Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 81 (2008) 972–995
The Journal of Systems and Software
www.elsevier.com/locate/jss

A practitioner's guide to light weight software process assessment and improvement planning

F. Pettersson [a,*], M. Ivarsson [a], T. Gorschek [b], P. Öhman [a]

[a] Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Göateborg, Sweden
[b] School of Engineering, Blekinge Institute of Technology, Box 520, SE-372 25 Ronneby, Sweden

Received 1 June 2007; received in revised form 21 August 2007; accepted 23 August 2007
Available online 4 September 2007

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 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.

Keywords: Software engineering; Software process improvement; Software process assessment; Requirements engineering

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 (Gorschek, 2006). Inductive methods, such as the quality improvement paradigm (QIP) (Basili, 1985), 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) (CMMI Product Team, 2006) and ISO/IEC 15504 (SPICE) (ISO/IEC, 2003–2006), 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

* Corresponding author. Tel.: +46 31 772 5721; fax: +46 31 772 3663.
E-mail addresses: fredrik.pettersson@chalmers.se (F. Pettersson), martin.ivarsson@chalmers.se (M. Ivarsson), tony.gorschek@bth.se (T. Gorschek), peter.ohman@chalmers.se (P. Öhman).

0164-1212/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2007.08.032

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 (Kuilboer and Ashrafi, 2000; Reifer, 2000). A typical capability maturity model (CMM) (Paulk, 1995) SPI cycle can for example take between 18 and 24 months to complete, demanding large resources and long term commitment (Zahran, 1998). 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 (Calvo-Manzano Villalón et al., 2002; Kuilboer and Ashrafi, 2000; Reifer, 2000). 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 (Fayad and Laitinen, 1997; Kitson and Masters, 1993). 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 (Basili and Green, 1994; Conradi and Fuggetta, 2002; Damian et al., 2003; El Emam et al., 2001; Herbsleb et al., 1997; Herbsleb and Goldenson, 1996; Jacobs, 1999; Rainer and Hall, 2003). However, basing improvements on quantitative analysis, as proposed by QIP, produces useful results only if the process exhibits significant maturity (Cattaneo et al., 2001). 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 improvement framework utilizing light weight assessment and planning (iFLAP). The constituents of iFLAP have previously been successfully applied in industry (Gorschek and Wohlin, 2003, 2004). 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 3 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" (Calvo-Manzano Villalón et al., 2002). A classic example of this continuous and in theory never ending cycle of improvement is seen in Shewhart–Deming's plan-do-check-act (PDCA) paradigm, which embraced the necessity of cyclic and continuous process improvement as early as 1939 (Shewhart and Deming, 1986).

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 (Gorschek, 2006).

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 (Briand et al., 2001). A well known example is Basili's quality improvement paradigm (QIP) (Basili, 1985), 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 (Basili, 1993). 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 (Briand et al., 2001; Zahran, 1998). 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.

2.2.1. CMMI

Capability maturity model integration (CMMI) is an integration and evolution of capability maturity model for software (SW-CMM) V.2.0C (SEI, 2007a), integrated product development capability maturity model (IPD-CMM) V0.98A (SEI, 2007b) and systems engineering capability model (SECM) (Electronic Industries Alliance, 1994) 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 (Ahern et al., 2003).

The appraisal methodology that is a part of CMMI is based on several appraisal requirements called appraisal requirements for CMMI (ARC). 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 standard CMMI appraisal method for process improvement (SCAMPI) (SCAMPI Upgrade Team, 2006) and is developed to meet all requirements described in ARC as well as those needed to be compliant with ISO/IEC 15504 (Ahern et al., 2003). In general CMMI supports three different classes of appraisals (CMMI Product Team, 2006; SCAMPI Upgrade Team, 2006). 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 80–640 person hours and Class C appraisals as little as 20–60 person hours (Zahran, 1998).

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.

2.2.2. ISO/IEC 15504

ISO/IEC 15504 or, as it is commonly known, software process improvement and capability determination

(SPICE), is an international standard for software process assessment influenced by the now retired CMM (SEI, 2007a), 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 (ISO/IEC, 1995) and ISO 15288 (ISO/IEC, 2002). 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 (SPICE User Group, 2006) and SPICE for SPACE (Synspace, 2005).

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 (El Emam et al., 1998; SPICE Development Team, 2007). The effort needed to perform ISO/IEC 15504 assessments ranges from 33 to 824 person hours (with a median of 110) (El Emam and Briand, 1997; SPICE Development Team, 2007).

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 (Basili and Green, 1994; Basili et al., 2002; Beecham et al., 2003; Calvo-Manzano Villalón et al., 2002; Conradi and Fuggetta, 2002; El Emam et al., 2001; Herbsleb et al., 1997; Herbsleb and Goldenson, 1996; Jacobs, 1999; Kuilboer and Ashrafi, 2000; Niazi et al., 2005; Rainer and Hall, 2003; Reifer, 2000; Schneider, 2000; Wiegers and Sturzenberger, 2000; Zahran, 1998) 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 (Wiegers and Sturzenberger, 2000). 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 (Zahran, 1998). 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) (Brodman and Johnson, 1995; El Emam and Briand, 1997; Hayes and Zubrow, 1995). 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 (Calvo-Manzano Villalón et al., 2002; Kuilboer and Ashrafi, 2000; Reifer, 2000). This is particularly the case for small and medium sized enterprises (SMEs) (e.g. less than 250 employees) (SME TechWeb, 2003) where time and resources are always an issue when it comes to process assessment and improvement (Calvo-Manzano Villalón et al., 2002; Kuilboer and Ashrafi, 2000; Reifer, 2000).

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 light weight type. Examples of this can be seen in the IMPACT project (Scott et al., 2001) 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 (Laryd and Orci, 2000) and Adept (Caffery et al., 2007).

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 (Basili et al., 2002; Conradi and Fuggetta, 2002; El Emam et al., 2001; Herbsleb et al., 1997; Herbsleb and Goldenson, 1996; Rainer and Hall, 2003) 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 (Basili and Green, 1994; Conradi and Fuggetta, 2002; Damian et al., 2003; El Emam et al., 2001; Herbsleb et al., 1997; Herbsleb and Goldenson, 1996; Jacobs, 1999; Rainer and Hall, 2003). 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 (Johansen and Jenden, 2003).

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) (Cattaneo et al., 2001). 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 (Kitson and Masters, 1993).

2.3.3. The guiding principles of iFLAP

The basic constituents used in iFLAP were first presented by Gorschek and Wohlin (2003, 2004) 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.

Fig. 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 Fig. 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

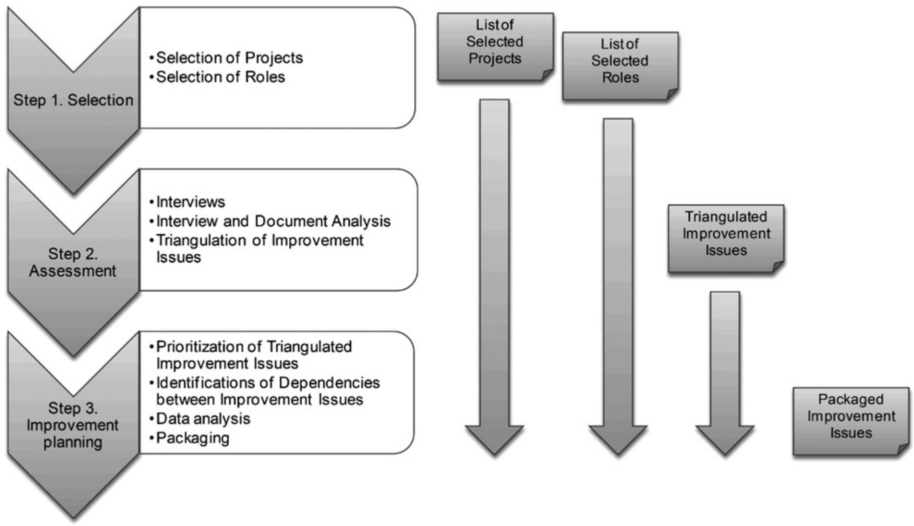

Fig. 1. Method overview.

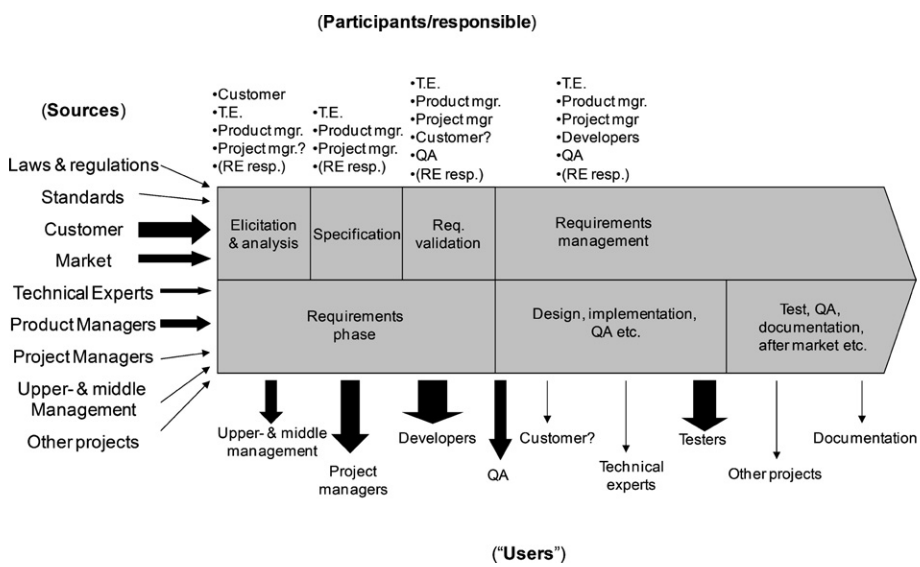

Fig. 2. Example of a requirements engineering process overview.

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.

3.1.2.1. 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 Fig. 3) while the line study consists of line interviews and an analysis of process documentation (C and D in Fig. 3). Triangulation of multiple data sources increases the validity of the findings compared to relying on a single source (Bratthall and Joergensen, 2002). 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.

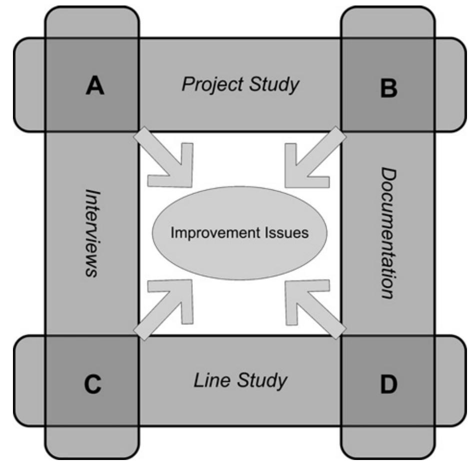

Fig. 3. Triangulation of data sources.

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 (Robson, 2002). 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 (Robson, 2002).

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 (IEEE Computer Society, 2004). 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 (1994), 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 (1994) and Robson (2002).

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 (Berander and Andrews, 2005) 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) (Gorschek and Wohlin, 2004). 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) (Saaty and Vargas, 2001), cumulative voting (Berander and Andrews, 2005), the top-ten approach and ranking (Berander and Andrews, 2005) 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 predefined 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. In contrast, 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 (2005))

Technique	Scale	Granularity	Scalability
AHP	Ratio	Fine	Low
Cumulative voting	Ratio	Fine	Medium
Ranking	Ordinal	Medium	High
Top-ten	–	Extremely coarse	High

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 (Regnell et al., 2001) 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 (Regnell et al., 2001) 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 (Saaty and Vargas, 2001), 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) (Kachigan, 1986) 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.

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 (ISO, 2005) 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 iden-

Table 2
Applicability of prioritization analysis methods

	Scale	Disagreement Charts	Satisfaction charts	Consistency ratio	Principal component analysis
AHP	ratio	X	X	X	X
Cumulative voting	ratio	X	X		X
Ranking	ordinal		X		
Top-ten	–				

tified 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.

5.1.1.1. 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.

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.

5.1.2.1. 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

Table 3
Selected projects

Department	Project	Description
A	Alpha	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
	Beta	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
B	Gamma	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
	Delta	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

Table 4
Selected roles

Department	Organization	Role	Description
A	Line	Domain expert	Technical expert in a specific technical area. Domain experts are one of the sources when eliciting requirements and are often involved in the specification
		Group Manager	Responsible for resources and competence development. Group Managers also have responsibility for the development process and its documentation
		Product Area Manager	Responsible for initiating (selling) and follow-up on projects. This also includes responsibility for the communication with customers
	Project	Tester	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
		Developer	The developers use the requirements specifications when designing and implementing the system
		Project Manager	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
B	Line	Group manager	See Department A above
		Quality Manager	Responsible for software quality assurance and process improvement
		Advanced Engineer	Technical expert that often works in research projects
		Product Area Manager	See Department A above
	Project	Project Manager	See Department A above
		Developer	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

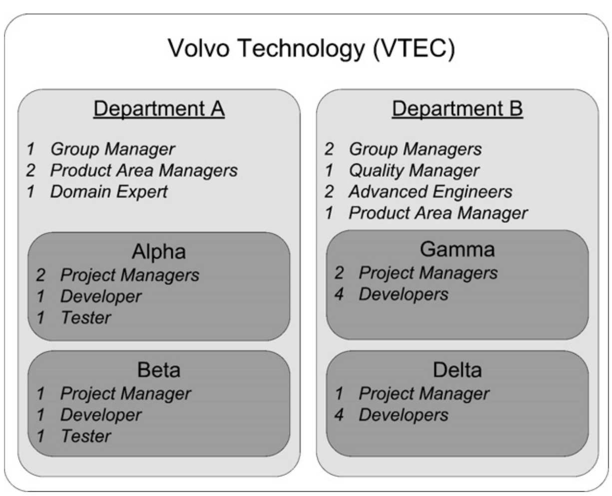

Fig. 4. Appointment of participants at departments A and B.

B are presented in Fig. 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.

5.1.2.2. 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.

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.

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

Table 5
Time spent in interviews

Department	Interviews	Mean time (h)	Standard deviation for time	Total time (h)
A	11	89	12.4	979
B	16	86	12.5	1376

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.

5.2.1.1. 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 (SEI, 2007a) and SWEBOK (IEEE Computer Society, 2004), 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.

Table 6
Identified improvement issues

ID	Name	Description
1	Abstraction and Contents of requirements	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
2	Allocation of requirements	Requirements are not allocated to different parts of design or to modules, which is viewed as an issue as it makes module testing more difficult
3	Customer relations	One important issue is that it is unclear who in the customer's organization that has the right to make decisions regarding system content. In addition, communication between project team and customer is in some cases unstructured (customers speak directly to developers which causes problems). Furthermore, customers have the tendency not to improve the quality of requirements over time and give late feedback on inquiries which causes late changes
4	Dependencies between requirements	There is no uniform way of analyzing dependencies between requirements and the results of such analyzes are not documented. This makes it more difficult to evaluate the effects of changes
5	Estimation	It is an issue that estimation of project time and resources are not good enough. A defined method to use when estimating project time and resources is needed and requirements engineering in general need more resources
6	Project vs. Product	Much of the work consists of product maintenance but is run as development projects. New requirements specifications are received for each project, resulting in consistency and reusability issues. Furthermore part of the work done, such as support, is not contracted
7	Requirements Engineering Process	There is a need of a usable requirements engineering process describing all activities, roles and responsibilities
8	Requirements Prioritization	Here prioritization means assigning a measure of importance to each requirement. It is an issue that it is not always clear which requirements that is the most important to fulfill and that there is no defined way of prioritizing them
9	Requirements Specification	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
10	Requirements Traceability	The most prominent issue concerning traceability is that, in some projects, it is not possible to trace tests to requirements. This makes it difficult to see whether tests cover all requirements or not
11	Requirements Upkeep	It is an issue that requirement specifications are not continually updated when changes are made during projects. This gives unreliable and unusable documents and subsequently hinders reuse as well as renders testing from requirements difficult
12	Roles and responsibilities in the RE process	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
13	State/progress of requirements	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
14	System tests from requirements	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
15	Requirements elicitation	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
16	Requirements version handling	Version handling is necessary to be able to identify different versions of the same requirement and thus be able to compare them

- 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.

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 Tables 8 and 10 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

Table 7
Improvement issue support numbers at departments A and B

ID	Issues	Support number at department A			Support number at department B		
		Project	Line	Total	Project	Line	Total
1	Abstraction and Contents of requirements	3	3	6	8	6	14
2	Allocation of requirements	0	1	1	0	0	0
3	Customer relations	5	4	9	5	3	8
4	Dependencies between requirements	1	0	1	3	1	4
5	Estimation	3	2	5	1	2	3
6	Project vs. product	0	0	0	1	2	3
7	Requirements engineering process	4	1	5	6	4	10
8	Requirements prioritization	1	0	1	1	0	1
9	Requirements specification	5	3	8	6	6	12
10	Requirements traceability	2	2	4	5	3	8
11	Requirements upkeep	0	1	1	5	3	8
12	Roles and responsibilities in the RE process	2	0	2	6	3	9
13	State/progress of requirements	1	0	1	0	2	2
14	System tests from requirements	3	1	4	5	2	7
15	Requirements elicitation	0	1	1	0	0	0
16	Version handling	1	0	1	0	0	0

Table 8
Triangulation matrix for department A

ID	Issues	Interviews		Documentation		Supporting sources
		Project	Line	Project	Line	
1	Abstraction and contents of requirements	X	X	X	X	4
2	Allocation of requirements		X	X	X	3
3	Customer relations	X	X	X		3
4	Dependencies between requirements	X		X		2
5	Estimation	X	X	X		3
6	Project vs. product					0
7	Requirements engineering process	X	X	X		3
8	Requirements prioritization	X				1
9	Requirements specification	X	X	X		3
10	Requirements traceability	X	X	X	X	4
11	Requirements upkeep		X	X		2
12	Roles and responsibilities in the RE process	X		X		2
13	State/progress of requirements	X		X		2
14	System tests from requirements	X	X	X		3
15	Requirements elicitation		X	X	X	3
16	Version handling	X				1

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.

5.2.2.1. 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 Tables 9 and 10 respectively.

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

ID	Issues	Interviews		Documentation		Supporting sources
		Project	Line	Project	Line	
1	Abstraction and contents of requirements	X	X	X		3
2	Allocation of requirements					0
3	Customer relations	X	X		X	3
4	Dependencies between requirements	X	X	X		3
5	Estimation	X	X		X	3
6	Project vs. product	X	X		X	3
7	Requirements engineering process	X	X	X		3
8	Requirements prioritization	X		X		2
9	Requirements specification	X	X	X		3
10	Requirements traceability	X	X	X		3
11	Requirements upkeep	X	X		X	3
12	Roles and responsibilities in the RE process	X	X	X	X	4
13	State/progress of requirements		X		X	2
14	System tests from requirements	X	X	X		3
15	Requirements elicitation					0
16	Version handling					0

Table 10
Normalized prioritization results

ID	Issues	Priorities at department A			Priorities at department B		
		Project	Line	Combined	Project	Line	Combined
1	Abstraction and contents of requirements	0.085	0.097	0.089	0.088	0.095	0.091
2	Allocation of requirements	0.048	0.117	0.071	0.036	0.048	0.041
3	Customer relations	0.072	0.097	0.08	0.066	0.067	0.066
4	Dependencies between requirements	0.055	0.047	0.052	0.066	0.043	0.056
5	Estimation	0.088	0.05	0.076	0.056	0.07	0.062
6	Project vs. product	0.053	0.067	0.058	0.078	0.068	0.074
7	Requirements engineering process	0.088	0.057	0.078	0.1	0.158	0.125
8	Requirements prioritization	0.072	0.043	0.062	0.038	0.035	0.036
9	Requirements specification	0.062	0.07	0.064	0.06	0.102	0.078
10	Requirements traceability	0.042	0.03	0.038	0.096	0.048	0.076
11	Requirements upkeep	0.082	0.057	0.073	0.085	0.052	0.071
12	Roles and responsibilities in the RE process	0.062	0.047	0.057	0.055	0.063	0.059
13	State/progress of requirements	0.087	0.063	0.079	0.048	0.038	0.044
14	System tests from requirements	0.062	0.07	0.064	0.079	0.052	0.067
15	Requirements elicitation	0.043	0.09	0.059	0.05	0.06	0.054

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.

Table 11
Weight of dependencies above 20% threshold at departments A and B

Department A			Department B		
From	To		From	To	
9	1	22%	2	1	21%
10	1	22%	1	7	21%
14	1	44%	2	7	21%
13	2	33%	4	7	21%
9	5	22%	5	7	43%
11	6	22%	6	7	21%
5	7	22%	8	7	21%
12	7	44%	9	7	29%
6	11	33%	11	7	29%
10	14	22%	12	7	50%
			13	7	21%
			14	7	21%
			15	7	43%
			10	9	21%
			14	11	21%

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 11. 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). Figs. 5 and 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.

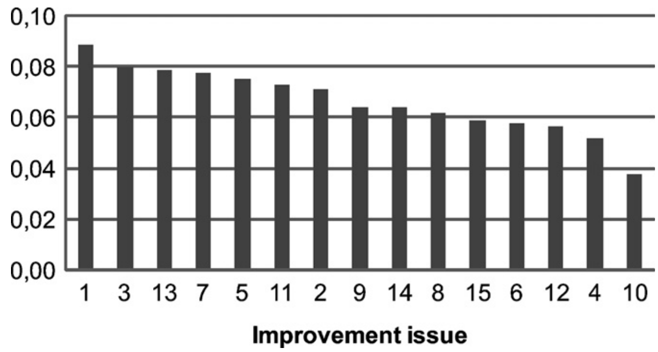

Fig. 5. Department A improvement issue prioritization.

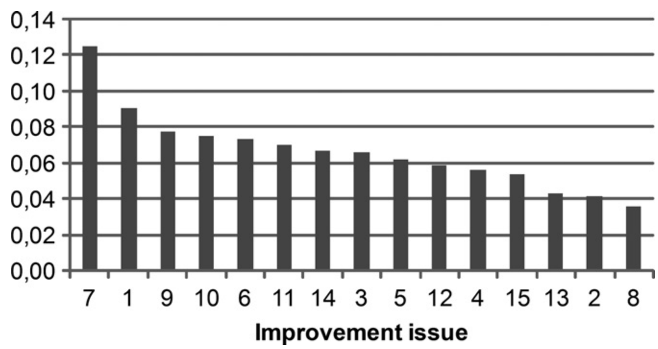

Fig. 6. Department B improvement issue prioritization.

5.3.1.1. 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 12. These are also illustrated in Fig. 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.

5.3.2.1. 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 Figs. 8 and 9. The satisfaction charts, showing how well each participant's priorities correlate with the department's average, are found in Figs. 10 and 11.

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

Table 12
Resource usage for the VTEC study

iFLAP step	Activity	Company	Assessor	Total (hour)
Step 1 – selection	Selection workshop	6	6	12
Step 2 – assessment	Interviews	39.25	78.5	117.75
	Interview and document analysis	0	80	80
Step 3 – improvement planning	Workshop 1	22	4	26
	Workshop 2	22	8	30
	Workshop analysis	0	2	2
	Packaging	3	6	9
Total (h)		92.25	184.5	276.75

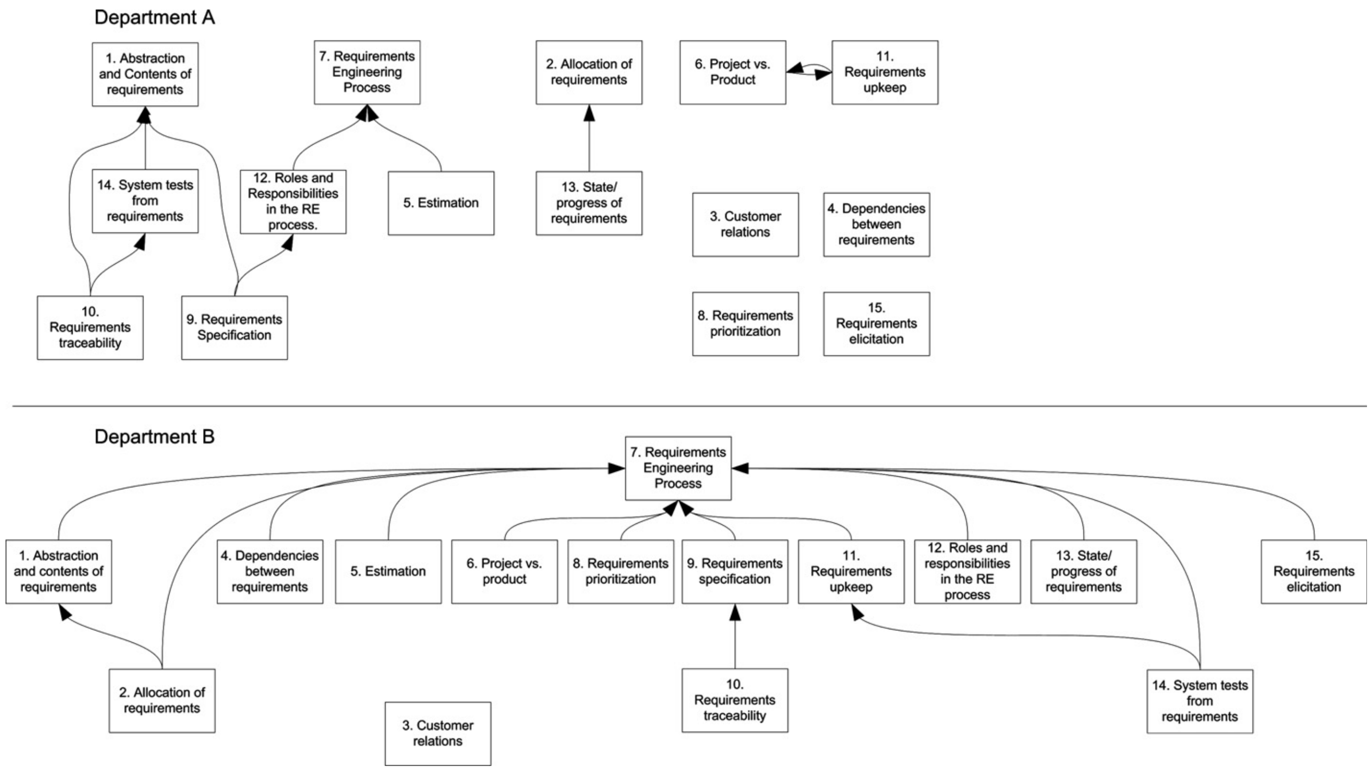

Fig. 7. Dependencies between improvement issues.

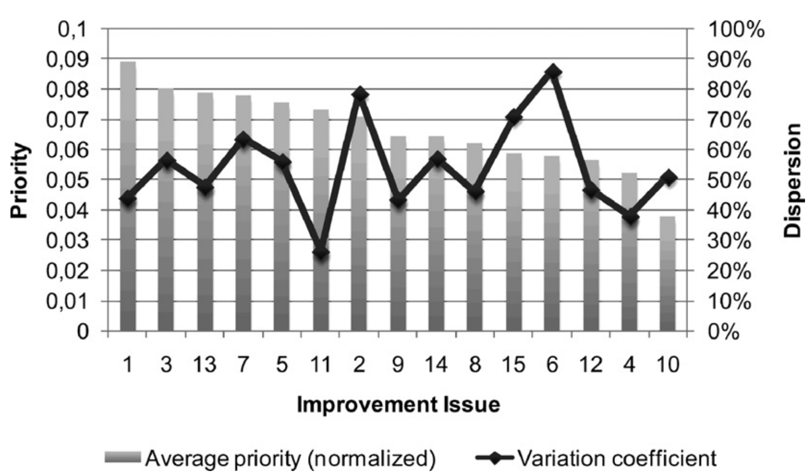

Fig. 8. Disagreement chart for department A (variation coefficient = std. var./mean %).

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.

5.3.3.1. *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

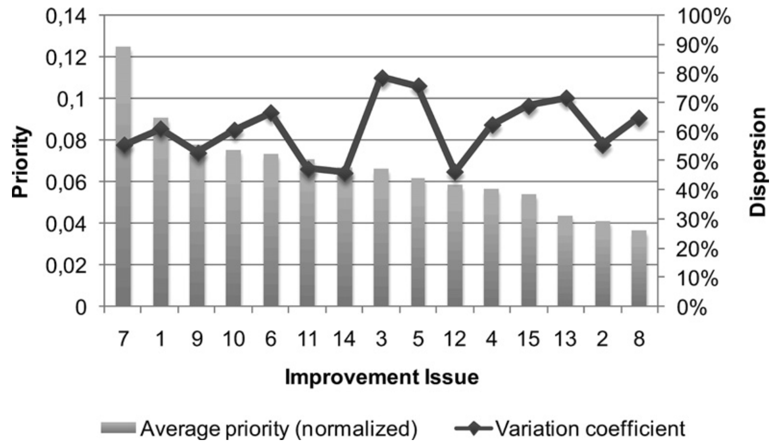

Fig. 9. Disagreement chart for department B (variation coefficient = std. var./mean %).

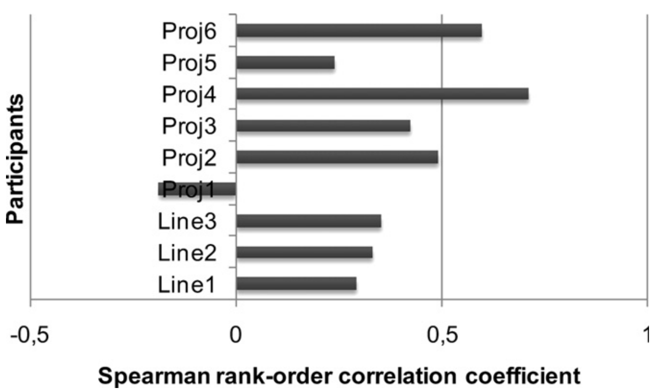

Fig. 10. Satisfaction chart department A.

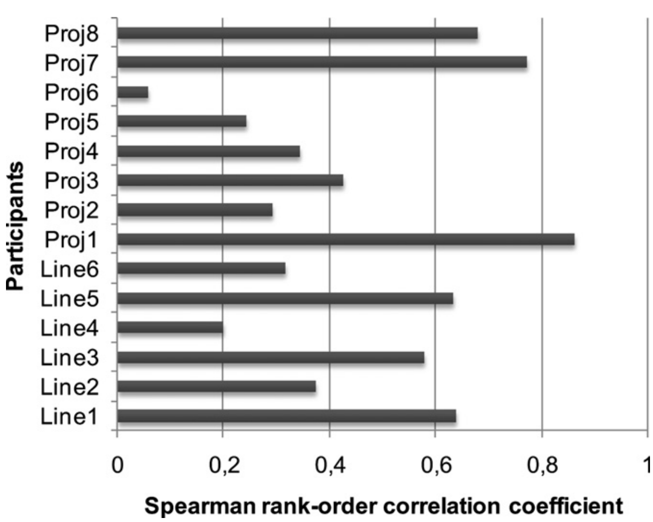

Fig. 11. Satisfaction chart department B.

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 (Karlsson and Ryan, 1997), 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) (Gorschek and Wohlin, 2006) 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

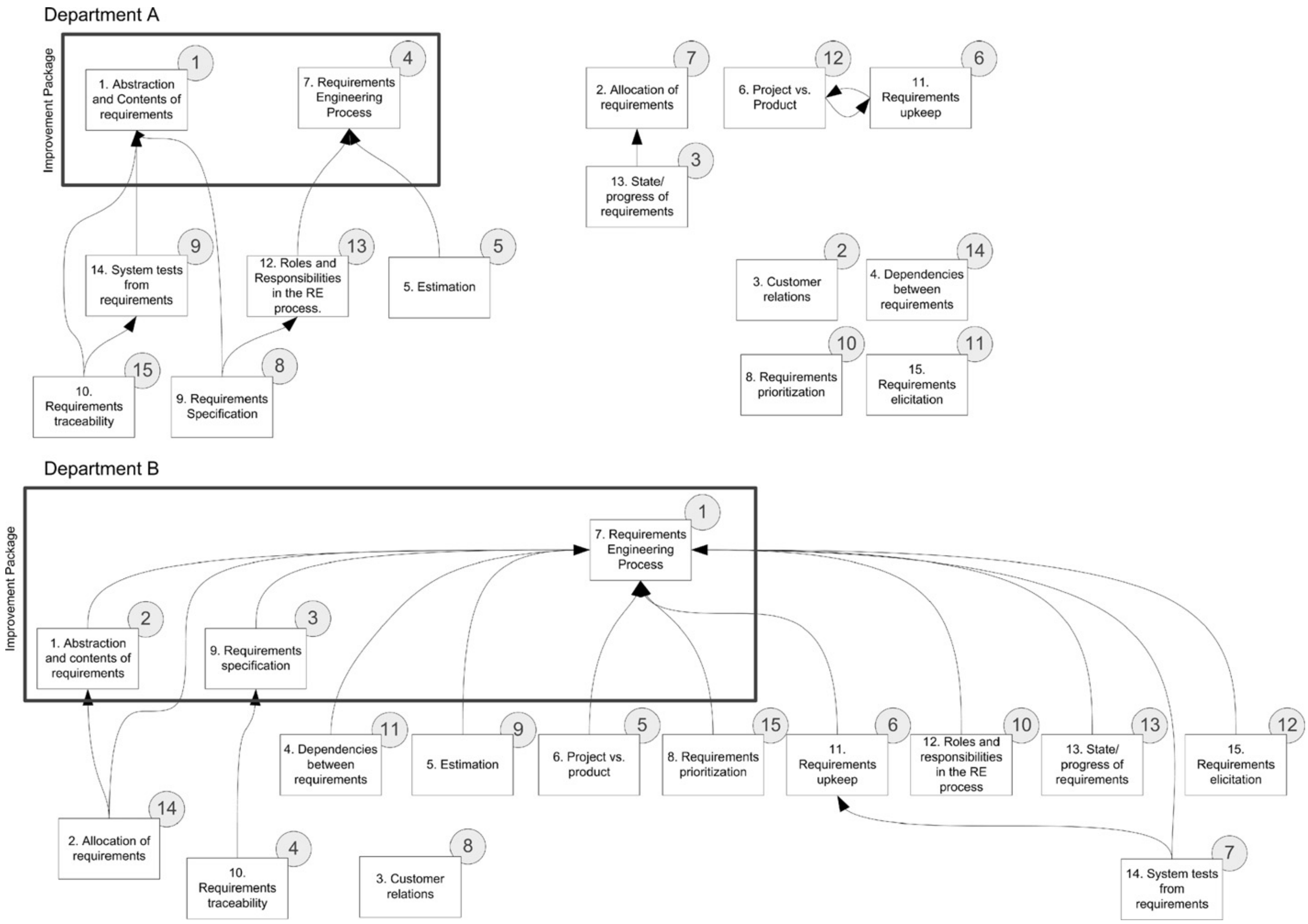

Fig. 12. Resulting improvement packages at departments A and B.

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 Fig. 12.

5.3.4.1. 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. (2000) 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 (Gorschek and Wohlin, 2003, 2004) and nothing has been tailored specifically for these or 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 (Cattaneo et al., 2001), 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 see n in the study done at Danaher Särö (Gorschek and Wohlin, 2004). 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.

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 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 trough
- (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

Ahern, D.M., Clouse, A., Turner, R., 2003. CMMI Distilled: A Practical Introduction to Integrated Process Improvement. Addison-Wesley, Boston MA.

Basili, V.R., 1985. Quantitative Evaluation of Software Methodology. University of Maryland, College Park, Maryland.

Basili, V.R., 1993. The experimental paradigm in software engineering. In: International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions, Lecture Notes in Computer Software, vol. 706, pp. 3–12.

Basili, V., Green, S., 1994. Software process evolution at the SEL. IEEE Software 11, 58–66.

Basili, V.R., McGarry, F.E., Pajerski, R., Zelkowitz, M.V., 2002. Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory. In: Proceedings of the 24th International Conference on Software Engineering (ICSE'02), Orlando, FL.

Beecham, S., Hall, T., Rainer, A., 2003. Software process improvement problems in twelve software companies: an empirical analysis. Empirical Software Engineering 8, 7–42.

Berander, P., Andrews, A., 2005. Requirements prioritization. In: Aurum, A., Wohlin, C. (Eds.), Engineering and Managing Software Requirements. Springer.

Bratthall, L., Joergensen, M., 2002. Can you trust a single data source exploratory software engineering case study? Empirical Software Engineering 7, 9–26.

Briand, L., El Emam, K., Melo, W., 2001. An inductive method for software process improvement: concrete steps and guidelines. In: El Emam, K., Madhavji, N.H. (Eds.), Elements of Software Process Assessment & Improvement. Wiley-IEEE Computer Society.

Brodman, J., Johnson, D., 1995. Return on investment (ROI) from software process improvement as measured by US industry. Software Process: Improvement and Practice 1.

Caffery, F.M., Taylor, P.S., Coleman, G., 2007. Adept: a unified assessment method for small software companies. IEEE Software 24, 24–31.

Calvo-Manzano Villalón, J.A., Cuevas Agustín, G., San Feliu Gilabert, T., De Amescua Seco, A., García Sánchez, L., Pérez Cota, M., 2002. Experiences in the application of software process improvement in SMEs. Software Quality Journal 10, 261–273.

Cattaneo, F., Fuggetta, A., Sciuto, D., 2001. Pursuing coherence in software process assessment and improvement. Software Process improvement and Practice 6, 3–22.

CMMI Product Team, 2006. CMMI for Development Version 1.2, Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA.

Conradi, R., Fuggetta, A., 2002. Improving software process improvement. IEEE Software 19, 92–100.

Damian, D., Chisan, J., Vaidyanathsamy, L., Pal, Y., 2003. An industrial case study of the impact of requirements engineering on downstream development. In: Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03), Los Alamitos, CA.

Electronic Industries Alliance, 1994. EIA Interim Standard: Systems Engineering (EIA/IS-632), Washington, DC.

El Emam, K., Briand, L., 1997. Costs and Benefits of Software Process Improvement. Fraunhofer – Institute for Experimental Software Engineering.

El Emam, K., Drouin, J.-N., Melo, W., 1998. SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. IEEE, Los Alamitos, CA.

El Emam, K., Goldenson, D., McCurley, J., Herbsleb, J., 2001. Modeling the likelihood of software process improvement: an exploratory study. Empirical Software Engineering 6, 207–229.

Fayad, M.E., Laitinen, M., 1997. Process assessment considered wasteful. Communications of the ACM 40, 125–128.

Gorschek, T., 2006. Requirements Engineering Supporting Technical Product Management. Ph.D. Thesis, Department of Systems and Software Engineering, Blekinge Institute of Technology, Ronneby, Sweden.

Gorschek, T., Wohlin, C., 2003. Identification of improvement issues using a lightweight triangulation approach. In: European Software Process Improvement Conference, Graz, Austria.

Gorschek, T., Wohlin, C., 2004. Packaging software process improvement issues – a method and a case study. Software: Practice and Experience 34, 1311–1344.

Gorschek, T., Wohlin, C., 2006. Requirements abstraction model. Requirements Engineering Journal 11, 79–101.

Hayes, W., Zubrow, D., 1995. Moving On Up: Data and Experience Doing CMM-Based Process Improvement. Technical Report CMU/SEI-95-TR-008, Carnegie Mellon, Software Engineering Institute.

Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., Paulk, M., 1997. Software quality and the capability maturity model. Communications of the ACM 40, 30–40.

Herbsleb, J.D., Goldenson, D.R., 1996. A systematic survey of CMM experience and results. In: Proceedings of the 18th International Conference on Software Engineering (ICSE'96), Los Alamitos, CA.

IEEE Computer Society, 2004. SWEBOK, IEEE, Los Alamitos, CA.

ISO, 2005. ISO 9001:2000 – Quality management systems – Requirements.

ISO/IEC, 1995. ISO/IEC 12207:1995 Information Technology – Software life cycle processes.

ISO/IEC, 2002. ISO/IEC 15288:2002 Systems Engineering – System life cycle processes.

ISO/IEC, 2003–2006. ISO/IEC 15504 Information Technology – Process Assessment (Parts 1–5).

Jacobs, S., 1999. Introducing measurable quality requirements: a case study. In: proceedings of the IEEE International Symposium on Requirements Engineering (RE'99), Limerick, Ireland.

Johansen, J., Jenden, K.O., 2003. Did you already know? How good is an organization at identifying its own Strengths and weaknesses? In: Proceedings of the European Software Process Improvement Conference (EuroSPI'2003), Graz, Austria.

Kachigan, S.K., 1986. Statistical Analysis – an Interdisciplinary Introduction to Univariate and Multivariate Methods. Radius Press, New York, NY.

Karlsson, J., Ryan, K., 1997. A cost–value approach for prioritizing requirements. IEEE Software 14, 67–74.

Kitson, D.H., Masters, S.M., 1993. An analysis of SEI software process assessment results: 1987–1991. In: Proceedings of the 15th International Conference on Software Engineering (ICSE'93), Los Alamitos, CA.
Kuilboer, J.P., Ashrafi, N., 2000. Software process and product improvement: an empirical assessment. Information and Software Technology 42, 27–34.
Laryd, A., Orci, T., 2000. Dynamic CMM for small organizations. In: First Argentine Symposium on Software Engineering, Tandil, Argentina.
Miles, M.B., Huberman, A.M., 1994. Qualitative Data Analysis: An Expanded Sourcebook. Sage, Thousand Oaks, CA.
Niazi, M., Wilson, D., Zowghi, D., 2005. A maturity model for the implementation of software process improvement: an empirical study. Journal of Systems and Software 74, 155–172.
Paulk, M.C., 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Reading, MA.
Rainer, A., Hall, T., 2003. A quantitative and qualitative analysis of factors affecting software processes. Journal of Systems and Software 66, 7–21.
Regnell, B., Höst, M., Natt och Dag, J., Beremark, P., Hjelm, T., 2001. An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. Requirements Engineering 6, 51–62.
Reifer, D.J., 2000. The CMMI: it's formidable. Journal of Systems and Software 50, 97–98.
Robson, C., 2002. Real World Research: A Resource for Social Scientists and Practitioner – Researchers. Blackwell Publishing, Cornwall, UK.
Saaty, T.L., Vargas, L.G., 2001. Models, Methods, Concepts & Applications of the Analytic Hierarchy Process. Kluwer Academic Publishers, Boston, MA.
SCAMPI Upgrade Team, 2006. Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A. Version 1.2: Method Definition Document, Carnegie Mellon, Software Engineering Institute, Pittsburgh, PA.
Schneider, K., 2000. Active probes synergy in experience-based process improvement. Springer Lecture Notes in Computer Science 1840, 6–19.
Scott, L., Jeffery, R., Carvalho, L., D'Ambra, J., Rutherford, P., 2001. Practical software process improvement – the IMPACT project. In: proceedings of the 13th Australian Software Engineering Conference (ASWEC'01), pp. 182–189.
SEI, 2007a. Capability Maturity Model for Software (SW-CMM), Carnegie Mellon, Software Engineering Institute. <http://www.sei.cmu.edu/cmm/>.
SEI, 2007b. Integrated Product Development Capability Maturity Model (IPD-CMM), Carnegie Mellon, Software Engineering Institute. <http://www.sei.cmu.edu/cmm/ipd-cmm.html>.
Shewhart, W.A., Deming, W.E., 1986. Statistical Method from the Viewpoint of Quality Control. Dover, New York, NY.
SME TechWeb, 2003. <http://sme.cordis.lu/home/index.cfm>.
SPICE Development Team, 2007. SPICE. <http://www.sqi.gu.edu.au/spice/>.
SPICE User Group, 2006. Automotive SPICE, <http://www.automotivespice.com>.
Synspace, 2005. SPiCE for SPACE (S4S). <http://www.synspace.com/EN/Assessments/s4s.html>.
Wiegers, K.E., Sturzenberger, D.C., 2000. A modular software process mini-assessment method. IEEE Software 17, 62–70.
Wohlin, C., Runeson, P., Höst, M., Ohlson, M.C., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Boston, MA.
Zahran, S., 1998. Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, Reading, MA.

Fredrik Pettersson is a PhD student at the School of Computer Science and Engineering at Chalmers University of Technology, Sweden. He received his MSc in computer engineering, with distinction in software development, from Chalmers in 2004. His research interests include requirements engineering, automotive system development, and software process assessment and improvement.

Martin Ivarsson is a graduate student at the School of Computer Science and Engineering at Chalmers University of Technology, Sweden. His research interests include empirical software engineering and software process improvement. He received an MSc in computer engineering from Chalmers University of Technology in 2004.

Tony Gorschek is a researcher based at Blekinge Institute of Technology, Sweden. His research interests include requirements engineering, technical product management, process assessment and improvement, and quality assurance. He works in close collaboration with industry having Danaher Motion and ABB as primary research partners. He received a PhD in Software Engineering and a Bachelors degree in Business Administration from Blekinge Institute of Technology. He has also worked in industry as a consultant, CTO and project manager for about 6 years. He is a member of the IEEE and the ACM.

Peter Öhman is currently an associate professor in software engineering at the School of Computer Science and Engineering at Chalmers. After receiving his PhD in dependable computer systems in 1995 he worked 10 years at AB Volvo and Volvo Car Corporation in positions such as technical specialist, manager and project leader. His main research interest is empirical software engineering within the automotive domain.