaluations and discusses the research questions. First, the results from the academic evaluation are given in Section 5.1. The industry evaluation is presented in Section 5.2 and addresses research questions one and two. Section 5.3 presents challenges to using PSF and improvement proposals needed to improve the tool and thus address research question three.

5.1. Evaluation in academia

The aim of the evaluation in academia is to initially test PSF before starting evaluations in industry. To assure that results from the academic evaluation are as representative as possible for industry, PhD students participating in industry collaborations in their studies are used. The results from this evaluation are used to refine the tool prior to industrial evaluation. The participants in academic evaluation are given in Table 3.

<table>
<thead>
<tr>
<th>ID</th>
<th>Research area</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>RE, SPI</td>
</tr>
<tr>
<td>A2</td>
<td>Cost estimation</td>
</tr>
<tr>
<td>A3</td>
<td>RE, Test</td>
</tr>
<tr>
<td>A4</td>
<td>SPI, RE, test</td>
</tr>
<tr>
<td>A5</td>
<td>Test</td>
</tr>
</tbody>
</table>

The questions in the questionnaire concern estimating the change to current practice that would come from using PSF. The participants in the academic evaluation are asked to provide answers that reflect their understanding of industry practice from their industry collaborations. This means that the answers becomes specific to the companies that the students have worked with.

5.1.1. RQ 1: Is PSF usable?

With regards to the usability of PSF and the PSF tool, several aspects were estimated by the participants in the evaluation. All estimations are made on the scales presented in Section 4.2.2 and are in this Section summarized for all participants which means that the maximum value for estimations is 20 and the minimum value is -20. First, the effort needed for using PSF and the resulting quality of experience documentation are estimated (shown in Figure 9). All participants agree that PSF would increase the effort for documenting and gathering experiences. However, the increase is only expected to be minor. In return, the quality of experience documentation is expected to improve considerably. This is due to few experiences being documented today.
With regard to the usability of the PSF tool and the PSF concept, all participants agree that usage of the PSF concept would improve from having better tool support. In addition, some modifications to the PSF tool were also seen to be needed to improve usability. The estimates for usability of the PSF tool and how the usability of PSF would change with sufficient tool support is given in Figure 10.

5.1.2. RQ 2: Is PSF useful for improving practices related to diffusing good practices and identifying improvements?

The estimations related to finding good practices in the organizations are given in Figure 11. It is estimated that using PSF would lower the effort for finding out what practices work well in the organization and that the precision in finding out what practices work well in the organization would improve. The participants also estimate that enabling finding out what practices work well would provide value to the organization e.g. by diffusing good practices.
It is also estimated that the subjective estimations used as decision support in PSF is trustworthy in identifying good practices. As long as subjective estimations are only used to discern what have worked well in a project, it is seen as trustworthy. However, it might be hard to discern changes in practice performance over time or between projects.

For identifying improvements, the results are similar to identifying good practices (see Figure 12). However, the value for identifying good practices is estimated to be higher than for identifying improvements.

PSF is also seen to provide value in identifying experiences from specific projects (see Figure 13). Projects can identify similar previous projects and use the experiences from these as input, i.e., practices that have work well in similar projects might be useful in new projects.
PSF is also estimated to provide a better overview of knowledgeable persons in the organization (see Figure 14). In comparison to other knowledge management approaches for mapping competence and knowledgeable persons [14, 37], PSF also provides context and valuation to the experiences that persons have. This means that it is easier to identify persons that have experiences from similar contexts e.g. similar problems.

Overall, the results from the academic evaluation are positive. All perspectives of PSF usage is estimated to provide value over current practices. Improvements suggested to the PSF tool was implemented after the evaluation. All improvements are related to providing a better overview of experiences and projects in which they have been used. For all estimations presented in this Section, all but one participant were unanimous. Participants A1-A4 all had similar estimations while participant A5 estimated that using PSF would not provide any change, except for adding effort for documenting experiences.

**5.2. Evaluation in industry**

Since the evaluation in academia showed promising results, the evaluation was moved into industry to collect feedback on applicability. The evaluation involves three
different companies given in Table 4. The reason for including several organizations in the evaluation is to investigate how different organizations improve and diffuse good practices and what factors influence PSF usage. The companies were chosen as they all are geographically distributed. In a localized organization, experiences and practices are often exchanged by word of mouth.

Table 4: Companies in the evaluation.

<table>
<thead>
<tr>
<th>Company</th>
<th>Size</th>
<th>Project size</th>
<th>Products</th>
<th>Projects</th>
<th>KM and SPI approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1000+</td>
<td>Large projects (100+ persons over several years)</td>
<td>Systems</td>
<td>A few different types of projects. Many projects of similar type.</td>
<td>C1 uses a prescribed set of processes and practices in all projects. The process is documented in an EPG. C1 uses CMMI for process assessment and postmortems to learn from projects. In addition, an issue tracking system is used to document methodology and tool issues encountered in projects.</td>
</tr>
<tr>
<td>C2</td>
<td>1000+</td>
<td>Medium (10-30 persons over 1-2 years)</td>
<td>Embedded software and systems</td>
<td>A few different types of projects. Many projects of the same type.</td>
<td>C2 uses a prescribed process but chose development practices on a per product basis. The process is documented in an EPG while practices are documented in a wiki for easier modification. C1 uses CMMI for process assessment and postmortems to learn from projects.</td>
</tr>
<tr>
<td>C3</td>
<td>~50</td>
<td>The project size varies. In-house projects for customers are usually small (1-5 developers over a couple of months). Staff takes part in projects of all sizes as consultants.</td>
<td>Consultancy, Software systems</td>
<td>Different</td>
<td>Internal development follows a tailored SCRUM [41] process. Practices are informal and based on knowledge and expertise of staff. Experienced staff move from site to site and thus a contact network is upheld in which experiences and best practices are shared.</td>
</tr>
</tbody>
</table>

At C1 and C2, two full evaluation, as described in Section 4.2, were carried out. In addition, a walkthrough of the PSF tool and interview was conducted in organization C1 and C3. The reason for not carrying out the full evaluation was due to time constrains. The subjects are described in Table 5. The subjects were chosen based on expertise and experience.
Table 5: Participants in the industry evaluation.

<table>
<thead>
<tr>
<th>ID</th>
<th>Company</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>C1</td>
<td>RE, architecture, SPI</td>
</tr>
<tr>
<td>I2</td>
<td>C1</td>
<td>RE</td>
</tr>
<tr>
<td>I3</td>
<td>C1</td>
<td>PM</td>
</tr>
<tr>
<td>I4</td>
<td>C2</td>
<td>PM, RE</td>
</tr>
<tr>
<td>I5</td>
<td>C2</td>
<td>PM, RE</td>
</tr>
<tr>
<td>I6</td>
<td>C3</td>
<td>PM</td>
</tr>
</tbody>
</table>

5.2.1. RQ 1: Is PSF usable?

Starting with company C3, the effort for using PSF is seen as too high to be a viable solution. The company is small (about 50 employees) and documenting practices and experiences is seen as adding too much overhead. However, C3 still capitalize on experiences in the organization. Best practices are diffused by senior staff moving from site to site. In addition, as experienced employees have high mobility these also have a strong contact network and everyone know who to ask when faced with issues. The threshold for initiating contacts is lowered by first having had face-to-face contact. Company C3 manages to retain knowledge and experience in the company by having low staff turnover for senior personnel.

When it comes to companies C1 and C2, the perceived effort and quality of documentation compared to today is given in Figure 15 for both companies. The estimations are the summary of two persons at each company and thus the maximum value is eight and the minimum is minus eight. The effort is expect to increase at both C1 and C2 while the quality of experience documentation is expected to improve. Both companies currently use postmortems to capture and analyze experiences from concluded projects. Experiences captured often relate to process, customer and product issues and the added effort is seen to come from identifying and documenting experiences from practices. Today, practices are often only judged implicitly in postmortems.

![Figure 15: Effort required for using PSF and quality of resulting experience documentation.](image_url)

Figure 16 shows the estimations for usability of the PSF tool and PSF usage would change with improved tool support. Both companies estimate that PSF usage would improve with better tool support. The main area of improvement involves finding relevant experiences faster and is further discussed in Section 5.3.2. In addition, it was argued that not only process and project managers are responsible for assuring that
projects learn from the organization experiences. Everyone should take part in leveraging experiences inherent to the organization. This increases the importance of usability, as not only a few experts should be able to use the tool.

![Figure 16: Usability of the PSF tool.](image)

**Summary of RQ1:** To summarize research question one, the initiation threshold for using PSF is perceived as being high if practices are not already documented in the organization. In addition, tool support for PSF needs improvement to make it practically usable. More support is needed for finding and evaluating relevant experiences. Potential improvements are discussed in Section 5.3.2. Having insufficient tool support in the evaluation might affect the validity of the estimations. As the usability of the tool is estimated to need improvement, other estimates might be influenced negatively. There is also a noticeable difference in valuation between company C1 and C2. C2 is more positive to PSF usage on all perspectives for research question one.

### 5.2.2. RQ 2: Is PSF useful for improving practices related to diffusing good practices and identifying improvements?

PSF can support diffusing good practices and to identify practices that need improvement. For sharing good practices, these are currently shared with persons that you have contact with at both C1 and C2. At both companies, this means that practices and experiences from using these are seldom shared with other sites. However, using PSF to share good practices is estimated differently at the companies (see Figure 17). At C1, sharing practices is estimated to add no value. Even if PSF alleviate finding practices to adopt, the participants do not expect that practices would be adopted. At company C2, PSF is seen as supporting finding practices that have worked well in other places in the organization. Currently, practices at C2 are decided on a per product basis. Experienced staff at the site meet and decide on what practices to use or if new ones need to be developed. Using PSF it is expected that practices used in other parts of the organization can be used as input to this activity and that potential new practices to use can be found. The effort for finding out what have worked well in other projects is expected to decrease and PSF is expected to add value for identifying good practices. However, the value is dependent on the practices available. There needs to be a gap between different parts of the organization to leverage practices, i.e., there needs to be practices that add enough value to transfer. At company C2, this is expected to happen occasionally.
For finding improvement potential both company C1 and C2 estimate usage of PSF in a similar way (see Figure 18). PSF is expected to have little impact on finding improvement potential. However, the reasons for the valuations are different. At both companies, there is already a good understanding of what needs to be improved in individual projects. At C1 where practices are prescribed, using experiences collected in postmortems is not seen as reliable enough to make changes affecting practices used organizational wide. At C2, there is no need to make improvements known on the organizational level. Improvements deemed important enough are handled in the project. This is possible as practices only affect the individual project. However, at both C1 and C2, improvements initiated on the project level are localized to the project in which they are implemented. Experiences from them are only shared within the project or site at which they are implemented.

At company C2, a potential benefit of using PSF is to enable evaluating practices over project boundaries e.g. reuse of requirements needs to be evaluated by considering both the project in which reusable artifacts are produced and the projects that use them. This is currently not handled at C2.
Similar to identifying practices that have worked well in other projects, company C1 and C2 differ on the estimations for using PSF to identify practices and experiences from similar projects. At C1, identifying similar projects and practices used in these are not seen to add value as the experiences are not expected to be used. Project managers use white books from previous versions of products, i.e., projects that are directly related to the product, as input to new projects. However, aside from project management, experiences from other projects are seldom used as input to how to do development except for the experiences individuals bring to the project. This is to a large degree due to practices being prescribed, i.e., all projects officially use the same practices. However, informal practices are used when circumstances in projects, often time pressure, makes prescribed ones unfit. These are not documented but based on knowledge of staff in the project. The decision on how to proceed when prescribed practices are not usable thus depends on the staff in the project. One example from the interviews is a practice for reviewing requirements. C1 use a quite ambitious requirements review practice that if used often produces good results. As it is ambitious, it requires much effort to use. Some projects omit requirements reviews as the resources are not available while other projects use a scaled down review practice to ensure some verification of requirements. It is seen as important to not only have the prescribed practice documented but also alternatives that can be used when the prescribed one is not usable. However, these practices need to be managed and caution taken to not institutionalize practices to cope with situations that need to be addressed [42] i.e. a lack of time in projects might indicate that the prerequisites for the project or estimation practices do not work satisfyingly.

At C2, gaining access to practices and experiences from similar projects are seen as important and PSF is estimated to add value for this activity. Currently, experiences from previous versions of the same product is both explicitly used as input by using white books and implicitly by staffing the project with persons that have experience from the product. PSF would enable finding out what practices are used in similar projects and experiences from using these. At C2, it is estimated that practices used in projects developing different products but under similar conditions could be used in new projects. Gaining access to what practices have been used in similar projects and experiences from using these could influence the decision on what practices are used.
The part of PSF that is estimated to add most value is adding traceability to persons in the organizations that have experience from using specific practices in specific contexts (see Figure 20). Enabling identifying persons that have experiences from practices and being able to evaluate the context in which the experiences have been gained is seen as important. Adding traceability to knowledgeable persons enables sharing tacit knowledge. However, at both C1 and C2, it is seen as important to not be too intrusive on ongoing projects thus documentation is important for transferring knowledge.

**Summary of RQ2:** To summarize the results for research question two, it seems that the use of practices to transfer knowledge is dependent on organizational characteristics. Company C1 that have prescribed practices do not estimate that practices or experiences from using these are likely to be shared. However, at C2 where practices are chosen by the project, and documented to reflect usage of them, making practices and experiences from using them available to other projects is estimated to add value. However, practices are not expected to be transferred as is, but rather be an inspiration to how to tailor practices used in projects. Experiences collected in postmortems are expected to have a limited influence on improvements to practices used throughout the organization i.e. prescribed practices.
These results both corroborate and add to previous work on using formal routines to transfer knowledge [43]. In organizations where practices are used to certify the company, i.e., not to reflect actual practice, the usefulness of them to transfer knowledge is seen to be limited. This study shows that organizations that have practice documentation that reflect practice might benefit from sharing knowledge and experiences from these.

5.3. Discussion of the evaluations

This section presents challenges and improvements collected at the companies. Section 5.3.1 addresses research question three and presents challenges to PSF usage. Section 5.3.2 presents the improvements seen necessary to improve PSF usage. Section 5.3.3. Finally, the differences between the evaluation in academia and industry are discussed in Section 5.3.3.

5.3.1. Challenges to PSF usage

The evaluation uncovered that PSF is faced with many of the same problems as other improvement and knowledge management approaches [44-46]. The challenges uncovered in the evaluations are:

- Dedicated resources for change
- Resources for documentation
- Trust
- Legacy
- Measurement

PSF is developed to be lightweight in adding low overhead for identifying improvements and learning from success [8]. Still, without dedicated resources for change, the effort will be in vain. This is the case at C1 where, today, even the most urgent improvement needs suffers from a lack of dedicated resources. At C2, changes happen and practices are chosen on a per product basis. At C2, the added resources needed for using PSF is estimated to add value in leveraging the organization’s experiences regarding practices used, as resources are available to capitalize on them, i.e., if a practice used in the organization is perceived to be beneficial it will be adopted.

Organizations that do not currently have documentation of practices will most likely not be able to use PSF. The idea is that the practice contains most of the knowledge to be transferred. If practices are not currently documented, or the documentation do not reflect practice, the effort and resources needed to implement and run PSF will probably be too high.

Trust is a major determinant in having a successful knowledge management approach. If people do not trust the experience of another or do not value it, learning will not take place. This is the case at C1 where only trusted persons affect decisions regarding what practices to use. At C2, trust is not seen as an issue. If the conditions under which a projects have been carried out are understood, the experiences from it are considered to be trustworthy. Having an understanding of the product and the customer it is being developed for is seen as important to value experiences at C2.
Legacy is also limiting the potential for change. Products are seldom developed from scratch and legacy from previous products in the form of the product itself and related artifacts needs to be considered. Legacy influence the resources needed for changing practices. Changing the way requirements are specified might mean having to reengineering the specification while adopting a new practice for reviewing requirements might be a small investment. This challenge is most notable at company C1 where large projects are carried out.

Central to PSF is using subjective estimations to discern practices that work well and ones that can be improved. At both C1 and C2, postmortems are successfully used to identify improvement needs in projects. However, it is also seen that complementary approaches such as traditional process assessments are needed to not only focus on immediate project issues. For PSF, one issue is whether subjective estimates are good enough for discerning practice performance. In the evaluation, reliability of measurements was considered an issue determining the use of postmortem data. It is seen as important that the experiences collected do not only represent individuals’ perception. Thus, ensuring validity of experiences collected are important [44]. For improvements, subjective estimations are thought reliable enough to change the project in which the issue is uncovered, or following project but not to change organizational wide practices. For identifying good practices, subjective estimations are considered reliable. Nevertheless, even if a practice is estimated to have performed well in a similar project there is no guarantee that it will work well in another project. PSF can however provide additional input for deciding what practices to use. Integrating experience capture into regular engineering processes using postmortems is also seen as important. Experiences regarding practices are not expected to be documented and shared without a mandate from management.

One additional problem with subjective estimations is that it is hard to use these as input to quantify the potential return on investment of using new practices. This might lower the commitment for adopting new practices. Practices that have been very successful in other projects might be picked up by others, but these are estimated to come along less frequently than small improvements.

5.3.2. Improvement proposals for PSF

During the evaluations, improvements needed to make PSF applicable in practice were discussed with the participants. The major improvements to PSF are listed below:

- Effort for using PSF
  - Effort for finding relevant experiences
- Practices
  - Definition of practices
  - Documentation of practices

The effort needed to find relevant practices and experiences need to be lowered for PSF to be usable. If users do not immediately find relevant experiences or find an overwhelming number of experiences without a proper overview or summary there is a risk that they will abandon use of it [22]. The structure of the experiences, i.e., the process view, where experiences are related to processes were valued during the evaluation. Retrieving experiences is a goal oriented activity and users need to be able to locate relevant experiences easily. To improve usability enabling browsing the process for practices and experiences in the context of similar projects are needed.
Only viewing experiences and practices from similar projects limits the effort needed to analyze the information available. On the other hand, only viewing experiences from similar projects might also miss relevant experiences. Thus, users need to be able to choose what project attributes are important for the practice they are looking for. A review practice might not be influenced by the context in which it is used and thus all other projects can be considered while a practice for specifying requirements might not scale to all sizes of projects and only projects of similar size might be considered. Expert users should thus be able to choose what project attributes they consider affect practice performance and view projects that have similarities on these attributes while novice users can select only to view similar projects. In addition, specifying practice performance from several perspectives was seen as important in the evaluation. Different projects have different priorities and enabling finding practices that support the priority e.g. quality or cost, was seen as important. The current overview of performance attributes, i.e., ranked lists for each performance attribute, is not seen as good enough. Instead, users should be able to provide the priorities for the current project and get a summarized performance overview for practices. In addition, some additional attributes need better overview e.g. a composite of pros and cons for specific practices. Currently, users need to access each experience in the PSF tool to view all pros and cons collected for a specific practice.

An issue with using practices to disseminate experiences is that practices are very concrete ways of performing activities. This means that different projects will use slightly different practices. Instead of having each project document it as a new practice, hierarchies of practices is seen as a solution in order to keep the repository structured and to avoid having experiences fragmented over several slightly different practices.

Additional improvements include documenting dependencies between practices, i.e., the use of a practice depends on using another, and prerequisites for the use of a practice e.g. a practice used for a product line might not be usable for single system development.

5.3.3. Industry and academia

The results from the academic and industrial evaluations are different. In academia, PSF is estimated to provide higher value than in industry. One explanation is that PhD students taking part in the evaluation all perform studies in contexts similar to company C2. However, there is still an overestimation regarding documenting experiences to use for improvements. In the industry evaluation, documenting experiences to use for improvements was estimated to add little value as staff working in the project often already understands problems. However, a lack of resources often hampers realizing improvements as development is given priority over improvements. Using students in evaluations might thus show that improvements are possible, but not that they will actually occur in industry. Thus, care should be taken when using students in process improvement or knowledge management evaluations.

6. Conclusions

This paper presents tool support for Practice Selection Framework (PSF). PSF is an Experience Factory approach aiming to bridge the gap between different parts of the organization and share experiences regarding practices between these.
The tool was used in an initial evaluation of PSF. The evaluation aimed to collect feedback on challenges for implementation and improvements to PSF and the tool. The evaluation was limited in scale and more participants evaluating PSF and the tool are needed to ensure validity of the results. However, given the challenge of getting real industry practitioners to participate in an validation, thus this validation can be seen as a promising first step, where the actual design of the tool, as well as the design of the validation are parts of the contribution.

The initial results points to that to enable PSF usage, organizations need to already have a high degree of practices documented. Otherwise, the initiation threshold and effort needed to use PSF is too high. However, utilizing already existing documentation, and transferring it to PSF is a low cost way for sharing knowledge. Experiences collected in postmortems are not estimated to affect improvements to practices used organizational wide. However, if organizations use different practices in different parts of the organization, making practices available with experiences from usage, as well as context information, could influence decisions regarding what practices to use. Practices used in other parts of the organization add to the number of potential practices to use in projects. In addition, evaluating practice performance from different perspectives can aid projects in selecting practices when projects priorities change.

The challenges for diffusing practices points towards the trust placed in others’ experiences and resources available for change. Experiences are not likely to be used if they are not trusted. However, even if experiences are available and trusted, resources are needed to realize changes. Small changes are expected to happen without dedicated resources, but larger ones need commitment from management. Related to tool support, improvements are needed to enable practitioners to find relevant experiences quickly. There is a risk that PSF is abandoned if practitioners cannot quickly decide if usable experiences exist in PSF.

Future work with PSF includes creating a stability index pertaining to practices, which would enable practitioners to gauge the reliability of a practice. In addition to this, efforts are underway to pre-fill PSF with good practices, available as input to practitioners looking to solve or handle a specific task. As these efforts are undertaken further validation is planned.

References


An Empirical Quality Assessment of Automotive Use cases

Reprint from

RE06 – The 14th IEEE International Requirements Engineering Conference
Minneapolis/St. Paul, USA
September, 2006
An Empirical Quality Assessment of Automotive Use cases

F. Törner¹, M. Ivarsson², F. Pettersson², P. Öhman²

¹Volvo Car Corporation, ²Chalmers University of Technology
ftorner@volvocars.com, {martin.ibarsson, fredrik.pettersson, peter.ohman}@chalmers.se

Abstract
As functionality in vehicles grows more complex and development becomes distributed over several geographical sites, elicitation and visualization of requirements become more critical. This paper presents a set of evaluation criteria for the quality of use cases. The criteria are applied to use cases that are currently used in industry and developed according to current industrial practice. The paper presents statistics of quality defects that occur in industry and proposes academic solutions that may be applied to solve them. The study is based on 43 use cases from Volvo Car Corporation spanning three different function areas of a vehicle. The most common quality defect classes of the evaluated use cases are missing elements, irrelevant steps, incorrect linguistics and level of detail. Furthermore, it is concluded that a common taxonomy and cross team reviews are needed to further improve the quality and usefulness of use cases.

1. Introduction
As functionality in vehicles grows more complex and higher flexibility is demanded, requirement specifications need to achieve higher quality.

When development is distributed over multiple geographical sites, unambiguous requirements documents become even more important. Today many of the ambiguities are resolved through social interaction as the geographical distance allows it. However, this solution will not be possible when large OEMs (Original Equipment Manufacturers) distribute research and development departments over several sites. Many of the frequent defects identified in this study are the source of ambiguities.

As a consequence of these factors, use cases are increasingly employed to elicit and communicate functional requirements [10][25][26]. This puts high demands on use cases to adhere to quality attributes such as correctness, consistency and readability. In addition, high quality use cases enable a wider application in areas such as safety analysis [21] and testing.

The academic best practice is well known at the present [7][8][9][10] but less is known about the current industrial practice owing to a lack of published empirical data. This paper presents a quality assessment of use cases at Volvo Car Corporation (VCC). For confidential reasons no details can be given of individual use cases, but aggregated quality data from a set of 43 use cases are presented. The quality assessment, which
An Empirical Quality Assessment of Automotive Use cases

consists of a review process, identifies the most common defects found in the set of use cases. From this a number of improvements to the industrial process are proposed based on well-known academic research.

Several studies, such as [4][5][6][7] (see section 3), deal with guidelines for writing and evaluating use cases and their quality [12][18]. However, these evaluation strategies are not applicable to the industrial setting studied in this paper as plausibility cannot be evaluated nor can comparisons with expert use cases be made. Further, both [4] and [12] evaluate their respective guidelines’ efficiency while this paper considers the evaluation of the quality of pre-existing industrial use cases subjected to very simple guidelines. A new set of evaluation criteria for use cases has thus been developed based on the work of [4], [12] and [18].

2. Background

During the development of the third generation electrical system for vehicles a need for a more functional oriented development method was identified at VCC. To that time a process focusing on the development of the electrical components was employed. As a consequence a new development method was designed: Model based Electrical System Concept (MESC). The MESC process has a model-based approach and utilizes a subset of the UML [11], including, but not limited to, entities such as use cases, sequence diagrams, class diagrams and component diagrams. The methodology is further described in [1].

At VCC, functions that are closely related are grouped together in function areas, e.g. Locking. Several roles are connected to each functional area, e.g. function developer, system developer and component developer. The function developer is responsible for defining the behavior of an electrical function area. Working with modeling tools, the function developer describes functions with use cases, scenarios and sequence diagrams.

Use cases in MESC serve several purposes: handshaking functionality with product planning; input to system development; complete system test; basis for the driver manual etc. The first purpose of writing use cases at VCC is for the function developer and product planning (PPL) to be able to identify and agree on the user functionality at a high abstraction level. The use cases in combination with virtual expeditions, e.g. early prototyping, constitute a possibility to validate the functionality in an early stage of development.

The first customer of the use cases is the system developer. The use cases are used in collaboration with sequence diagrams to create the class structure and class diagrams. The system developer is also responsible for writing the requirements and connecting the functional requirements to the identified classes.

As described, the collection of use cases for a function area is the only formal input of functional requirements from the function developer to the system developer. This motivates the importance of correct, consistent and understandable use cases in order to reach good quality. Earlier research has shown that cost is reduced when errors can be corrected as early as possible in a development process [24].
The importance of high quality use cases is even further aggregated by the trend in the automotive industry that the number of OEMs is decreasing by acquisitions. This trend creates organizations with several internal development departments that will have to work together in a more international and multi-sited environment, which places higher demands on having correct and high quality requirement documents in order to avoid ambiguities.

Use cases at VCC are created on the basis of an internally developed template based mainly on [2]. A mockup example of how a use case can be modeled is shown in Figure 1.

<table>
<thead>
<tr>
<th><strong>Use Case:</strong></th>
<th><strong>Lock the car</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Owned by FA:</strong></td>
<td>FAX</td>
</tr>
<tr>
<td><strong>Revision History:</strong></td>
<td>001</td>
</tr>
<tr>
<td><strong>Reference:</strong></td>
<td>Other related documents.</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Allowing for the car to be locked.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Driver, tester</td>
</tr>
<tr>
<td>** Preconditions:**</td>
<td>Vehicle unlocked</td>
</tr>
<tr>
<td><strong>Main Flow of Events</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>The actor locks the vehicle by pressing the lock button on the remote control.</td>
</tr>
<tr>
<td>2</td>
<td>The system locks all doors.</td>
</tr>
<tr>
<td>3</td>
<td>The system enables the alarm after a delay of 15 seconds.</td>
</tr>
<tr>
<td>4</td>
<td>The system double flashes with all available turn indicators.</td>
</tr>
</tbody>
</table>

**Alternative Flow of Events**

1b | The actor locks the vehicle by pressing the lock button on the door handle. |
4b | The system flashes one time to indicate that vehicle were not able to enable all locks. |

**Special Requirement**

On market X the system shall lock only the front doors.

*Figure 1 - Mockup example of a use case*

In addition to the template a guideline describes how the use cases should be authored. The guideline presents a small set of rules that have the purpose of supporting the author while at the same time not imposing too many boundaries:

- The use case should be viewed as a process with a start and an end.
- Use cases should be kept simple; if they are too long they should be divided.
- Use case naming should describe what the use case does in the form of a short and active verb phrase.

Some examples are also given that show how functionality can be grouped together in use cases. This is the only guidance given on how to handle use case composition.

3. Related work

There have been a number of studies on guidelines for writing use cases. The now defunct ESPRIT 21.903 Cooperative Requirements Engineering With Scenarios (CREWS) project developed several tools to author narrative use cases and during that process developed guidelines for use case authoring [4].
The guidelines are split into two categories:

- Style guidelines, derived from their own model as well as current best practice, e.g. [7].
- Content guidelines, derived from theoretical research in linguistics and artificial intelligence, e.g. [6].

Several empirical evaluations of these guidelines [4] [5] concluded that guidelines do improve use case authoring. The guidelines and related experimental hypotheses provided in [4] have served as input to the criteria used in the evaluation presented in this paper.

Cox et al. suggested that the CREWS guidelines can be difficult to apply and in [12] propose a simplified set of guidelines, named CP, that in their evaluation perform at least as well. The heuristic assessment scheme used is based on four characteristics: Coverage, Coherence of logic and its readability, Consistency of structure and Consideration of alternatives.

Fantechi et al. [13] proposed linguistic techniques to support semantic analysis of use cases. In a series of experiments on use cases from industry they evaluated their proposed set of quality metrics through the employment of three different tools. They conclude that linguistic techniques effectively support the achievement of quality requirements in three different problem categories: Expressiveness, Consistency and Completeness. They do however conclude that natural language techniques do not cover these categories completely.

Anda et al. [18] studied different authoring guidelines and their effect on the quality of the use cases produced. In experiments involving students they found that quality attributes, especially understandability, of use cases benefits greatly from a combination of a well defined template and guidelines for writing flows. In addition to these, they suggest a number of style guidelines (based on the work of Achour et al. [4]) describing how to author each step in the flow of events. Their evaluation scheme aimed to evaluate readability, content, level of detail and consistency.

Further suggestions on guidelines have been offered by Firesmith [3], Cockburn [7], Armour et al [9] and Pettersson et al. [18] but none of these has been evaluated to a satisfactory degree.

Regarding employing use cases in the automotive industry, the research has been quite diverse and not as thorough. The most salient work regarding use cases in the automotive domain is that of Ian Alexander, spanning from use cases for systems engineering [20] and recycling of requirements [16] to the introduction of misuse cases [17].

Other works in the domain include the context-driven, as opposed to goal-driven, use case creation process proposed by Omasreiter et al. [14]. Use case modeling has gained the attention of both academia and the industry, but few empirical findings from industrial projects have been reported.
4. Defect classification

As Cox et al. [12] state, counting the number of times a certain guideline has been applied (or how many times it has been neglected) is a rather misleading way of judging the quality of a use case [5]. Further, they conclude that applying heuristics, as a qualitative assessment method for documents, appears to offer a suitable way to assess use cases, as with many aspects of software engineering.

The same reasoning is applicable to the evaluation of use cases utilizing the criteria presented here, which have hence been formulated as questions. A set of criteria was developed to evaluate the use cases. The set are based on the characteristics of a good software requirements specification as stated in IEEE Std. 830-1998 [15] as well as use case academic best practice. As use cases are primarily employed in communication between different stakeholders, readability is an important issue. The readability characteristic has thus been added to the ones described in [15].

Another attribute that is important to consider, unique to use cases, is the level of abstraction at which the functionality is described: too many details decrease readability, while too high a level of abstraction does not meet the industrial requirements of usefulness. Including too many details also limits the possibility for re-use and tends to put limits on system development. The main characteristics that must be assessed are therefore: Correctness, Consistency, Unambiguousness, Completeness, Readability and Level of detail.

Identifying defects related to the functionality described in the use cases requires extensive domain knowledge. As the evaluation team’s knowledge of the specific functionality described is limited, the evaluation scheme was developed to be able to assess the use cases irrespective of domain knowledge.

As mentioned earlier, CREWS [4] and Cox et al. [12] applied different evaluation schemes. These strategies are not applicable, however, as neither the plausibility of the functionality can be evaluated nor can comparisons with expert use cases be made. Further, both CREWS and Cox evaluate their respective guidelines’ efficiency, while this paper considers the evaluation of the quality of use cases subjected to very simple guidelines. Thus a new set of evaluation criteria for use cases was developed based on the works of CREWS [4], Cox et al. [12] and Anda et al. [18]. These are presented in Table 1, which also includes two columns with references to earlier guideline rules.
5. Methodology

The purpose of this paper is to evaluate the quality of use cases. This is achieved by identifying defects, according to the criteria defined in section 4, and determining their intensity.

5.1. Hypothesis

The hypotheses are set up to compare the defect intensities. The criteria are organized into three groups according to their corresponding size measure (see Section 6). To determine the defect types with the highest intensity, a statistical analysis of the empirical data is conducted based on the following hypothesis:

\[ H_C – There \ is \ no \ difference \ in \ defect \ intensity \ between \ the \ defect \ types \ within \ each \ group. \]

In addition, the statistical analysis enables identification of differences between function areas, which corresponds to the following hypothesis:

\[ H_{FAX,FAY,CZ} – There \ is \ no \ difference \ in \ defect \ intensity \ between \ the \ function \ area \ X \ and \ Y \ for \ defect \ type \ Z. \]

If these hypotheses can be rejected it is statistically determined that there is a difference in intensity between the defect types and function areas.
5.2. Design

The criteria were first evaluated in a small pilot study in which a small set of use cases was assessed. Following this study the criteria descriptions were clarified in an effort to reach a uniform interpretation of the criteria among the researchers.

An overview of the methodology used to assess the quality of the use cases based on the established criteria is shown in Figure 2. The three reviewers performing the study have a background in software engineering and electrical engineering and one has previous experience of use cases at VCC.

Each reviewer analyzed all 43 use cases included in the study, identifying defects according to the criteria presented in Table 1. The defects found were indicated in the reviewed document and references to the defect type were noted. Defect types correspond to the criteria from which the defect was derived.

An inspection meeting attended by all reviewers was held in which the use cases were walked through and the defects identified were discussed and then compiled in the defect log.

The inspection data were analyzed with statistical techniques [22] and checked for normal distribution. As a normal distribution was not found, the Kruskal-Wallis non-parametric test was chosen to test the hypotheses. Since a significant difference was found with Kruskal-Wallis, the multiple comparison procedure [23] was used to compare the defect intensities. The significance value for rejecting the hypotheses was set to 0.05 for all tests.
5.3. Threats to validity

The threats to validity can be discussed in terms of Conclusion, External, Internal and Construct validity according to [22].

**Conclusion validity.** The function areas belong to the same functional domain (the body domain). Hence it can be assumed that the use cases involved belong to a rather homogeneous group. The threats related to the statistical test used in the analysis are considered to be under control as non-parametric tests are used, which do not require a certain underlying distribution.

The reliability of measures can not be assessed, as the total number of defects in the use cases investigated is unknown. Furthermore, some defect types may be less difficult to identify than others, affecting the conclusions regarding the ordering of the defect types.

Fishing and error rate problems are present as many dependent variables are measured. This problem is reduced by using the multiple comparison procedure, which corrects for the family wise error rate. However, this weakens the statistical power of the analysis, increasing the risk of type II errors and causing a hypothesis not to be rejected when it should be.

**External validity.** It is difficult to draw general conclusions based on a single study. However, the data consist of use cases from the actual development at VCC, which decreases the threat to generalization as the setting is representative for the domain.

**Internal validity.** The reviewers’ backgrounds are slightly different. To limit the effect of these factors, contributing to differences in defect identification, a common inspection meeting was held.

**Construct validity.** The criteria defined may not cover all important quality properties. They are however based on standards for software requirement specifications and earlier research. The definitions of the criteria themselves may contain ambiguities as they are heuristics and thus influenced by the reviewers’ assumptions.

6. Empirical data

The empirical data collected for each defect identified consists of a defect ID, defect type and use case in which the defect was found. Table 2 summarizes basic size measurements for the complete set of use cases and for each function area.

<table>
<thead>
<tr>
<th>Use Case Metric</th>
<th>Complete set</th>
<th>FA1 set</th>
<th>FA2 set</th>
<th>FA3 set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of steps</td>
<td>531</td>
<td>160</td>
<td>251</td>
<td>120</td>
</tr>
<tr>
<td>Number of alternative Flows</td>
<td>88</td>
<td>17</td>
<td>49</td>
<td>22</td>
</tr>
<tr>
<td>Number of use cases in set</td>
<td>43</td>
<td>18</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

Criteria C1, C3, C4, C5, C6, C10 and C11 are related to the use case size, measured in steps. It is not relevant to relate criteria C2, C7 and C12 to use case size; instead they...
are counted per use case. The third group, consisting of C8 and C9, is related to the number of alternative flows per use case. Hence, the criteria are only comparable within each group.

For each use case the intensity of each defect type is calculated by counting the number of defects and weighting with the relevant size metric, e.g. for a use case with ten steps and containing five defects of the same type, the defect intensity will be 0.5 defects/step. These measures are then used to test the hypothesis and visualize the defect type distribution.

6.1. Complete data set

Table 3 gives the descriptive statistics of the complete data set. For confidentiality reasons the values are scaled with a factor.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.226</td>
<td>0.154</td>
</tr>
<tr>
<td>2</td>
<td>0.007</td>
<td>0.022</td>
</tr>
<tr>
<td>3</td>
<td>0.031</td>
<td>0.054</td>
</tr>
<tr>
<td>4</td>
<td>0.014</td>
<td>0.041</td>
</tr>
<tr>
<td>5</td>
<td>0.007</td>
<td>0.022</td>
</tr>
<tr>
<td>6</td>
<td>0.091</td>
<td>0.107</td>
</tr>
<tr>
<td>7</td>
<td>0.000</td>
<td>0.012</td>
</tr>
<tr>
<td>8</td>
<td>0.035</td>
<td>0.067</td>
</tr>
<tr>
<td>9</td>
<td>0.014</td>
<td>0.032</td>
</tr>
<tr>
<td>10</td>
<td>0.119</td>
<td>0.116</td>
</tr>
<tr>
<td>11</td>
<td>0.140</td>
<td>0.256</td>
</tr>
<tr>
<td>12</td>
<td>0.035</td>
<td>0.048</td>
</tr>
</tbody>
</table>

Figure 3 shows a box plot diagram of the defect types related to the step size metric. Each defect type in Figure 3 contains 43 measure points corresponding to the defect intensity of the use cases. The defect intensity scale is hidden for confidentiality reasons.
The first hypothesis (H_C) is rejected for all groups, which gives that there is a statistical significant difference in defect intensity between the defect types. Further, multiple comparisons of the defect types establish a partial order as follows.

For the defect group related to the step size metric, defect type C1 can not be distinguished from defect type C10, but C1 is found to be of a higher order than all other defect types. C10 and C1 are of a higher order than C3, C4 and C5. However C10 can not be distinguished from C6 and C11. All defect types except C3 are of a higher order than C4 and C5. To clarify, the partial order is visualized in Figure 4.

By the same means, for the group of defect types that are counted per use case, the following order was established: C12 is of the highest order but C2 and C7 can not be distinguished from each other. For the defect types related to the size measure alternative flows, it is established that C8 is of a higher order than C9.

6.2. Function Area comparisons

Figure 5 visualizes the differences between the function areas as a box plot by showing the distribution and median for each defect type per function area.
H_{FAX,FAY,CZ} were tested for all combinations of function areas and defect types. The following differences were concluded:

- FA1 has fewer defects than FA2 for defect types C9, C10 and C11.
- FA1 has fewer defects than FA3 for defect types C1, C8, C10 and C11.
- FA2 has fewer defects than FA3 for defect type C10.
- FA3 has fewer defects than FA2 for defect type C9.

No other statistically significant differences can be shown between the function areas based on H_{FAX,FAY,CZ}.

7. Discussion of results

This discussion is divided into four sections based on the ordering of defect types established in the analysis. First the defect types related to the step size measure are discussed, and similar discussions follow for the per use case and per alternative flow groups. Finally, there is a discussion of the differences between the three function areas.

7.1. Defect types related to the use cases’ number of steps

Defect type C1 has higher defect intensity than all defect types except C10. Since C10 is closer to C6 and C11 than C1 is, it is assumed that C1 has the highest intensity. Defect type C3 can not be distinguished from C6, C11, C4 or C5. However, it has a lower intensity than C10, which leads to the assumption that its intensity is closer to that of C4 and C5. These assumptions are not supported by statistics but by the box plot in Figure 3. Given these assumptions we focus the discussion on C1, C10, C6 and C11, in that order.

One of the primary causes of the defect type missing element, C1, is the exclusion of elements such as actors in the use cases’ action descriptions. This often leaves room for misinterpretations and possible misunderstandings. This may in turn lead to defect propagation from the use case model to later requirement and design documents. A possible cause may be that the guidelines used when authoring the studied use cases do not provide guidance to explicitly state all elements in each active verb phrase.

Incorrect Linguistics, C10, can basically lead to the same consequences as Missing Element. It is important to note that 51.7% of the original defects of this type were removed since they were judged to be of too low a severity. These removed defects were the result of the praxis that an actor was addressed by a synonym in the use case. i.e. “user” referred to the single actor “driver” in the actor list. The incorrect use of synonyms, homonyms and pronouns is most probably caused by the lack of a domain specific taxonomy. If no guidance is given, the writer tends to vary the language in order to, in his/her opinion, increase the natural flow of the text with a richer writing style. This however causes possible inconsistencies in technical documents such as use case models. The causes of the defects that are misreferences are mostly dependent on the language capabilities of the authors.

Irrelevant step, C6, captures defects related to when the author fails to treat the use cases as describing a black box system and the fact that each step should describe
interaction between an actor and the system relevant for the goal of the use case. Failure to achieve this may lead the reader off track, causing confusion and misinterpretations. Typical causes to the defect irrelevant step are: descriptions of something that happens inside the black box system, statements regarding system state or simple comments.

Level of detail, C11, is in this evaluation exemplified by very detailed entities such as variables and human machine interface design. This distracts the reader from the main objective, the functional behavior. This is especially important in interdisciplinary development projects, such as vehicle projects, where use cases are the basis for the communication between PPL and the engineers, who have different backgrounds and knowledge but need to understand each other. Inclusion of human machine interfaces, HMI, is inappropriate since it in most cases will be developed in a later stage than the functional behavior. Being part of design decisions, the HMI will probably be changed later, causing costly rework of the use cases. This defect also limits the possibility for easy reuse of use cases in later or parallel projects.

7.2. Defect types that are counted per use case
Misuse of precondition, C12, is the most frequent in this group. Many of the defects found of this type are conditions that are raised and later contradicted. This is a threat to the use case correctness which can, if undetected, propagate to later design documents.

7.3. Defect types that are counted per alternative flows
Misuse of alternatives flow, C8, is the most frequent in this group. These defects were often identified where the flow was too complicated and a better use of alternative flows would increase the readability as well as correctness of the use case.

7.4. Function area comparison
As the statistical analysis shows, there are some differences between use cases in the three function areas. Examples of this are that function areas 1 and 2 have fewer use cases with the missing element defect type than function area 3. The irrelevant step defect type also shows a considerable difference in the number of use cases in which the defect is present. These differences may have several causes, including the situation that separate development teams have used the same guidelines but interpreted them differently or that the functionality is different, which leads to differences in style.

8. Industrial practice improvements
The problems identified in this evaluation do not have straightforward solutions. However, in this section, we will present how the defects identified can be avoided or mitigated by applying methods available in academia.

A domain specific taxonomy is a collection of the entities available and known at an early stage in a development project. In an automotive context, a domain specific
A domain specific taxonomy would guide the use case author to a vocabulary that is stringent; the usage of interchangeable names for the same entity would be avoided, shifting the focus from selecting the right words to specifying the right functional behavior. This would provide a base for stringency among all use cases in the organization, also avoiding unambiguous documents. By applying a domain specific taxonomy the level of detail or level of abstraction is set for all systems. Unnecessary details are identified and removed by setting the level of detail so that the complete vehicle becomes the main system.

A need for well defined guidelines has also been identified. Earlier studies on guidelines focus on the formalization of linguistic rules. In this evaluation a need of guidance on another level has been identified.

First of all, the guidelines should give good support in selecting and composing the use cases from the functionality. It is also important that the guideline has clear style rules, which in combination with the proposed taxonomy can help the author to focus upon the functionality to be described instead of the process of writing a use case. Further it is important for the guideline to contain examples of some easy to understand functionality; these examples are critical for success since they set the trend, especially in an industrial context. A standard for what a good use case contains and how it is formulated is also recommended; a possible form may be as a checklist to support reviewers.

The defect types affecting the control flow of the use case should be addressed since they directly affect the functionality described. A way to avoid these defects would be to use some form of graphical visualization of the flows. The activity diagram type in UML could support this need. It would help in providing a good overview of the use case and keep track of possible dead ends and missed references. It would not only help the author in his goal to achieve correctness but it would also help the reviewers to comprehend the structure of large and complex use cases.

To further strengthen the quality, extended reviews are vital. A checklist should be used and certain care should be taken to address the problems stated in this evaluation. The criteria presented in this paper and the related heuristics can be used as a checklist. However, questions to evaluate the functional content must be developed and included in the checklist, since the criteria presented here only focus on the technical usage of the use case method.
An Empirical Quality Assessment of Automotive Use cases

Table 4 - Summary of defects and the proposed corrective actions.

<table>
<thead>
<tr>
<th>Defect type</th>
<th>Proposed process improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Improved reviews (checklist and cross team)</td>
</tr>
<tr>
<td>Missing element</td>
<td>Taxonomy, extended style guideline.</td>
</tr>
<tr>
<td>Level of detail</td>
<td>Taxonomy, extended guideline</td>
</tr>
<tr>
<td>Incorrect linguistics</td>
<td>Taxonomy, extended guideline</td>
</tr>
<tr>
<td>Irrelevant steps</td>
<td>Guideline</td>
</tr>
<tr>
<td>Flow control related</td>
<td>Extended guideline, activity diagram</td>
</tr>
</tbody>
</table>

What competence the review group should have is beyond the scope of this paper, but differences between how use cases are authored in the different functional areas have also been identified. This indicates that learning within the organization would increase if competences from different development teams participated in the reviews. This is important in order to spread the best practice within an organization. Table 4 presents the common defects and the corresponding proposed solutions.

9. Conclusion

To perform this assessment of use cases at Volvo Car Corporation a set of twelve criteria was selected from earlier research and modified for this context. The criteria are connected to the desired quality properties: correctness, consistency, unambiguosity, completeness, readability and level of detail. The criteria focus on use cases as a technique, which means that criteria for evaluation of functionality need to be added for each industrial context.

The evaluation of industrial use cases at Volvo Car Corporation shows that the defect types with the highest defect density are either Missing element or Incorrect linguistics for defects related to the steps in a use case. Misuse of preconditions has the highest density for defects related to the use case as a whole and Misuse of alternative flows for defects related to the use cases’ alternative flows.

Based on the defects identified, a number of improvements to the industrial process are proposed based on well-known academic research. Introduction of a domain specific taxonomy would avoid or mitigate the effect of several defect classes and set the focus on functionality by providing a set of predefined entities. Guidelines for use case composition and style would support the author in making the use cases relevant and understandable. On an organizational level, reviews are also considered important in order to detect defects as early as possible and encourage best practice propagation.
10. Acknowledgements
This research has been funded by the Swedish Automotive Research Program. Further funding has been provided by the Swedish industry and government joint research program IVSS – Intelligent Vehicle Safety Systems – through the CEDES project. We would also like to thank Volvo Car Corporation for allowing access to the documentation.

11. References


