JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS
*J. Softw. Evol. and Proc.* 2014; **26**:251–279
Published online 10 October 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/smr.1637

# A conceptual framework for SPI evaluation

Michael Unterkalmsteiner[1,*,†], Tony Gorschek[1], A. K. M. Moinul Islam[1],
Chow Kian Cheng[2], Rahadian Bayu Permadi[3] and Robert Feldt[1]

[1]*Software Engineering Research Lab, School of Computing, Blekinge Institute of Technology, SE-371 79, Karlskrona, Sweden*
[2]*General Electrics Healthcare, Healthcare IT, 79111, Freiburg, Germany*
[3]*Amadeus SAS, Product Marketing and Development, 06902, Sophia Antipolis Cedex, France*

## SUMMARY

Software Process Improvement (SPI) encompasses the analysis and modification of the processes within software development, aimed at improving key areas that contribute to the organizations' goals. The task of evaluating whether the selected improvement path meets these goals is challenging. On the basis of the results of a systematic literature review on SPI measurement and evaluation practices, we developed a framework (SPI Measurement and Evaluation Framework (SPI-MEF)) that supports the planning and implementation of SPI evaluations. SPI-MEF guides the practitioner in scoping the evaluation, determining measures, and performing the assessment. SPI-MEF does not assume a specific approach to process improvement and can be integrated in existing measurement programs, refocusing the assessment on evaluating the improvement initiative's outcome. Sixteen industry and academic experts evaluated the framework's usability and capability to support practitioners, providing additional insights that were integrated in the application guidelines of the framework. Copyright © 2013 John Wiley & Sons, Ltd.

Received 19 May 2012; Revised 10 April 2013; Accepted 3 September 2013

KEY WORDS: Software Process Improvement; software measurement; software process evaluation

## 1. INTRODUCTION

With the increased importance of software in product development [1], the software engineering discipline and the study of the involved processes have started to gain more popularity among researchers and practitioners in industry [2–4]. Software Process Improvement (SPI) encompasses the assessment and improvement of the processes and practices involved in software development [5]. SPI involves the understanding of the software processes as they are used within an organization and suggests areas for improvements in achieving specific goals such as increasing product quality, operation efficiency, and cost reduction [6]. The SPI literature provides many case studies of successful companies and descriptions of their SPI programs [7]. Examples are presented by [8–16] and also cover the recently popular development practices classified as agile or lean [17, 18].

Assessing the outcomes of SPI initiatives is as important as their actual implementation because without a clear understanding of gains or losses, it is impossible to reason about the performance of an SPI initiative [19]. Measurement in SPI can be of descriptive, evaluative, or predictive nature [20]. Descriptive and predictive measurement is the primary facility to enable the software process to perform

*Correspondence to: Michael Unterkalmsteiner, Software Engineering Research Lab, School of Computing, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden.
†E-mail: mun@bth.se
This article was published online on [11 October 2013]. The affiliation of the third author was subsequently changed. This notice is included in the online and print versions to indicate that both have been corrected [18 October 2013].

with predictable performance and capability and to ensure that process artifacts meet their requirements [21, 22]. Evaluative measurement aims at providing support for operative decisions [20]. In this paper, we focus on the evaluative nature of measurement, targeted at assessing the impact of SPI initiatives.

The success of improvement initiatives also means different things to different people [23]. Hence, various stakeholders' points of view have to be taken into consideration when assessing the outcome on an SPI program [24]. The causal relationship between the improvement initiative and its effect is complex, and it is hard to determine whether the effect being measured is stemming exclusively from the improvement initiative [25].

The lack of guidelines for conducting evaluative SPI measurements has raised the challenge to develop and to implement effective performance measurement programs for SPI [26]. Because the evaluation of the outcome of an SPI initiative is not only complex but also crucial to the organization, there is a need for a measurement and evaluation framework that guides SPI practitioners in their work, helps in preserving effort and cost, and enables return on investment to be ascertained. Such a framework should promote an evaluation that considers the improvement from different views, increase the visibility, and consequentially facilitate the assessment of the achieved benefits.

The challenges in process improvement evaluation are diverse, ranging from defining an appropriate measurement scope, eliciting the required metrics, to the consideration of confounding factors in the evaluation [27]. This paper proposes a conceptual framework that aims to address these challenges, offering a structured approach. The framework was derived from an extensive systematic literature review [27], which collected best practices in the field. Subsequently, we followed the technology transfer model proposed by Gorschek *et al*. [28] to statically validate the framework by experts from both academia and industry.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. In Section 3, we present four major challenges, basing their formulation on the results of a systematic literature review on SPI measurement and evaluation [27]. With the aim to address those challenges, we developed the SPI Measurement and Evaluation Framework (SPI-MEF). Section 4 describes the framework, and an example scenario is provided in [29]. The usefulness and usability of SPI-MEF was validated through the help of nine research experts and seven industry practitioners as described in Section 5. The results and the refinements applied to SPI-MEF are discussed in Section 5.3. Threats to validity are discussed in Section 5.2. Finally, conclusions and motivations for future work are given in Section 6.

## 2. RELATED WORK

In this section, we briefly review previous work relevant to the measurement and evaluation framework proposed in this paper.

Software process appraisal methods, for example, SCAMPI [30], or guides to process assessment, for example, ISO/IEC 15504 (Part 4) [31], evaluate whether an organization conforms to a certain industry standard. The assessment identifies areas for improvement [32] and can steer the implementation of process improvements [33]. Such assessments provide a benchmark against a set of goals, however do not evaluate the actual impact of process changes.

Software Process Improvement research into measurement programs has developed and suggested several metrics [34]. For example, the *Assess, Analyze, Metricate, Improve* (ami) approach integrates an analytic, bottom-up with a benchmarking, top-down approach to process improvement [35, 36]. The rationale for the expected synergy is that top-down approaches, such as the Capability Maturity Model (CMM) [37], do not consider the specific business goals of a company [38]. Hence, the proposed goal-oriented measurement in ami, on the basis of the Goal Question Metric (GQM) paradigm [39, 40], serves to analyze the identified improvement opportunities more in depth and to monitor the implemented changes, assuring that the followed best practices lead to the achievement of the targeted business goals. Similarly, the GQM + strategies approach aims at linking business strategies with measurement goals (MGs) [41], because the Capability Maturity Model Integration [42] does not provide an explicit link from the improvement to business value [43].

Inspired by the ami approach, Park *et al*. [44] developed the GQ(I)M method. Extending the GQM paradigm, GQ(I)M introduces the notion of *indicators*, which reflect the idea of asking 'What do I want to know?' as opposed to the question 'What do I want to measure?'. Indicators are therefore representation of measurement data, backed by one or more metrics, and support with a clear definition of their construction the decision making process [45].

The integration of process assessment, modeling, and measurement is the goal of the product-focused improvement approach [46]. As opposed to ami and GQ(I)M, in which *company-specific* business goals define the measurement strategy, product-focused improvement approach promotes continuous assessment against *reference models* such as CMM or ISO/IEC 15504, supported by measurements derived by GQM [47, 48]. The expected benefits are higher visibility of process changes and therefore better control on the improvement process and lower assessment costs, as the time needed for data collection is reduced [47].

An alternative to the ubiquitous GQM paradigm is the Practical Software and System Measurement (PSM) approach [49] that influenced and was influenced [50] by the in-parallel developed international standard for Software Process Measurement, ISO/IEC 15939 [51]. In contrast to the more general, goal-oriented GQM, PSM is designed to establish a measurement process for project evaluation, following the plan–do–check–act cycle [50]. Measurement in PSM has the purpose to satisfy the project manager's information needs that stem from the following: (i) the achievement of project success; and (ii) obstacles or issues related to achieving success [52].

We reviewed the literature on SPI evaluations conducted in industry [27], identifying current practices that also were built upon, or were inspired by the approaches discussed in this section. The analysis of these practices led to the definition of several challenges related to the evaluation of SPI initiatives, which are discussed in further detail in Section 3.

## 3. CHALLENGES IN MEASURING AND EVALUATING SOFTWARE PROCESS IMPROVEMENT INITIATIVES

Obstacles and issues in implementing measurement programs in general were previously identified by Herbsleb and Grinter [53] (difficult communication across organizational boundaries, rigidity of data collection mechanisms, and nontransparent data usage), Berry and Ross [54] (the complexity of combining sociological and technological aspects in a measurement program), Kasunic *et al.* [55] (poor data quality), and Umarji and Seaman [56] (different perceptions of metrics between developers and managers).

Because the focus of these issues is predominantly on the implementation of measurement programs, which is a critical but not the sole aspect of SPI evaluation, we devised four fundamental challenges in measuring and evaluating SPI initiatives. The formulation of these challenges bases upon the findings from a systematic literature review on measurement and evaluation of SPI [27].

Sections 3.1–3.4 characterize the identified challenges and explain how they are addressed by the six concepts presented in SPI-MEF.

### *3.1. Challenge I – heterogeneity of Software Process Improvement initiatives*

The spectrum of SPI initiatives ranges from the application of tools for improving specific development processes, to the implementation of organization-wide programs to increase the software development capability as a whole [27]. As a consequence of this variety and diversity in scope and complexity of SPI initiatives, we designed SPI-MEF as a set of interrelated concepts, each one addressing one or more challenges. Figure 1 summarizes the relationships between challenges and concepts.

In each concept, we provide a set of practices that can be used to fulfill the goals of the concept and hence addressing the challenge. These practices however may need to be adapted and scaled to the specific context in which the framework is used. Hence, the first concept is termed *gap analysis of evaluation quality*. It provides means to assess the current and to define the aspired evaluation quality, enabling the customization and scaling of the framework to different types of SPI initiatives.

### *3.2. Challenge II – partial evaluation*

The outcome of SPI initiatives is predominately assessed by evaluating measures that are collected at the project level [27, 57]. As a consequence, the improvement can be evaluated only partially, neglecting

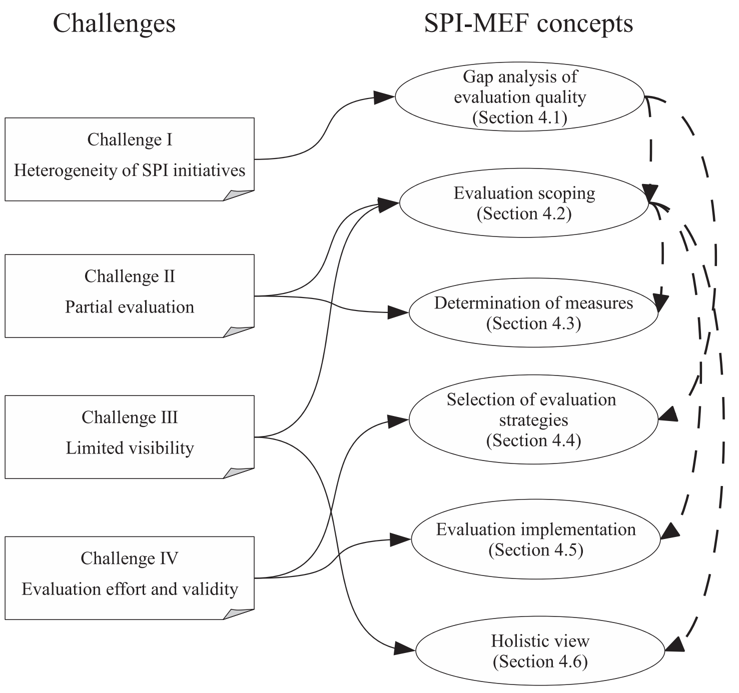

Challenges
SPI-MEF concepts
Challenge I
Heterogeneity of SPI initiatives
Challenge II
Partial evaluation
Challenge III
Limited visibility
Challenge IV
Evaluation effort and validity
Gap analysis of evaluation quality (Section 4.1)
Evaluation scoping (Section 4.2)
Determination of measures (Section 4.3)
Selection of evaluation strategies (Section 4.4)
Evaluation implementation (Section 4.5)
Holistic view (Section 4.6)

Figure 1. Conceptual map of the framework.

effects that are visible only outside individual projects. Such evaluations can therefore lead to suboptimizations of the process [53]. By focusing on the measurement of a single attribute, for example, effectiveness of the code review process, other attributes might inadvertently change, for example, time-to-market of a product.

To address this challenge, we propose the concept of *evaluation scoping*, which provides means to determine the extent of the improvement evaluation. This concept provides the answer to the question: where to measure? Complementary, we also propose the concept of *determination of measures*, which aims at providing the answer to the question: what to measure?

## *3.3. Challenge III – limited visibility*

This challenge is a consequence of the previous one because a partial evaluation implies that the gathered information is targeted to a specific audience that may not cover all important stakeholders of an SPI initiative. This means that information requirements may not be satisfied and that the actual achievements of the SPI initiative may not be visible to some stakeholder as the measurement scope [27] is not adequately determined.

Evaluation scoping aims to address this issue by providing a structured approach to identify the relevant stakeholders and to provide them with the information they need. The concept of *holistic view*, on the other hand, provides a way to collect and to present the gathered information, supporting a multifaceted view on the improvement initiative.

## *3.4. Challenge IV – evaluation effort and validity*

Because of the vast diversity of SPI initiatives (see challenge I), it is not surprising that the evaluation strategies vary. The evaluation and analysis techniques are customized to the specific settings were the initiatives are embedded [27]. Because there exist no formal guidelines for implementing an SPI evaluation [26], one can assume that the design and development of the evaluation strategies require a considerable amount of effort. Furthermore, confounding factors are seldom taken into account in the industrial practice of improvement evaluation [27]. This can be a major threat to the evaluation validity because the predominant practice of improvement evaluation is based on pre-post comparison [27].

To address this challenge, the concept *selection of evaluation strategies* provides support in identifying and implementing adequate means for SPI evaluation. In addition, the concept *evaluation implementation* discusses timing factors that should be considered and provides support for conducting the evaluation itself.

### *3.5. Summary*

Figure 1 shows a conceptual map, indicating how challenge I–IV are addressed by the SPI-MEF concepts and how the concepts are related to each other. Gap analysis of evaluation quality, whose intent is to tune the overall measurement and evaluation approach, is directly connected to evaluation scoping and selection of evaluation strategies and indirectly to determination of measures. Similarly, evaluation scoping, whose intent is to define where to measure and who will see the results, influences the evaluation implementation and the holistic view concepts. In Section 4, we describe all SPI-MEF concepts in detail and present practices used to realize the concept in practice.

## 4. SOFTWARE PROCESS IMPROVEMENT MEASUREMENT AND EVALUATION FRAMEWORK

Software Process Improvement Measurement and Evaluation Framework (SPI-MEF), developed on the basis of an extensive study of SPI research and industry case studies [27], not only maps best practices but also gaps in knowledge. The reviewed primary studies provided an excellent source of practices applied successfully in industry but, even more importantly, allowed us to extract generic guidelines to support the evaluation of SPI initiatives.

This section presents these guidelines, complemented with nine interconnected, although fictitious, examples (starting with example box 1), which in essence constitute SPI-MEF. As the relationships shown in Figure 1 suggest, there exist dependencies between the concepts. They are however ordered in a way, reflected in the structure of the guidelines and the examples, in which one would typically conduct the planning for the evaluation. The extended scenario provided in [29] shows how the framework is applied in an iterative manner, following a phased approach.

### *4.1. Gap analysis of evaluation quality*

This concept, addressing Challenge I - heterogeneity of SPI initiatives, reflects the need to capture the context in which SPI initiatives are implemented. The particular context characteristics steer further decisions, for example, in evaluation scoping (Section 4.2) and selection of evaluation strategies (Section 4.4). The basic information that should be recorded is as follows (refer also to example box 2):

(a) A description of the initiative and its purpose.
(b) Concrete improvement goals.
(c) The affected process areas.
(d) The target entities of the initiative, that is, specific projects, products, or departments.
(e) A tentative schedule for the implementation.

Example box 1: introduction

The scenario is embedded within the context of a medium-sized software development organization called ALPHA. ALPHA has 70 full-time software professionals developing off-the-shelf software applications for various industries. Software development in the company follows an iterative spiral life-cycle model, and a typical software product is released after 6 months.
ALPHA develops products on top of a software platform. ProductA has just been released, and incremental feature releases are scheduled biyearly. Because of quality assurance issues at the end of the development life-cycle of ProductA, it was decided to introduce code inspections in the projects for the feature releases. As planning for ProductB has commenced and new development staff was employed, code inspections were also deemed to be an effective means to introduce new developers to the software platform.

Furthermore, the organization's capability to implement a measurement program and to conduct an evaluation needs to be assessed by a gap analysis. In general, gap analysis uses two sets of information: the current status and the aspired status [58]. The current status of measurement capability can be determined by following one of the maturity assessment approaches presented by Daskalantonakis *et al*. [59], Comer and Chard [60], Niessing and Vliet [61], and Diaz-Ley *et al*. [62]. Then, the aspired measurement and evaluation quality have to be determined. The subsequently identified gap then shows what refinements are needed in the measurement program.

SPI-MEF proposes to use a '2 × 2' matrix to support the decision process, using the context information and current measurement capability as input (Figure 2). The evaluation quality in SPI-MEF is defined by two dimensions: accuracy and coverage. Accuracy can be improved by considering primary and complementary measures (Section 4.3.1), selecting the appropriate evaluation strategy, and by taking confounding factors into account (Section 4.4). Coverage is determined by to what extent measurement levels (MLs) and viewpoints (Section 4.2) are included in the evaluation. It is possible to address both dimensions simultaneously, but cost and effort constraints may prohibit such a strategy. The 2 × 2 matrix has been proven to be an excellent tool to address such decision dilemmas [63]. The roles involved in the discussion about the long-term strategy should include management that provides the funding for the improvement initiative and measurement program experts. The Evaluation Opportunity Matrix serves as a starting point for the discussion (Figure 2 and Example box 3).

Example box 2: initiative context

The current software development processes in ALPHA offers several opportunities for improvement. A process team has been set up to investigate the process areas that need attention, and they proposed to introduce code inspections. The initiative's context is documented in the succeeding texts:

| Type of SPI initiative | Practice (code inspections) |
|---|---|
| Description | Code inspection is a systematic process in reviewing one developer's work product in the coding phase. The code inspection follows a structured process that involves the planning, preparation, inspection meeting, rework, and follow-up. The code written by the developer will be inspected by two or more peers, usually more experienced or senior developers, in the project team. |
| Improvement goal(s) | Improve product quality |
| Process areas/phases | Coding phase |
| Target entities | All development projects |
| Implementation schedule | Phase I: pilot projects<br>Phase II: all projects in the organization |

Aside from accuracy and coverage another important aspect to consider is the cost of the evaluation. Cost is denoted as a function of the quality and scope of the evaluation. Therefore, the resources an organization is willing to invest have to be taken into consideration when choosing the desired path to improve the quality of evaluation. The cost of evaluation can arise from the amount of metrics defined and collected, the resources needed to manage the metrics, the number of people involved in the metric collection and evaluation, and so on. Although achieving high accuracy and coverage in the context of SPI-MEF seems to inherently require more metrics, the potential reuse of metrics should be considered when evaluating the cost for evaluation implementation. During the decision process of which strategy to follow, it is advisable to involve personnel who have expertise in implementing a measurement program and can estimate the cost of metric collection and management, which sponsors the improvement program.

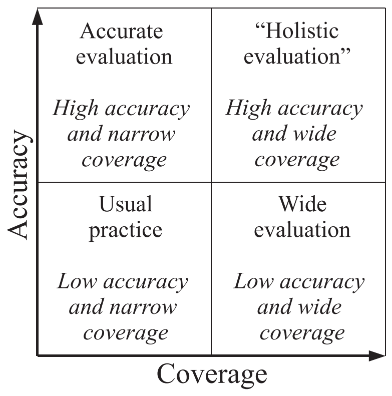
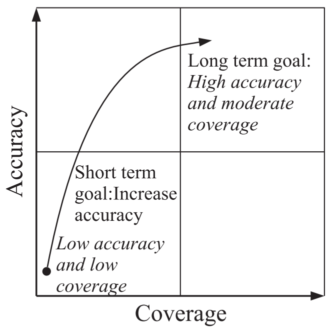

Accurate evaluation
High accuracy and narrow coverage
"Holistic evaluation"
High accuracy and wide coverage
Usual practice
Low accuracy and narrow coverage
Wide evaluation
Low accuracy and wide coverage
Accuracy
Coverage

Figure 2. Opportunity matrix.

Example box 3: evaluation opportunity matrix

The Software Engineering Process Group (SEPG) at ALPHA met with the upper-level management and employees who are currently in charge of the measurement program. To increase accuracy in the short term, plans are to consider primary and complementary measures and to control the major confounding factors by establishing a baseline from the appropriate historical data. Considering the process improvement budget and the implementation schedule, the company's short-term goal is to focus on process-level and project-level accuracy first and then address coverage by including the product ML.


Long term goal: High accuracy and moderate coverage
Short term goal:Increase accuracy
Low accuracy and low coverage
Accuracy
Coverage

## 4.2. *Evaluation scoping*

The evaluation scope is determined using two dimensions. The measurement levels (MLs) represent the spectrum of measurable entities that can potentially be assessed in the evaluation. Identifying the MLs for evaluation counteracts Challenge II - partial evaluation as it leads the practitioner to consciously define the coverage of the evaluation. Section 4.2.1 explains the MLs in more detail.

The second dimension, evaluation viewpoints (EVs), represents the stakeholders and their information needs in relation to the evaluation of the improvement. Defining EVs counteracts Challenge III - limited visibility as it clarifies the stakeholders' data requirements to evaluate the improvement initiative. Section 4.2.3 discusses the EVs in more detail.

*4.2.1. Measurement levels.* The MLs represent the spectrum of entities that are affected by SPI initiatives and need to be measured in order to achieve a holistic evaluation of the SPI initiative's outcome. The MLs are inspired by the levels of dependent variables proposed by Gorschek and Davis [57].

On the *process* level, the efficiency and effectiveness of the implemented process improvement initiative can be assessed. For example, if the process change consists of involving testers in requirements reviews, it can be measured how many faults are identified compared with the previous instance of the process. A measurable gain at this level is however not sufficient to assert that the improvement goal (e.g., improved product quality) has been reached. Furthermore, the improvement of one process may produce side effects on other processes, or, more generally, affect the output of the process, which is not measurable at this level.

The measurement at the *project* level is mainly concerned with project control by monitoring budget, schedule, and resources. A projects' success or failure is often evaluated by determining the discrepancy between estimated and actual values. Additionally, it is possible to measure the effects of newly introduced or modified processes by assessing the work products created during the project. For example, if the requirements review process leads to fewer specification changes during the project life cycle. Adherence to project estimates not only can indicate process improvement but also can be misleading when considered in isolation as product quality is not assessed. Linberg [64] reports on a case study in which a project faced severe schedule and budget overruns. From the management's point of view, the project was perceived as a failure, whereas the product, once shipped, was highly successful. Thus, considering the project *and* the product perspective in an improvement evaluation is important.

Increasing product quality is often the major improvement goal when establishing an SPI initiative. Measurement at the *product* level assesses both internal quality attributes, which are mostly visible to software developers, and external quality attributes, which are observed by the user of the product. Besides increased quality, process improvement may also target a reduction in cost and time-to-market of the product. Continuing on the previous example with requirements reviews, the involvement of testers could lead to a delay in other projects, to which they were originally assigned. The project with the improved review process and tester involvement could be completed earlier because of less rework; other projects however could be delayed because of the deduction of resources. It is therefore necessary to control and to assess all aspects of the improvement goals and to take them into consideration when evaluating the initiative's success.

The short-term and mid-term effects of an SPI initiative can be assessed in the process, project, and product level, but the long-term effects will prevail and only be visible at the *organization* level. An SPI initiative has to meet the business goals of a company and has to be aligned with its vision. Therefore, it is necessary to assess the improvements impact on the organization's business strategy, economy, and culture. The example with the involvement of testers in requirements reviews shows that a reorganization of the development process may be required in order to avoid resource and scheduling issues. Hence, the measurement and evaluation of the performance of the SPI initiative at the organization level is of importance for the design and implementation of forthcoming process improvements.

The previously mentioned MLs are focused on the measurement and evaluation of the SPI initiative within the company and neglect that the effect of the improvement may also transcend the organizational border to the exterior world. The *external* level is not only influenced by the produced goods but also by the organization itself, for example, through supplier dependencies. For example, the aforementioned improvement of the development process can also affect suppliers as they may need to interact differently with their client. Measurement at the external level assesses positive and negative externalities, which should be taken into consideration when evaluating the success of an SPI initiative.

*4.2.2. Effect traceability in measurement levels.* Effect traceability is an issue inherent in the MLs that was also brought up by Gorschek and Davis [57]. The traceability between the action and its empirically assessable effects diminishes with increasing distance from the process change.

Because of temporal distance, there is an increasing latency by which the effect of process improvement is measurable at the different levels. That is, the effect of the treatment will reach the process itself first, then the projects in which the process is applied, and eventually the products emerging from the various projects. Furthermore, the ability to isolate the effect on a particular improvement decreases. Each level is an aggregation of one or more entities of the previous level, for example, the external level includes, besides other things, a set of organizations, which in turn, include, besides other things, a set of products. These ‘other things’ can be seen as external variables and are defined in this context as confounding factors. Those may aggravate an accurate evaluation because they hide or amplify the effects of the improvement initiative.

In order to counteract these effect traceability issues, we propose EVs as the second dimension for evaluation scoping.

*4.2.3. Evaluation viewpoints.* According to Zahran [65], a SPI initiative has to be backed up by both organizational and management infrastructure, as well as a process technical infrastructure. The organizational and management infrastructure defines the stakeholders, which are usually involved in the improvement initiative, such as executive sponsors, a steering committee, a SEPG, and SPI teams. Besides the viewpoints represented by the previously mentioned SPI stakeholders, the evaluation should also consider the viewpoints from top-management and middle- management, product and project management, and software developers which are not directly in charge of the improvement initiative. Daskalantonakis [66] identified six target audiences for the evaluation and use of metrics in software organizations: software users, senior managers, software managers, software engineers, and software process engineers and software quality assurance. Similarly, Ebert [67] identified four roles with individual goals related to the improvement: practitioners, project managers, department head, and corporate executives.

The specific roles encountered in an organization and in an SPI initiative are highly dependent on the structure of the organization and the extent of the process improvement initiative. Hence, we generalize the potential stakeholders into three EVs: implementer, coordinator, and sponsor. The three EVs reflect the different angles from which the process improvement is perceived and, more importantly, which aspects of the improvement matter to whom when conducting the evaluation. The definition of different viewpoints also supports the idea of increasing the visibility of the process improvement, which is, presenting the information to the appropriate stakeholders and alleviate the decision-making process (example box 4). It is important to point out that for a holistic evaluation of the improvement initiative, it is necessary to consider and to account for all viewpoints and the respective evaluation results without isolating single aspects [24].

Example box 4: evaluation scoping

In example box 3, the SEPG decided to evaluate the improvement initiative, code inspections, on the process, project, and product levels. The SEPG defines the EVs for the respective levels. For illustrative purposes, the table in this example contains also the organization and external levels. The first column denotes the MLs, that is, the entities that are affected by the SPI initiative, whereas the remaining columns denote the EVs, that is, the stakeholders that have an interest in the evaluation of the SPI initiative. The table is read, for example, as follows: ‘The development team is interested in evaluating the impact of the SPI initiative on the process ML from the implementer EV’. Note that a specific role can have different interests when evaluating an improvement initiative and therefore represent more than one viewpoint. By looking at the table, the ‘product manager’ role subsumes both the coordinator viewpoint at the product level and the implementer viewpoint at the organization level.

| | EVs | | |
|---|---|---|---|
| MLs | Implementer | Coordinator | Sponsor |
| Process | Development team | SEPG | SPI steering committee |
| Project | Development team | Project manager/ SEPG | SPI steering committee/head of department |
| Product | Development team/ project managers | Product manager | Head of department |
| Organization | Product manager | Board of directors | Company shareholders/ customers |
| External | Product department | Board of directors | Product user/regulator |

The rationale for the mapping in the table is given by considering both the characteristics of MLs and EVs. The implementer viewpoint requires feedback in a short-term and mid-term time-frame of the improvement. The development team and project managers, who are responsible to put the code reviews into practice, represent therefore the implementer viewpoint. The coordination and control of the improvement activities is the coordinator’s viewpoint concern. In the case of ALPHA, the SEPG and the project/product manager need to know if the initiative was efficiently implemented and can be expanded to all projects within the company. The sponsor viewpoint, on the other hand, needs confirmation that the improvement benefits the organization in the midterm to long term. The SPI steering committee is interested in identifying conflicts/inefficiencies in the changed process, whereas the head of department needs to assess the financial payoff.

The *implementer* viewpoint represents all the roles that are dedicated to put the software development in general, and the process improvement in particular, into practice. The evaluation from this viewpoint is needed to make the effects of changes in behavior visible to the enactors of the process improvement. The rationale behind this argument is that a feedback loop on the effects of the improvement fosters the sustainment of process change. Additionally, if the implementer is well informed about the improvement and is conscious of its effects, he can serve as an accurate data source for the evaluation of the improvement [68], as well as be an active contributor to the improvement [69, 70].

The *coordinator* viewpoint comprises the roles that generally participate in software development and in a SPI initiative as coordination and control entities. Their areas of responsibility include managing and leading the implementers, and steering and promoting the process improvement through strategic (higher level, global), and tactical (lower level, local), decisions. The interests in evaluating the improvement initiative from this viewpoint are several, but in general, they boil down to two aspects: (i) to assess if the improvement goals have been achieved and use the output of the evaluation to drive and to guide further improvement activities; and (ii) to provide feedback to superiors.

The *sponsor* viewpoint represents those roles that fund and motivate the improvement initiative, and in parallel, those who are interested in evaluating the improvement according to its costs and benefits. The motivating roles' focus is towards the evaluation of the improvement process itself in order to assess if it delivered the anticipated benefits. This includes, for example, the SPI steering committee and/or the head of department. On the other hand, the evaluating roles' focus may be less on the improvement process itself and rather on the results that are visible in the environment in which the process change is embedded. This includes, for example, not only higher-level management financing the effort but also company-external entities such as shareholders, customers, or regulatory stakeholder. In either case, the evaluation needs to be able to confirm the long-term effects of the process improvement.

### *4.3. Determination of measures*

In order to perform an accurate evaluation, measurements need to provide the required data. The question is how to elicit the set of sufficient measurements that allow an evaluation to express reliably if an improvement goal has been reached or not. The common approach is to derive the required metrics from the improvement goal. For example, if a reduction in cost is targeted, one could measure and evaluate if the expended resources in a project that implements the process improvement were reduced as compared with a previous, similar, project.

There are two problems with this approach:

1. Not all the benefits in terms of cost reduction may be visible at the project level, that is, assessing on this level alone would only show a subset of the achieved benefits.
2. If the expenditure of resources in a project is the only assessed dependent variable, it is not possible to evaluate if the improvement did provoke any side effects. In particular, detrimental influences that are visible only with some delay, and on different MLs, would not be accounted for in an evaluation on the basis of pure project-level measurements. Further, using only a single metric as an achievement indicator could raise validity concerns in the subsequent evaluations [71]. One reason for this could be data collection issues, for example, incomplete data sets or incorrectly compiled data forms.

To address problem 1, one has to reason about selecting the appropriate target audience for the evaluation and then deriving the necessary measurements. This evaluation scoping was discussed in Section 4.2 were we introduced MLs and EVs as scoping instruments.

To address problem 2, we propose a method that builds upon the GQM paradigm [39, 40]. GQM is a systematic way to tailor and to integrate organizations' objectives into measurement goals (MGs) and refine them into measurable values. It provides a template for defining MGs and guidelines for top-down refinement of MGs into questions and then into metrics, and a bottom-up analysis and interpretation of the collected data [40]. SPI-MEF provides an interface with the conceptual level of GQM to define the appropriate MGs for improvement evaluation as illustrated in Figure 3.

We tailor the GQM approach to the context of SPI measurement and evaluation. The rationale for interfacing the GQM facets with SPI-MEF, illustrated also in example box 5, is as follows. The 'object of study' facet corresponds to the implemented SPI initiative, whereas the 'purpose' is to evaluate the impact of the change. The 'focus' corresponds to the notion of success indicator (SI), which is determined by the ML and the consideration of primary and complementary indicators. Table I lists SIs that are commonly encountered in SPI initiatives. Section 4.3.1 discusses primary and complementary indicators in further detail. The 'point of view' corresponds to the concept of EV.

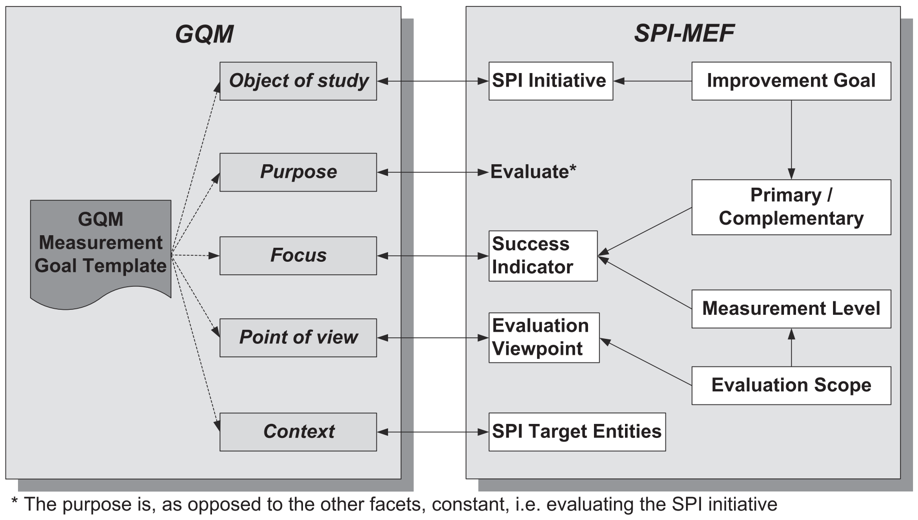


Figure 3. Software Process Improvement Measurement and Evaluation Framework (SPI-MEF) interface with Goal Question Metric (GQM).

Table I. Measurement levels and success indicators (on the basis of [27]).

| Measurement level | Success indicator | What is measured? |
|---|---|---|
| Process | Efficiency | The means of the process implementation. |
| | Effectiveness | The ends of the process implementation, visible in any work product and/or artifact. |
| Project | Defects | Artifact quality w.r.t. the different phases in the project life cycle. |
| | Cost | Investment in terms of resources and effort to conduct the implementation of project. |
| | Schedule | Calendar time of project and/or phases therein. |
| | Productivity | Effort input and size output in project activities. |
| | Estimation accuracy | Difference between planned and actual outcomes of project success indicators. |
| Product | Quality | Internal and external quality attributes of the software product. |
| | Cost | Total cost of product development and maintenance. |
| | Time to market | Calendar time between product inception and delivery. |
| Organization | Economics | Costs and benefits (including intangible assets). |
| | Employees | Employee satisfaction. |
| | Growth | Organizational growth, revenue, and innovation. |
| | Communication | Collaboration and communication between employees and/or customers. |
| External | Customer externalities | Any of the preceding texts, applied however to the customers' context. |
| | Society externalities | Effects on the environment of the organization. |

The 'context' facet corresponds to the SPI target entities, which set, as defined in Section 4.1, the scope of the SPI implementation.

*4.3.1. Primary and complementary indicators.* We define primary indicators as the set of measurements that are used to assess if the improvement goal has been reached. For example, given that the improvement goal is cost reduction, primary measurements could be elicited from the process (e.g., efficiency of the changed/added process) and project level (e.g., effort in man hours).

Complementary indicators capture the effects of process improvement that are not directly connected with the expected effect of the initiative. In other words, complementary indicators assess the side effects that may arise when, through an improvement initiative, the corresponding primary indicator is affected (Table II). This is needed in order to control measurement dysfunction that may arise from either wrongly

Table II. Identifying complementary indicators.

| | Primary | Complementary | |
|---|---|---|---|
| Success indicator (project level) | Cost | Quality | Schedule |
| Example metric | Effort in man hours | Defect density | Project cycle time |

reported data or from suboptimization of primary indicators. Iversen and Mathiassen [72] reported on a case where the measurement program was threatened because of mistrust in the collected data.

Example box 5: determination of measures

On the basis of the scoping decisions made in example box 4, ALPHA needs to identify SIs for the Process (PRC) and Project (PRJ) MLs (we omitted the product level because of space limitations). Representatives from the three EVs meet to elicit SIs for the improvement initiative (code inspections). The implementer (I) is represented by one role from the development team and by one of the project managers; the coordinator (C) by one representative from the SEPG and one product manager; and the sponsor (S) is represented by one member of the SPI steering committee and the head of the department. The SIs can either be primary (p) or complementary (c) from the viewpoint of the evaluator.

| ML | EV | SI | MG |
|---|---|---|---|
| PRC | Development team (I) | Effectiveness (p) | MG01 |
| | SEPG (C) | Effectiveness (p) | MG02 |
| | | Efficiency (c) | MG03 |
| | SPI steering committee (S) | Effectiveness (p) | MG04 |
| PRJ | Development team (I) | Defects (p) | MG05 |
| | Project manager/ SEPG (C) | Defects (p) | MG06 |
| | | Cost (c) | MG07 |
| | | Productivity (c) | MG08 |
| | SPI steering committee/ Head of department (S) | Defects (p) | MG09 |
| | | Cost (c) | MG10 |

The aforementioned table acts as a bridge that pulls together the elements from SPI-MEF to produce an instance of GQM for the purpose of evaluating an SPI initiative. The aim of the table is thereby to provide a structure for the forthcoming derivation of MG. The proposed procedure creates, in general, a high number of MGs (10 in this example). They can be grouped according to their respective EVs, that is, MG01 and MG05 belong to the development team (I), MG02 and MG07 belong to the SEPG (C), etc. MG01 can be formulated as: *Evaluate the process of code inspections with respects to its effectiveness on pretest defect detection in the pilot project from the viewpoint of the development team*. The following application of the GQM method (for example, van Solingen and Berghout [73] for detailed instructions) benefits from the concrete specification of MGs.For the identified measures at the process level, ALPHA could not define any baseline as code inspections were introduced for the first time in the pilot projects (phase I of the improvement initiative). The collected measures are however used to establish a baseline against which phase II is evaluated (example box 2 where the initiative context was defined). Project-level measures, on the other hand, are available from two previous projects. One project manager of the pilot projects and one expert from the SEPG were nominated as metric evaluators (example box 8).

The method we propose to identify complementary indicators borrows its central idea from the project management triangle [74, 75] whose respective edges represent cost, time, and quality. The aim of the project management triangle is to create the awareness that the entities at the edges are interrelated with each other and changing one will inevitably affect the others. Considering this principle in the context of process improvement evaluation, helps to identify complementary indicators. Table II exemplifies this idea where cost is a primary SI and quality of the produced artifacts and project schedule are complementary indicators.

Three basic SIs, cost, time, and quality, can be used as a starting point, because those are the commonly targeted improvement goals, for example, by Basili *et al.* [76], Debou and Kuntzmann-Combelles [38], Murugappan and Keeni [77], Weiss *et al.* [78], and Moreau *et al.* [79]. In [29], we provide an initial set of the SIs shown in Table I that can be refined, depending on the actual improvement goal(s) and the concrete context in which the initiative is conducted.

*4.3.2. Metrics baselining.* The SIs and their respective metrics need to be baselined in order to serve as the initial point for evaluating the improvement. There are various ways how organizations can set the baselines for their metrics. The most typical way would be creating the baseline from historical data collected from previously conducted processes or projects and already finished products. Some derived metrics that consist of two or more elementary measurements can sometimes be easily acquired from historical data. If there is no historical data available, the baseline can be obtained by collection of data from active projects that are currently running in the organization. The data collected from the active projects would serve as the baseline to evaluate the projects that are going to incorporate the SPI initiative.

With the definition of the baseline, an expert in interpreting the metric needs to specify the ranges indicating improvement, stagnation, and decline of that metric. The same evaluator assesses, later on in the evaluation (Section 4.5.2), the change of the metrics' value with respect to the defined baseline.

### *4.4. Selection of evaluation strategies*

Software Process Improvement evaluation strategies can be classified into four general categories [27], that are, in practice, often applied in combination: basic comparison, statistics-based analysis, survey, and cost-benefit analysis. The fundamental idea of the basic comparison strategy is to quantify the impact of an improvement initiative by assessing the change of measurements relative to a baseline. In statistics-based analysis, statistical tools help to identify and to control variation in processes over time; surveys are used to collect information on the improvement from people who are either directly (employees) or indirectly (customers) affected; cost-benefit analysis helps to quantify the financial impact of the SPI initiative.

As illustrated in Figure 1, the gap analysis of evaluation quality concept constrains that evaluation strategies may be eligible for the specific SPI initiative. Table III summarizes the criteria on which the evaluation strategy should be selected. The criterion 'Measurement levels' identifies a strategy depending on the selected SIs and the corresponding process, project, product, organization, and external level (Table I). The 'cost' criterion provides a relative rank of the required resources to perform the corresponding evaluation strategy.

The accuracy of the evaluation is influenced by the extent to which the last criterion, 'confounding factors', can be controlled. Confounding factors represent a fundamental threat for the evaluation of a process improvement initiative if any kind of comparison is used to assess its effects. A comparison is said to be confounded if the observed difference between two values (the effect of a treatment) is not solely caused by the treatment but can be partially attributed to an external factor [80]. Table IV summarizes typical confounding factors encountered in the evaluation of improvement initiatives.

Looking at Table III, we assess the confounding factors for the 'basic comparison' and 'statistics-based analysis' strategies as controllable. For example, in the case of the basic comparison strategy, it is common

Table III. Criteria for selecting evaluation strategies.

| Strategy | Measurement levels | Cost | Confounding factors |
|---|---|---|---|
| Basic comparison | Process, project, and product | Medium[a] | Controllable |
| Statistics-based analysis | Process, project, and product | High | Controllable |
| Survey | Product, organization, and external | Low | Challenging |
| Cost–benefit analysis | Product, organization, and external | Medium | Challenging |

[a]If metric collection is not factored in.

Table IV. Confounding factors (adapted from [27]).

| Factor | Description |
|---|---|
| Project type | New development and enhancement/maintenance projects have different properties, and they should not be treated as the same during evaluation, that is, comparison of success indicators from different project types should be avoided. |
| Development model | Different project development life cycles such as waterfall model and spiral model have different project characteristics and potentially confound the evaluation. |
| Product size and complexity | Product size (lines of code, function points, etc.) and complexity (number of features, cyclomatic complexity, etc.) have to be taken into consideration during the evaluation. |
| Product domain | The product domain difference also affects the evaluation. Front-end applications, server-side systems, and embedded software are different types of product domains that should not be put on par during the evaluation. |
| Technology factors | Technological factors such as the programming language and tool support can influence indicators such as productivity and effort. |
| Process compliance | The degree to which the standard process is followed in the actual implementation should be considered in the evaluation as this can give indications to what extent the improvement can be actually attributed to the SPI initiative. |
| Employee factors | The staff working in the project might differ in experience level and measurement on productivity and efficiency should take staff experience into consideration. In addition to that, employee turnover in the organization may also affect the evaluation result. |
| Time factors | Time can be seen as a factor that can affect the evaluation result. When conducting a customer survey on product quality, the time that the product has been in use needs to be considered. |
| Multiple improvement initiatives | It is difficult to ensure that a particular improvement is attributed to a specific SPI initiative. Several improvement initiatives that run in parallel would create traceability issues in the evaluation. For example, when calculating the cost saving from a specific improvement initiative, care should be taken not to count the saving twice as the saving might also be attributed to another improvement initiative. |

SPI, Software Process Improvement.

to apply the matching technique or linear regression models [27]. On the other hand, controlling confounding factors in ‘cost-benefit analysis’ or ‘survey evaluation’ strategies is more challenging. Surveys collect quantitative and qualitative data from human subjects. Hence, it is important to create a profile of the surveyed individuals in order to group the acquired data into homogeneous categories. In the cost-benefit analysis strategy, it is crucial to quantify both direct and indirect costs and tangible and intangible benefits [27].

As we discussed in Section 4.2.2, the traceability of improvement initiatives decreases along the MLs because of timing and isolation issues. A viable approach to compensate for the effects of multiple improvement initiatives is to let internal experts weigh the contribution of individual initiatives [81]. Confounding factors related to timing and potential solutions are discussed in Section 4.5.

Example box 6: evaluation strategy

As the initiative is confined initially to pilot projects, ALPHA decides to select the basic comparison strategy. In phase II, when the initiative is rolled out to all projects within the organization, the survey strategy will be used to assess the long-term effects on the developed products. As the organization already conducts customer surveys on a regular basis, they can serve as a baseline. Before commencing phase II, the survey instrument is updated to include the measures defined for the product level.

## *4.5. Evaluation implementation*

The goal of the improvement evaluation is to satisfy the information needs of the respective stakeholders defined in evaluation scoping (Section 4.2).

In SPI-MEF, an improvement evaluation is conducted according to a planned schedule, consisting of the analysis of measures collected at a certain ML, and requires the involvement of roles with the expertise to judge the impact of the improvement initiative. Therefore, each evaluation instance is assigned a time, a ML, and one or more experts that conduct the evaluation.

*4.5.1. Scheduling.* The motivation to plan the evaluation schedule is to introduce means to control timing as a potential confounding factor. The principle idea behind this is that the effects of a certain improvement initiative may be measurable at different points in time, depending which ML is considered in the evaluation.

Example box 7: scheduling

ALPHA defined the initial evaluation schedule for 12 months after the introduction of code inspections. The schedule serves as an indication and may be updated to the actual progress in the projects.ProductA – feature release 1 – pilot projectsThe first feature release comprises two projects (pilots 1 and 2 in Figure (a)) in which code inspections are piloted. The inspections are held monthly on a sample of the newly implemented code. As the practice requires some training, the lag factor (LF) is estimated to 4 months and evaluation *a*, establishing a baseline for the process-level measures, is scheduled accordingly. At month six, evaluation *b* is scheduled to assess the process level again and compare it with evaluation *a*.

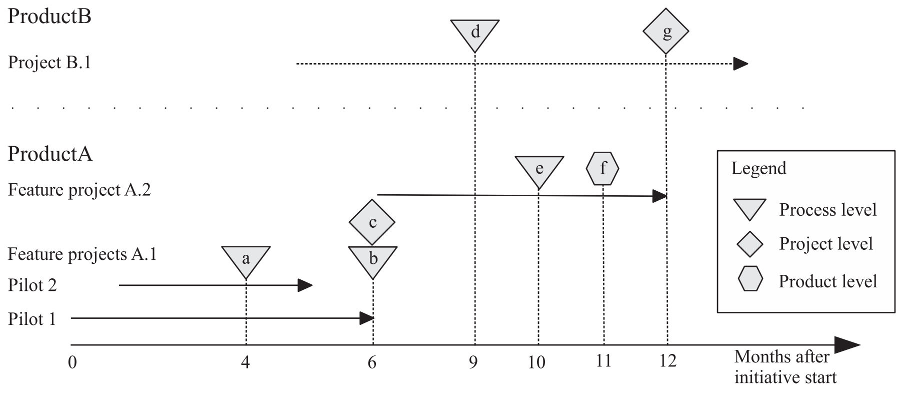


Figure (a): Evaluation schedule

Simultaneously, evaluation *c* assesses both pilots on the project level by comparing the collected measures with the expert estimates from a previous project in ProductA (no historic data was available, and the estimates were elicited when the measures where defined). The outcome of evaluation *c* serves as a baseline on the project-level measures.

*ProductA – feature release 2*

At month 10, code inspections were introduced in all projects. In evaluation *e*, as the degradation factor (DF) was estimated to 6 months, the baseline of process-level measures is updated. At month 11, a survey is initiated to elicit customer satisfaction on ProductA after the first feature release. Thereby, it is assumed that a LF of 5 months (after feature release) is needed to receive reliable feedback (the specific customers are known to ALPHA). The results of the survey (evaluation *f*) are added to the product-level baseline.

*ProductB*

Development of ProductB commenced at month five (Figure (a)). The first process-level evaluation (*d*) is scheduled for month nine. As evaluation *a* occurred in month four, it could still be used as a baseline (DF = 6 months). There is however evaluation *b* at month 6, which is then used as baseline for evaluation *d*. At month 12, the project-level measures are evaluated against evaluation *c* (the current baseline), and, as the DF for the PROJECT level was estimated to 6 months, the result of evaluation *g* is defined as the new baseline for the project level.

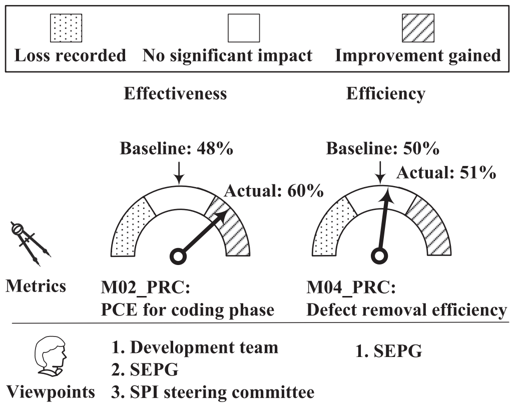
The temporal distance in the MLs (Section 4.2.2) supports the idea of a lag factor, which is also referred as the time lag between the cause (SPI initiative) and the corresponding effect (improvement) [82, 25]. This latency needs to be considered when determining the appropriate time to evaluate.

On the other hand, one has also to consider how long a measurement result is valid, that is, how long can it be of value to support decision making processes and be representative for what is actually assessed?[‡] Because of this validity decay of measurement results, periodic evaluations are needed in order to make the effects of the improvement visible over time, as exemplified by Herbsleb *et al.* [83], Jarvinen and van Solingen [47], Savioja and Tukiainen [84], Jarvinen *et al.* [48], Iversen and Ngwenyama [26], and Moreau *et al.* [79].

In SPI-MEF, we use the terms lag factor (LF) and degradation factor (DF) to designate the improvement effect latency and, respectively, the validity decay of measurement results. DF defines how long an evaluation result may support and be valid for the decision making process. As a consequence, a periodic evaluation schedule is required (example box 7). LF and DF are determined by the ML, the conducted improvement initiative, the degree to which the changes are actually implemented, and external factors that may stall progress. Dror [25] proposed statistical process control tools and data mining techniques to identify causality links between improvement action and effect. Historical data from organizations could therefore be used to create context-sensitive heuristics of improvement timings, that is, define the LF and DF either on the basis of collected data or on expert opinion gathered from employees.

Example box 8: analysis

Evaluation *b* is performed at month six to assess the initiative at the process level (example box 7). The project manager of pilot 1 analyses the effectiveness and efficiency of the introduced code inspections (according to the metrics derived in example box 5). The aim of evaluation *a*, conducted by the same product manager, was to establish a baseline and to define a deviation range that demarcates improvement, stagnation, or decline of a metric. Evaluation *b* shows a significant improvement in the effectiveness of the process, whereas efficiency remained stable. The analysis results are eventually communicated to the respective viewpoints that were defined in the evaluation scope. The figure shows the results of the evaluation (the process metrics are phase containment effectiveness for the coding phase and defect removal efficiency) and the targeted viewpoints.



*4.5.2. Analysis.* The aim of the analysis is to provide an evaluation to which degree a certain measurement has changed because of the enacted improvement initiative. To this end, the expert who was assigned to each measure when they were determined (Section 4.3) rates the change compared with the baseline. The analysis performed at this stage serves as an intermediate product that is reused when the holistic view is created, as discussed in Section 4.6.

## *4.6. Holistic view*

By defining a model that assesses the improvement from the viewpoint of the involved stakeholders, an important aspect of improvement evaluation can be addressed, namely increasing the visibility of the improvement initiative as a whole (Section 3.3, Figure 1). Such a representation would be beneficial for several reasons. First, the success of an initiative could be asserted with more confidence because it is assessed considering the involved stakeholders. Second, it could show, given that the appropriate metrics were collected, if the improvement has a positive impact on the organization as

[‡]For example, the standard CMMI appraisal method for software improvement (SCAMPI) [30] defines a degradation factor of 3 years for class A appraisals.

a whole or if the change negatively influences aspects that would not have been considered initially. Third, it can be used as an aid to communicate results of the improvement in an efficient way, as the amount of data produced in the individual evaluations is reduced.

The major aim of the holistic view concept is to provide an aid to communicate the improvement to the different stakeholders. To achieve this goal, we define improvement indicators that can be represented in a Kiviat diagram (example box 9). The important information that is shared between the stakeholders by looking at these diagrams is that the impact of the improvement may be different depending on who is assessing it. Interesting cases are given if there is a disagreement on the outcome of the initiative, that is, the EVs diverge. Such scenarios should give reason for further analysis of the implemented initiative.

*4.6.1. Considerations for the model.* In order to show the overall or compound impact of an improvement initiative it is necessary to define an appropriate model that is able to aggregate the results of individual evaluations (and metrics) into a representative score. We identified three basic aspects that need to be considered for such a construction of the model:

1. Normalization of the different metrics to enable a meaningful aggregation.
2. Compensation for the different orders of magnitude in the values of the metrics, that is, consider that a small difference in one metric may have effectively more impact than a larger difference in another.
3. Consideration of the individual viewpoints to include the relative importance in improving a specific metric.

The third point has less a technical rather than a qualitative rationale. The model should take the subjective change, as it was experienced by the involved parties, into account. This means that each metric should be given a weight, defined by the viewpoints that are interested in the result of the evaluation. It is assumed that in this way, the evaluation of the improvement initiative gains realism by representing the actual situation and reveals possible imbalances in the change effort, as it was perceived by the involved stakeholders.

The first two aspects could be implemented by an impact rating (IR), in which the evaluator maps the change in a metric into an ordinal scale, which would both normalize the metrics and compensate the differences in orders of magnitudes.

*4.6.2. Subjective value of improvement.* To calculate the improvement for each EV and ML, that is, the Subjective Value of Improvement (SVI), we use two components.

The first component is the subjective weight in which each viewpoint defines a weight of subjective importance to every metric. This means that the stakeholders of the improvement initiative within the implementer, coordinator, and sponsor viewpoints have to agree on a subjective weight. The second component is the Impact Rating (IR). Here, the expert who conducted the individual metric evaluations, as presented in Section 4.5.2, rates the impact of the improvement initiative on the respective metric according to an 11-point Likert scale (example box 9). One can choose also a 7-point or 9-point Likert scale, however research suggests that lower than 6-point scales generally produce less valid scores, have less discriminating power, and are less preferred by its users [85]. Because the IR is subjective in nature, the organization should discuss and agree upon guidelines on how to perform the rating with the aim to improve the consistency in the rating between different metric-experts.

The SVI is then calculated as follows:

$$SVI = \sum_{id} (SW_{id} * IR_{id})$$

where *id* refers to the respective metric identified in the determination of measures (Section 4.3). Because the IR component is based on previous evaluations, their degradation factor (Section 4.5.1) determines if the evaluation results are actually considered at the point in time when the SVI is calculated.

From a measurement theory point of view, the calculation of SVI is questionable as it involves mathematical operations, which, in a strict sense, are not applicable on ordinal scales [86]. On the other hand, Stevens [87] pointed out that it can be practical to treat an ordinal scale as an interval scale. Furthermore, several studies have empirically shown that it matters little if an ordinal scale is treated as an interval scale [86].

Because the aim of the holistic view is to provide an overview of the improvement (the individual evaluations are more appropriate data source for decision making), the SVI has to be seen as an index or score. It gives an indication of the improvement, rather than being a metric, which in its formal definition has to fulfill the representation condition of a measurement [88].

*4.6.3. Aggregated subjective value of improvement.* If evaluations, as presented in Section 4.5, are conducted on different target entities (e.g., projects or products), the calculation of the SVI needs to consider differences in invested resources. Therefore, the ASVI is used. The SVI is weighted by an investment unit (IU):

$$ASVI = \sum_{te} \left( \frac{SVI_{te} * IU_{te}}{IU_{Total}} \right)$$

where *te* refers to the SVI and the IU of the respective target entities (i.e., projects or products). $IU_{total}$ is the sum of all investments in the target entities. The IU can be regarded as the resources that were spent in the implementation of processes, projects, or products on which the individual evaluations are based on.

Example box 9: holistic view

ALPHA decided to conduct a holistic evaluation 12 months after code inspections were introduced as at least one evaluation for each targeted ML would have been performed at that time (example box 7 for the evaluation schedule). First, the SVI is calculated to see how the different EVs perceived the improvement. Then, the Aggregated SVI (ASVI) is used to illustrate the impact of the initiative on the three MLs that were scoped for the evaluation.The EV assessment uses the SVI as an indicator of the improvement at a specific ML. For this example, we show the outcome on the process level, using the results from evaluation *e* (Figure (a) in example box 7) as a basis. The project manager, responsible for evaluation *e*, reassesses the impact of code inspections on feature projectA.2, using the Likert scale shown in Figure (b). Representatives from the involved EVs (development team, SEPG, and SPI steering committee) weigh the individual metrics according to their relative importance, that is, whether they consider effectiveness or efficiency more critical to fulfill the aim of the initiative.

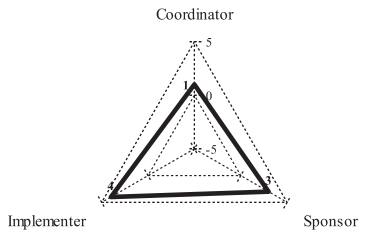


Figure (a): Process level / Individual viewpoints (SVI)

| Impact assessment scale | Value |
|---|---|
| High gain | +5 |
| Considerably high gain | +4 |
| Moderate gain | +3 |
| Considerably low gain | +2 |
| Low gain | +1 |
| No impact | 0 |
| Low low | -1 |
| Considerably low loss | -2 |
| Moderate loss | -3 |
| Considerably high loss | -4 |
| High loss | -5 |

Figure (b): Impact rating

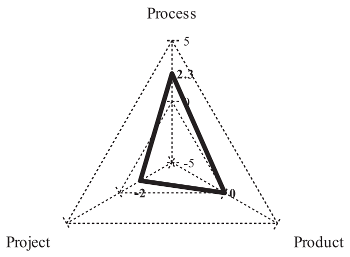


Figure (c): Aggregated viewpoints (ASVI)

Figure (a) shows that, on the process level, the implementer viewpoint perceived the introduced code inspections more valuable than the coordinator. Although all viewpoints identify an improvement, this outcome indicates that the implementation of the initiative can be further improved. Figure (c) is based on evaluations *d*, *e*, *f*, and *g* (Figure (a) in example box 7), as those are the only evaluations that are within the DF of 6 months. There, the ASVI is calculated on the process, project, and product level. Looking at the project-level assessment, we can observe a decline in performance. This indicates that the outcome of the initiative is not coherently seen as a success by all stakeholders.

## 4.7. Summary

Sections 4.1–4.6 described the framework, SPI-MEF, targeted at evaluating SPI initiatives. SPI-MEF aims at providing an SPI evaluation that addresses the challenges discussed in Section 3. The framework is based on several key concepts that span from scoping the evaluation, determining the required measures, to analyzing the gathered data (Figure 1). In Section 5, we illustrate how the framework was validated.

## 5. VALIDATION

The aim of the validation is to determine whether the framework is able to support practitioners in the evaluation of SPI initiatives. Section 5.1 describes the design of the validation, whereas Section 5.3 presents its results. Section 5.2 discusses threats to the validity.

### *5.1. Research method*

We designed a validation process in which expert judgment from researchers and industry experts was collected, analyzed, and used to refine SPI-MEF to its final version as it is presented in Section 4. The concrete objectives of this process were the following:

1. To identify deficiencies in the proposed concepts.
2. To assess the applicability of the framework from a practitioner's point of view.
3. To elicit improvement opportunities for the framework.

As a basis for the validation served, a document describing the concepts on which the framework is based upon.

*5.1.1. Selection of experts.* We validated the framework by using both researchers and industry experts in order to address both theoretical aspects and the practicality of the framework. Because the assessment of the framework's applicability was deemed critical, we selected researchers with industry experience or working closely with industry. Figure 4 provides an overview how the gathered expert judgment is distributed in industry and academia and which data collection mechanisms were employed. The number in curly braces indicates how many individual experts are included in the respective category.

The researchers' expert judgment was gathered through a semistructured interview [89] and was depending on the interviewee's accessibility, conducted in a face-to-face meeting or a telephone call. The group of industry experts was approached either by a semistructured interview or a self-administered questionnaire [90], again depending on their accessibility.

Thirteen researchers and 11 industry experts were selected and contacted, whereas nine and seven subjects of the respective groups agreed to participate in the study. All researchers agreed to provide 45–60 min for the interview and industry experts scheduled a 1.5 h meeting. Table V gives an overview of the characteristics of the participating subjects.

All but one researcher were at the time of the investigation employed at the Blekinge Institute of Technology; nevertheless, they experienced education in various universities in Sweden, Germany,

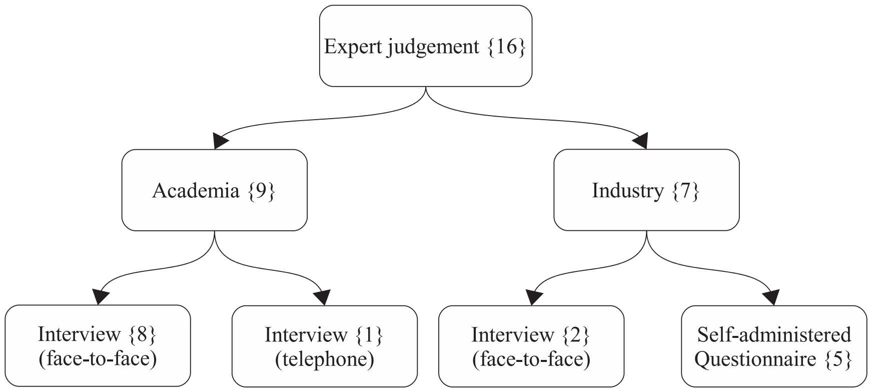

Expert judgement {16}
Academia {9}
Industry {7}
Interview {8} (face-to-face)
Interview {1} (telephone)
Interview {2} (face-to-face)
Self-administered Questionnaire {5}

Figure 4. Sample of expert judgment and data collection mechanisms.

Table V. Profile of the industry and academia experts.

| Academia | | |
|---|---|---|
| Years[a] | Research area | |
| 8/7 | Software measurement/estimation, SPI, project management, requirements engineering, and business process modeling | |
| 3/0 | Large-scale software management, software quality, product management, and software process management | |
| 3/3 | Software product line engineering, SPI, Agile and Lean software development, and software measurement | |
| 3/4 | Value-based software engineering | |
| 7/3 | Software verification and validation and search-based software engineering | |
| 3/6 | Strategic software engineering and software product management | |
| 10/0 | Requirements engineering and software architecture | |
| 30/10 | Software architecture/reuse, process engineering and measurement, and SPI | |
| 12/8 | Verification and Validation, Automated Software Engineering, requirements engineering, human/social aspects of software engineering, and search-based software engineering | |
| Industry | | |
| Years[b] | Business unit | Company size[c] |
| 18 | Multimedia | Large |
| 12 | Product Development Excellence | Large |
| 5 | Engineering | Small |
| 5 | Research and Development Engineering | Large |
| 10 | Data Networks Department | Large |
| 12 | Enterprise Mobility Solutions | Large |
| 10 | Research and Development | Large |

[a]The values denote the experience in academia/industry.
[b]The values denote total work experience.
[c]The company size is according to the European Recommendation 2003/361/EC.

Australia and Turkey, which allows the assumption that their expertise was not streamlined. The industry experts were employed in four different companies located in Sweden, the USA, Malaysia, and Singapore. The companies' core businesses were telecommunications, electronic and electrical manufacturing, and global communication solutions of software intensive systems.

*5.1.2. Definition of data collection instruments.* In this section, we describe the design of the interview and the questionnaire. For the design of the interview instrument, we followed the guidelines by Kvale [91]. The interview questions address the framework's concepts and were assigned a priority (Figure 1). The prioritization of the questions is important as the interview should be designed in a way that it can be finished in the stipulated amount of time while considering all high priority topics.

To clarify the purpose of the interview and the contents that need to be validated, a distilled description (10 pages) of the framework's concepts was prepared and sent as preparatory documentation to the interviewees. We held also a short presentation at the beginning of each interview meeting to provide an introduction to the framework and to give a refresher of its concepts. All interviews were conducted by the same interviewer to maintain the consistency in the way the questions are presented. Three note takers recorded the answers during the interview sessions.

The self-administered questionnaire was designed following Kasunic' guidelines [92]. The questions were formulated in a way such that the respondents could express their degree of agreement/ disagreement (using a Likert scale). Additionally, the respondents were urged to motivate their answers with a few sentences. Furthermore, the wording of the questions was selected very carefully, taking Salant and Dillman's [93] advises into consideration.

The quality of the interview instrument, the questionnaire, and the prepared supplementary material [29] was improved by piloting the interview and the questionnaire with three software engineering students. The understandability and clarity in presenting the concepts of the framework were verified, and the questionnaire was assessed regarding question formulation, layout, and the overall compilation process.

### 5.2. *Threats to validity*

The discussion on the threats to validity of this research is organized according to the categorization proposed by Wohlin *et al*. [94]. Threats to internal validity (Section 5.2.1) are concerned with the observed relationship between the treatment and the outcome, that is, the external factors that can influence an independent variable with respect to the causal relationship with a dependent variable. Threats to external validity (Section 5.2.2) are factors that can influence the ability to generalize the results to a wider scope than covered by the study. Construct validity (Section 5.2.3) is concerned with the relationship between theory and the observed outcomes of the research, that is, with the ability to generalize its results to the theoretical construct that motivate the research. Threats to the conclusion validity (Section 5.2.4) are concerned with factors that affect the ability to draw the correct conclusions from the conducted study.

*5.2.1. Internal validity.* Three threats to internal validity, related to the gathering of expert judgment, were identified: instrumentation, maturation, and selection.

The instrumentation threat is caused by bad design of artifacts [94] used in the expert judgment elicitation. Those can lead to misunderstandings regarding the discussed topic and weaken the results from the gathered data. To minimize this threat, the preparatory document and the interview questions were piloted first with three software engineering students to test whether the artifacts are clear and understandable. Afterwards, the preparatory document and the interview questions were refined.

The maturation threat exists if the experts' behavior changes during the elicitation process as the time passes [94]. This can distort the gathered results if the subjects acquire new knowledge during the process, or become detached [94]. This threat is regarded as minor because the interviews with researchers were conducted during a meeting, which lasted approximately 1 hour each. The written questionnaire was compiled and returned by all industry experts within 2 weeks; because no deadline to return the questionnaire was given to the subjects, the rather quick response indicates that they were committed to the task and had interest in providing useful information. Furthermore, the questionnaire was designed to present the needed information and the questions concisely and precisely such that it can be compiled within approximately 1 hour.

The selection threat is concerned with the varying human performance and potential biases introduced by the selected subjects for the investigation, for example, higher motivation of volunteers may lead to better results [94]. As the presented profiles in Section 5.1.1 show, both researchers and industry experts have several years of experience in their respective fields. Obviously there are differences in expertise in the specific areas of interest, but this was regarded rather as an advantage than a threat because a major goal of the validation was to identify new, not yet considered, aspects for the measurement and evaluation of SPI.

*5.2.2. External validity.* The threat of selection and treatment is caused by not having a representative sample of the population [94]. To address this threat, the selection of researchers took also the industrial experience of the subjects into consideration. The industry experts selected in this study were employed in different companies with different core businesses from Europe, the USA, and Asia. Nevertheless, there is a moderate threat of selection bias because of the convenience sampling of researchers and industry experts.

*5.2.3. Construct validity.* In this category, two threats for this research were identified: mono-operation bias and evaluation apprehension.

Mono-operation bias is caused by considering only a single subject, independent variable or case, and hence, the study may not fully represent the investigated theory [94]. This threat is considered moderate because two groups with different background were considered, and for each group (academic and industry experts), more than one subject was involved.

The threat of evaluation apprehension is caused by the human tendency to behave differently while being evaluated [94]. This can distort the result of the study because the subjects may perform better than in a regular, unobserved, situation. To tackle this issue, the experts were guaranteed their anonymity and that their answers were only used by the researchers involved in the study.

*5.2.4. Conclusion validity.* Three threats were identified in this study that fall under this category: random heterogeneity of subjects, random irrelevancies in experimental setting, and searching for a certain result.

Random heterogeneity of subjects is a threat caused by a heterogeneous sample such that individual differences within the sample could affect the study's result [94]. To minimize this threat, the experts were selected on the basis of their competencies and knowledge in software engineering and SPI.

Random irrelevancies are elements outside of the study setting, which can disturb its conduct [94]. This threat is considered as minor because the interviews were conducted in an uninterrupted session and in a quiet environment. There were no discussions about the questions before the interview that could have influenced the interviewee's answers.

Searching for results or 'fishing' is the tendency of the researchers to search for a certain result or answer and ignore the inconvenient information [94]. To minimize this threat, all answers from the experts, whether they were positive or negative, were recorded and analyzed regardless the researchers expected outcome.

## *5.3. Results*

The Subsections 5.3.1–5.3.7 summarize the main issues regarding the presented concepts (Figure 1). The impact on SPI-MEF and the applied refinements are reported in each subsection.

In essence, the approach proposed by SPI-MEF for SPI evaluation was taken very positively by both academic researchers and industry practitioners. Both groups agreed that SPI-MEF has the potential to provide a systematic way of evaluating process improvement impact. However, there were several suggestions brought forward to improve the framework in terms of increasing its applicability in practice.

*5.3.1. Gap analysis of evaluation quality.* Higher accuracy and better coverage (Section 4.1) is of course good to achieve. However, it may not be feasible for companies to achieve both simultaneously in the first place because resources may be constrained. Therefore, it is important to know which one is important to consider first. There was divergence in the answers of the interviewees on this issue. Some suggested considering accuracy first, while others considered coverage as more important. However, their answers revealed that giving emphasis on accuracy first has some formidable advantages. Achieving accurate and valid results first can increase the confidence on the quality of evaluation, which then can motivate to increase the coverage adding more complexity in the evaluation and investing more resources. If the intention of the evaluation is to see a more complete picture of the improvement benefits first and identifying the problem areas, then coverage should obtain more emphasis than accuracy.

The cost of the evaluation was considered as a very important factor. The absence of cost considerations may lead organizations to opt for a good enough evaluation and discourage them from expending money to gain high accuracy and coverage to achieve a holistic evaluation. Therefore, the cost factor should be included in this matrix, and a discussion on the relation between quality of evaluation and cost should be included in the concept.

Impact on SPI-MEF: In addition to the previously present two dimensions of accuracy and coverage, a third dimension covering the cost aspect was added to the framework (Section 4.1).

*5.3.2. Evaluation scoping.* Some confusion arose regarding the roles in each viewpoint and in the interpretation of the categorization of the viewpoints (Section 4.2.3). For example, it was not clear that the same role can subsume different viewpoints (e.g., a project manager who has the viewpoint of a coordinator in the project level can also have the implementer viewpoint in the product level). This may be due to the short document provided to the experts as an introduction to the framework that did not suffice to clarify this aspect.

Impact on SPI-MEF: To reduce chances of misinterpretation, the example describing the EVs in Section 4.2.3 explicitly discusses this point. The extended scenario [29] was enhanced with motivations for the allocation of roles to the different viewpoints.

*5.3.3. Determination of measures.* The feedback to the questions regarding the proposed method (Section 4.3) was twofold: on one hand, the approach was judged as systematic and comprehensive, which indicates that the method can be of practical use and provides appropriate support for practitioners. On the other hand, some experts perceived the approach as quite complex and time-consuming, which implies caveats in its applicability in terms of training and education of employees and in justifying the additional resources needed for its implementation.

Impact on SPI-MEF: The concerns about the complexity of the method can be addressed by considering that the process of derivation of measures is an iterative one and is indeed scalable to more realistic settings than those that were shown in the example given in the interview material. Adding to the framework, as it was proposed by one interviewee, a palette of goals, questions, and metrics on which the user can base his measurement program was regarded by the authors as inflexible and difficult to maintain. It would be more appropriate to define a step-by-step guide that leads the practitioner to formulate his own goals and questions and then provide a pool from which he can pick the needed metrics (Section 4.3.1). Clearly, this implies more effort on the part of the user of the framework; however, this approach makes it flexible and applicable in a wider range of scenarios.

*5.3.4. Primary and complementary measures.* The introduction of primary and complementary measurements (Section 4.3.1) necessitates a precise definition of these new terms. As it was observed by one academic expert, the term complementary may induce misunderstandings, and indeed, an industry expert interpreted the measures as the needed (primary) and good to have (complementary) ones. Clearly, this was not the intended interpretation, and several remedies were discussed to avoid this misinterpretation.

Impact on SPI-MEF: As a result, a renaming of the terms was discarded, because any naming inherits ambiguities depending on the background of the reader. Therefore, in order to minimize the space for interpretation, the definition of the terms primary and complementary were enhanced, and the exemplified measurement derivation was elaborated with more detailed steps. Additionally, it was made very explicit in the framework that complementary measures are not optional (good to have) but necessary for a complete evaluation (Section 4.3.1).

Furthermore, a pool of commonly used metrics, grouped according to MLs, was provided in [29]. This should support the practitioner initially in identifying primary and complementary measurements. It should be noted, however, that the pool has to be seen as a reference, and it should not be regarded as an exhaustive set of metrics.

*5.3.5. Confounding factors.* This concept (Section 4.4) was specifically put to the industry experts in order to exhibit if they consider it as an important issue in the practical evaluation of SPI. Compared with the other questions, the input to this concept was rather thin, although positive. Indeed, it was deemed as a necessary step to create awareness for confounding factors and consider them appropriately in the construction of baselines, and practical ways to control them in an industrial setting were needed according to the industry experts.

Impact on SPI-MEF: In the final framework, a short description of typical confounding factors that need to be taken into consideration for evaluation planning or during the evaluation was included, along with guidelines on how to address them (Table IV; Section 4.4).

*5.3.6. Evaluation scheduling.* Both the academic and industry experts agreed that the concept of LF and DF, and periodic evaluation is conceptually right (Section 4.5.1). The main concerns were, however, how to come up with these values in the first place when the initiative is new or when several improvement initiatives are running in parallel. DF was considered harder to define as compared with LF. DF is the key concept that helps to define the time bounds for periodic evaluation and could also help to determine the optimum interval between successive evaluations, which is important to minimize the cost of evaluation.

Impact on SPI-MEF: It was suggested to provide some guidelines on how to come up with the values of LF and DF. These could however be misleading as long as empirical evidence or heuristics for LF and DF are not available. Therefore, at the beginning when the framework is introduced in an organization, experienced practitioners and experts in the field of process improvement could help to define these values. Thereafter, organizations can learn and improve their accuracy to determine LF and DF when they gain more and more empirical evidence for appropriate values of LF and DF.

*5.3.7. Holistic view.* The scrutiny of the holistic view concept (Section 4.6) revealed some important characteristics regarding this approach to present improvement and which strengths and weaknesses are inherent in this approach. It was confirmed that the target audience for the holistic representation resides in top-level management, for which the reduction in details can be seen as an advantage. The tool is therefore less adequate as decision support for the continuation or further refinement of an improvement initiative (this has to be performed at a lower level where details are conserved) but rather expresses the health on the initiative and reveals if the expected benefits are achieved. The subjective element, gut feeling, as it is integrated in the model, was judged both positively and negatively. Subjective ratings in improvement assessment are used in industry and therefore applicable in the holistic view. The contribution of the framework would therefore be the formalization of that process.

Impact on SPI-MEF: To make the subjective rating in the improvement assessment more homogenized and consistent among the different stakeholders, the framework prescribes to create guidelines on how to perform such a rating (Section 4.6.2). The extended scenario [29] provides an example how such a guideline can be realized as a help to homogenize IR.

## 6. CONCLUSION

This paper presents a framework for the measurement and evaluation of SPI initiatives (SPI-MEF). SPI-MEF describes and exemplifies the use of the concepts of evaluation quality and scoping, determination of measures, and evaluation scheduling and analysis. The framework's concepts were derived from the best practices gathered in a systematic literature review on SPI measurement and evaluation [27].

Once the framework was created initially, it was evaluated by 16 academic and industry experts with a median of 6 years of combined SPI experience in both research and practice. The focus of the evaluation was to validate that the framework integrates the important aspects of SPI evaluation and, on the other hand, provides support for practitioners. According to the experts, the contribution of the framework lies in the structured and nevertheless flexible approach.

SPI-MEF gives concrete guidelines on how to scope the evaluation *before* the improvement initiative is implemented. This allows practitioners to increase the visibility of the improvement effort within the company and to plan the required resources needed for the evaluation. As SPI-MEF builds upon the widely known GQM paradigm, existing measurement programs in an organization can be refocused on the evaluation of SPI and hence reusing existing resources and infrastructure. On the other hand, SPI-MEF provides also guidance to initiate a new measurement and evaluation program.

Perception of improvement success varies within the functional structure of an organization. SPI-MEF provides means to capture and to communicate improvement outcomes from different viewpoints, facilitating the understanding of the effects of process change. As such, SPI-MEF is a step forward in the ability to determine the success of SPI initiatives, increasing the confidence in the evaluation results.

### *6.1. Future work*

Future work, refining, and extending SPI-MEF will include the integration of a cost model that will further increase the adaptability of the framework and improve the support for practitioners selecting evaluation strategies. Furthermore, we target a dynamic evaluation [28] of SPI-MEF, instantiating the framework in a specific company context and piloting the implementation within an initiative that aims at improving the alignment between requirements engineering and verification activities.

## REFERENCES

1. Kitchenham B, Pfleeger S. Software quality: the elusive target. *IEEE Software* 1996; **13**(1):12–21.
2. Scacchi W. Process models in software engineering. Encyclopedia of Software Engineering. John Wiley & Sons: New York, 2001; 993–1005.
3. Sjøberg DIK, Dybå T, Jorgensen M. The future of empirical methods in software engineering research. *Proceedings Future of Software Engineering* (*FOSE*), IEEE: Minneapolis, USA, 2007; 358–378.
4. Wirth N. A brief history of software engineering. *IEEE Annals of the History of Computing* 2008; **30**(3):32–39.
5. Card DN. Research directions in software process improvement. *Proceedings 28th Annual International Computer Software and Applications Conference (COMPSAC)*, Hong Kong, China, 2004; 238.
6. O'Connor R, Coleman G. Using grounded theory to understand software process improvement: a study of Irish software product companies. *Information and Software Technology* 2007; **49**(6):654–667.
7. Dybå T. An instrument for measuring the key factors of success in software process improvement. *Empirical Software Engineering* 2000; **5**(4):357–390.
8. Canfora G, Garcia F, Piattini M, Ruiz F, Visaggio C. Applying a framework for the improvement of software process maturity. *Software - Practice and Experience* 2006; **36**(3):283–304.
9. Ferreira AIF, Santos G, Cerqueira R, Montoni M, Barreto A, Rocha AR, Barreto AOS, Silva RC. ROI of software process improvement at BL informatica: SPIdex is really worth it. *Software Process Improvement and Practice* 2008; **13**(4):311–318.
10. Goldenson D, Gibson D. Demonstrating the impact and benefits of CMMI: an update and preliminary results. *Technical Report CMU/SEI-2003-SR-009*, Software Engineering Institute, 2003. (Available from: http://www.sei.cmu.edu/library/abstracts/reports/03sr009.cfm.) [Accessed on 18 September 2013].
11. Hyde K, Wilson D. Intangible benefits of CMM-based software process improvement. *Software Process Improvement and Practice* 2004; **9**(4):217–228.
12. Mohagheghi P, Conradi R. An empirical investigation of software reuse benefits in a large telecom product. *ACM Transactions on Software Engineering and Methodology* 2008; **17**(3):1–31.
13. Murugappan M, Keeni G. Blending CMM and six sigma to meet business goals. *IEEE Software* 2003; **20**(2):42–48.
14. Redzic C, Baik J. Six sigma approach in software quality improvement. *Proceedings 4th International Conference on Software Engineering Research, Management and Applications (SERA)*, Seattle, USA, 2006; 396–406.
15. Sommerville I, Ransom J. An empirical study of industrial requirements engineering process assessment and improvement. *ACM Transactions on Software Engineering and Methodology* 2005; **14**(1):85–117.
16. Trienekens J, Kusters R, van Solingen R. Product focused software process improvement: concepts and experiences from industry. *Software Quality Journal* 2001; **9**(4):269–281.
17. Achatz R, Paulisch F. Industrial strength software and quality: software and engineering at Siemens. *Proceedings 3rd International Conference on Quality Software (QSIC)*, Dallas, USA, 2003; 321–326.
18. Salo O, Abrahamsson P. Integrating agile software development and software process improvement a longitudinal case study. *International Symposium on Empirical Software Engineering (ISESE)*, Noosa Heads, Australia, 2005; 193–202.
19. Jones C. The economics of software process improvement. *IEEE Computer* 1996; **29**(1):95–97.
20. Briand LC, Differding CM, Rombach HD. Practical guidelines for measurement-based process improvement. *Software Process: Improvement and Practice* 1996; **2**(4):253–280.
21. Paulish DJ, Carleton AD. Case studies of software-process-improvement measurement. *IEEE Computer* 1994; **27**(9):50–57.
22. Florac WA, Carleton AD. Measuring the software process: statistical process control for software process improvement. Addison-Wesley: Boston, 1999.
23. Freeman M, Beale P. Measuring project success. *Project Management Journal* 1992; **1**(23):8–17.
24. Abrahamsson P. Measuring the success of software process improvement: the dimensions. *Proceedings European Software Process Improvement (EuroSPI2000) Conference*, Copenhagen, Denmark, 2000.
25. Dror S. A process causality approach given a strategic frame. *Journal of Modelling in Management* 2007; **2**(1):40–55.
26. Iversen J, Ngwenyama O. Problems in measuring effectiveness in software process improvement: a longitudinal study of organizational change at Danske Data. *International Journal of Information Management* 2006; **26**(1):30–43.
27. Unterkalmsteiner M, Gorschek T, Islam AKMM, Cheng CK, Permadi RB, Feldt R. Evaluation and measurement of software process improvement – a systematic literature review. *IEEE Transactions on Software Engineering* 2012; **38**(2):398–424.
28. Gorschek T, Wohlin C, Carre P, Larsson S. A model for technology transfer in practice. *IEEE Software* 2006; **23**(6):88–95.
29. Unterkalmsteiner M, Gorschek T, Islam AKMM, Cheng CK, Permadi RB, Feldt R. Extended material to 'a conceptual framework for SPI evaluation', 2011. (Available from: http://www.bth.se/com/mun.nsf/pages/spi-mef.) [Accessed on 18 September 2013].

30. Standard CMMI appraisal method for process improvement (SCAMPI), version 1.3: method definition document. *CMU/SEI-2011-HB-001*, March 2011. (Available from: http://www.sei.cmu.edu/library/abstracts/reports/11hb001.cfm.) [Accessed on 18 September 2013].
31. ISO/IEC TR2 15504 – software process assessment: part 1–part 9. *Technical Report ISO/IEC TR2 15504*, ISO, Geneva, Switzerland, 1998.
32. Ares J, García R, Juristo N, López M, Moreno AM. A more rigorous and comprehensive approach to software process assessment. *Software Process: Improvement and Practice* 2000; **5**(1):3–30.
33. Ekdahl F, Larsson S. Experience report: using internal CMMI appraisals to institutionalize software development performance improvement. *EUROMICRO Conference*, IEEE Computer Society: Los Alamitos, USA, 2006; 216–223.
34. Fenton N. Metrics for software process improvement. Software Process Improvement: Metrics, Measurement and Process Modelling. No. 4 in Software Best Practice, Springer: Berlin, Germany, 2001.
35. Pulford K, Kuntzmann-Combelles A, Shirlaw S (eds.). A Quantitative Approach to Software Management: The AMI Handbook. Addison-Wesley: Boston, 1995.
36. Kuntzmann-Combelles A. Quantitative approach to software process improvement. Objective Software Quality, *Lecture Notes in Computer Science*, vol. **926**. Springer: Berlin, Germany, 1995; 16–30.
37. Paulk MC, Curtis B, Chrissis MB, Weber CV. Capability maturity model for software version 1.1. *Technical Report CMU/SEI-93-TR-024*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA, 1993. (Available from: ftp://ftp.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf.) [Accessed on 18 September 2013].
38. Debou C, Kuntzmann-Combelles A. Linking software process improvement to business strategies: experiences from industry. *Software Process Improvement and Practice* 2000; **5**(1):55–64.
39. Basili V, Weiss D. A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering* 1984; **10**(6):728–738.
40. Basili VR, Caldiera G, Rombach HD. The goal question metric approach. *Encyclopedia of Software Engineering* 1994; **1**:528–532.
41. Basili V, Heidrich J, Lindvall M, Munch J, Regardie M, Trendowicz A. GQM + strategies – aligning business strategies with software measurement. *First International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Madrd, Spain, 2007; 488–490.
42. Capability maturity model integration (CMMI) for development, version 1.3. *Technical Report CMU/SEI-2010-TR-033*, Software Engineering Institute, 2010. (Available from: www.sei.cmu.edu/reports/10tr033.pdf.) [Accessed on 18 September 2013].
43. Basili V, Lindvall M, Regardie M, Seaman C, Heidrich J, Munch J, Rombach D, Trendowicz A. Linking software development and business strategy through measurement. *Computer* 2010; **43**(4):57–65.
44. Park RE, Goethert WB, Florac WA. Goal-driven software measurement – a guidebook. *Technical Report CMU/SEI-96-HB-002*, Software Engineering Institute, 1996. (Available from: http://www.sei.cmu.edu/reports/96hb002.pdf.) [Accessed on 18 September 2013].
45. Goethert W, Siviy J. Applications of the indicator template for measurement and analysis. *Technical Report CMU/SEI-2004-TN-024*, Software Engineering Institute, 2004. (Available from: http://www.sei.cmu.edu/pub/documents/04.reports/pdf/04tn024.pdf.) [Accessed on 18 September 2013].
46. Järvinen J, Komi-Sirviö S, Ruhe G. The PROFES improvement methodology – enabling technologies and methodology design. Product Focused Software Process Improvement, *Lecture Notes in Computer Science*, vol. **1840**. Springer: Berlin, Germany, 2000; 269–302.
47. Jarvinen J, van Solingen R. Establishing continuous assessment using measurements. *Proceedings 1st Internation Conference on Product Focused Software Process Improvement (PROFES)*, Oulu, Finland, 1999; 49–67.
48. Jarvinen J, Hamann D, van Solingen R. On integrating assessment and measurement: towards continuous assessment of software engineering processes. *Proceedings 6th International Software Metrics Symposium (METRICS)*, Boca Raton, USA, 1999; 22–30.
49. McGarry J, Card D, Jones C, Layman B, Clark E, Dean J, Hall F. Practical Software Measurement: Objective Information for Decision Makers. Addison-Wesley: Boston, 2001.
50. Card DN, Jones CL. Status report: practical software measurement. *Proceedings 3rd International Conference on Quality Software (QSIC)*, Dallas, USA, 2003; 315–320.
51. International Organization for Standardization. ISO/IEC 15939 – software measurement process, 2002.
52. Bailey E, Jones C, Card D, Layman B, Dean J, McGarry J, Hall F. PSM guide 4.0b part 1, 2003.
53. Herbsleb JD, Grinter RE. Conceptual simplicity meets organizational complexity: case study of a corporate metrics program. *Proceedings 20th international conference on Software engineering (ICSE)*, Kyoto, Japan, 1998; 271–280.
54. Berry M, Jeffery R. An instrument for assessing software measurement programs. *Empirical Software Engineering* 2000; **5**(3):183–200.
55. Kasunic M, McCurley J, Zubrow D. Can you trust your data? Establishing the need for a measurement and analysis infrastructure diagnostic. *Technical Report CMU/SEI-2008-TN-028*, 2008. (Available from: http://www.sei.cmu.edu/reports/08tn028.pdf.) [Accessed on 18 September 2013].
56. Umarji M, Seaman C. Gauging acceptance of software metrics: comparing perspectives of managers and developers. *Proceedings 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM)*, IEEE: Lake Buena Vista, USA, 2009; 236–247.
57. Gorschek T, Davis A. Requirements engineering: in search of the dependent variables. *Information and Software Technology* 2008; **50**(1–2):67–75.

58. Stalhane T. Root cause analysis and gap analysis – a tale of two methods. Software Process Improvement, *Lecture Notes in Computer Science*, vol. **3281**. Springer: Berlin, Germany, 2004; 150–160.
59. Daskalantonakis M, Basili V, Yacobellis R. A method for assessing software measurement technology. *Quality Engineering* 1990; **3**(1):27–40.
60. Comer P, Chard J. A measurement maturity model. *Software Quality Journal* 1993; **2**(4):277–289, doi:10.1007/BF00403770.
61. Niessink F, Vliet HV. Towards mature measurement programs. *Proceedings 2nd Euromicro Conference on Software Maintenance and Reengineering (CSMR)*, Florence, Italy, 1998; 82–88.
62. Diaz-Ley M, Garcia F, Piattini M. MIS-PyME software measurement maturity model-supporting the definition of software measurement programs. Product-Focused Software Process Improvement, *Lecture Notes in Computer Science*, vol. **5089**. Springer: Berlin, Germany, 2008; 19–33.
63. Lowy A, Hood P. The power of the 2 × 2 matrix. Jossey-Bass: San Francisco, 2004.
64. Linberg KR. Software developer perceptions about software project failure a case study. *Journal of Systems and Software* 1999; **49**(2–3):177–192.
65. Zahran S. Software Process Improvement: Practical Guidelines for Business Success. Addison-Wesley, 1998.
66. Daskalantonakis MK. A practical view of software measurement and implementation experiences within Motorola. *IEEE Transactions on Software Engineering* 1992; **18**(11):998–1010.
67. Ebert C. Technical controlling and software process improvement. *Journal of Systems and Software* 1999; **46**(1):25–39.
68. Parkinson ST, Hierons RM, Lycett M, Norman M. Practitioner-based measurement a collaborative approach. *Communication of the ACM* 2010; **53**(3):142–147.
69. Gorschek T, Wohlin C. Packaging software process improvement issues a method and a case study. *Software: Practice and Experience* 2004; **34**(14):1311–1344.
70. Ivarsson M, Gorschek T. Tool support for disseminating and improving development practices. *Software Quality Journal* 2012; **20**(1):173–199.
71. Ramil J, Lehman M. Defining and applying metrics in the context of continuing software evolution. *Proceedings 7th International Software Metrics Symposium (METRICS)*, London, UK, 2000; 199–209.
72. Iversen J, Mathiassen L. Cultivation and engineering of a software metrics program. *Information Systems Journal* 2003; **13**(1):3–19.
73. van Solingen R, Berghout E. The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. The McGraw-Hill Companies: Maidenhead, UK, 1999.
74. Atkinson R. Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International Journal of Project Management* 1999; **17**(6):337–342.
75. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling (10th edn). John Wiley: Hoboken, 2009.
76. Basili V, Zelkowitz M, McGarry F, Page J, Waligora S, Pajerski R. SEL's software process improvement program. *IEEE Software* 1995; **12**(6):83–87.
77. Murugappan M, Keeni G. Quality improvement – the six sigma way. *Proceedings 1st Asia-Pacific Conference on Quality Software (APAQS)*, Hong Kong, China, 2000; 248–257.
78. Weiss D, Bennett D, Payseur J, Tendick P, Zhang P. Goal-oriented software assessment. *Proceedings 24th International Conference on Software Engineering (ICSE)*, Orlando, USA, 2002; 221–231.
79. Moreau B, Lassudrie C, Nicolas B, I'Homme O, d'Anterroches C, Le Gall G. Software quality improvement in France Telecom Research Center. *Software Process Improvement and Practice* 2003; **8**(3):135–144.
80. Pearl J. Why there is no statistical test for confounding, why many think there is, and why they are almost right. *Department of Statistics, UCLA*, July 1998.
81. van Solingen R. Measuring the ROI of software process improvement. *IEEE Software* 2004; **21**(3):32–38.
82. Norreklit H. The balance on the balanced scorecard – a critical analysis of some of its assumptions. *Management Accounting Research* 2000; **11**(1):65–88.
83. Herbsleb J, Carleton A, Rozum J, Siegel J, Zubrow D. Benefits of CMM-based software process improvement: initial results. *Technical Report CMU/SEI-94-TR-013*, Software Engineering Institute, 1994. (Available from: http://www.sei.cmu.edu/reports/94tr013.pdf.) [Accessed on 18 September 2013].
84. Savioja E, Tukiainen M. Measurement practices in financial software industry. *Software Process Improvement and Practice* 2007; **12**(6):585–595.
85. Preston CC, Colman AM. Optimal number of response categories in rating scales: reliability, validity, discriminating power, and respondent preferences. *Acta Psychologica* 2000; **104**(1):1–15.
86. Knapp TR. Treating ordinal scales as interval scales: an attempt to resolve the controversy. *Nursing Research* 1990; **39**(2):121.
87. Stevens SS. On the theory of scales of measurement. *Science* 1946; **103**(2684):677–680.
88. Fenton NE, Pfleeger SL. Software Metrics: A Rigorous and Practical Approach (2nd edn). PWS Publishing Company: Boston, 1997.
89. Seaman C. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering* 1999; **25**(4):557–572.
90. Jenkins CR, Dillman DA. Towards a theory of self-administered questionnaire design. *Journal of Sociology* 1997; **89**:373–401.
91. Kvale S. Interviews: An Introduction to Qualitative Research Interviewing (1st edn). Sage Publications: Thousand Oakes, 1996.

92. Kasunic M. Designing an effective survey. *Technical Report CMU/SEI-2005-HB-004*, Software Engineering Institute, 2005. (Available from: www.sei.cmu.edu/reports/05hb004.pdf.) [Accessed on 18 September 2013].
93. Salant P, Dillman DA. How to Conduct Your Own Survey. John Wiley and Sons: New York, USA, 1994.
94. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A. Experimentation in software engineering: an introduction. Kluwer Academic Publishers: Norwell, 2000.

## AUTHORS' BIOGRAPHIES

**Michael Unterkalmsteiner** received a BSc degree in Applied Computer Science from the Free University of Bolzano/Bozen in 2007, and an MSc degree in Software Engineering at the Blekinge Institute of Technology in 2009. He is working toward a PhD degree at the Blekinge Institute of Technology where he is with the Software Engineering Research Lab. His research interests include software repository mining, software measurement and testing, process improvement, and requirements engineering. His current research focuses on the cooptimization of requirements engineering and verification and validation processes.

**Dr Tony Gorschek** is a Professor of Software Engineering at Blekinge Institute of Technology. He has over 10 years industrial experience not only as a Senior Executive Consultant and Engineer but also as Chief Architect and Product Manager. Currently, he manages his own consultancy company, works as a Chief Technology Officer, and serves on several boards in companies developing cutting edge technology and products. His research interests include requirements engineering, technology and product management, process assessment and improvement, quality assurance, and practical innovation. Dr Gorschek bases his research on challenges identified in industry, then develops solutions in collaboration with industry practitioners, and ultimately validates and tests the solutions in a real industrial setting. Industrial partners include, but not limited to IBM, Qtema, Ericsson, Daimler AG, and Volvo Cars. For more information/ publications, contact http://www.gorschek.com.

**A. K. M. Moinul Islam** is a Software Engineer and Information Technology Security professional. He holds a Bachelor in Computer Science and Engineering from the Rajshahi University of Engineering and Technology, Bangladesh, and a joint Master's degree and MSc in Software Engineering, from the University of Kaiserslautern, Germany, and the Blekinge Institute of Technology, Sweden. Currently, he is doing a part-time Masters in Advanced Security Engineering at FH Joanneum, Austria. Besides his academic achievements, he has working experience of more than 6 years in software, information technology, and telecommunication fields.

**Chow Kian Cheng** is currently a Software Engineer at GE Healthcare, based in Freiburg, Germany. He is responsible for the development of enterprise software solutions for cardiology in the healthcare industry. He holds a joint Master's degree and MSc degree in software engineering, from the Blekinge Institute of Technology, Sweden, and the Free University of Bolzano/Bozen, Italy. He has broad software development experience of more than 10 years in the areas of healthcare, telecommunication, and banking.

**Rahadian Bayu Permadi** received a Bachelor's degree in Informatics from Bandung Institute of Technology, Indonesia. In 2009, he received a double Master's degree in Software Engineering from the Free University of Bolzano/Bozen, Italy, and the Blekinge Institute of Technology, Sweden. Currently, he is working as a Chief Technology Officer of ID Cloud, a cloud computing company based in Indonesia. His interests are lean startup, software development process, cloud computing, and mobile computing.

**Dr Robert Feldt** is a Professor of Software Engineering at Chalmers University of Technology, Sweden, and at Blekinge Institute of Technology, Sweden. He has also worked as an Information Technology and Software Consultant for more than 18 years. His research interests include human-centered software engineering, software testing and verification and validation, automated software engineering, requirements engineering, and user experience. Most of the research is of empirical nature and conducted in close collaboration with industry partners. He received a PhD (Techn. Dr.) in Computer Engineering from Chalmers Universty of Technology in 2002.