Evaluation and Measurement of Software Process Improvement - A Systematic Literature Review

Michael Unterkalmsteiner, *Student Member, IEEE,* Tony Gorschek, *Member, IEEE,* A. K. M. Moinul Islam, Chow Kian Cheng, Rahadian Bayu Permadi, Robert Feldt, *Member, IEEE*

Abstract—BACKGROUND—Software Process Improvement (SPI) is a systematic approach to increase the efficiency and effectiveness of a software development organization and to enhance software products. OBJECTIVE—This paper aims to identify and characterize evaluation strategies and measurements used to assess the impact of different SPI initiatives. METHOD—The systematic literature review includes 148 papers published between 1991 and 2008. The selected papers were classified according to SPI initiative, applied evaluation strategies and measurement perspectives. Potential confounding factors interfering with the evaluation of the improvement effort were assessed. RESULTS—Seven distinct evaluation strategies were identified, whereas the most common one, "Pre-Post Comparison", was applied in 49% of the inspected papers. Quality was the most measured attribute (62%), followed by Cost (41%) and Schedule (18%). Looking at measurement perspectives, "Project" represents the majority with 66%. CONCLUSION—The evaluation validity of SPI initiatives is challenged by the scarce consideration of potential confounding factors, particularly given that "Pre-Post Comparison" was identified as the most common evaluation strategy, and the inaccurate descriptions of the evaluation context. Measurements to assess the short and mid-term impact of SPI initiatives prevail, whereas long-term measurements in terms of customer satisfaction and return on investment tend to be less used.

Index Terms—Process implementation and change, Process measurement, Metrics/Measurement, Systematic Literature Review

1 Introduction

With the increasing importance of software products in industry as well as in our every day's life [62], the process of developing software has gained major attention by software engineering researchers and practitioners in the last three decades [93], [97], [98], [106]. Software processes are human-centered activities and as such prone to unexpected or undesired performance and behaviors [44]. It is generally accepted that software processes need to be continuously assessed and improved in order to fulfill the requirements of the customers and stakeholders of the organization [44]. Software Process Improvement (SPI) encompasses the assessment and improvement of the processes and practices involved in software development [25]. *SPI initiatives* are henceforth referred to activities aimed at improving the software development process (see Section 3.4.3 for a definition of the different types of initiatives).

The measurement of the software process is a substantial component in the endeavor to reach predictable performance and high capability, and to ensure that process artifacts meet their specified quality requirements [41], [219]. As such, software measurement is acknowledged as essential in the improvement of software processes and products since, if the process (or the result) is not measured and evaluated, the SPI effort could address the wrong issue [52].

Software measurement is a necessary component of every SPI program or change effort, and empirical results indicate that measurement is an important factor for the initiatives' success [33], [47]. The feedback gathered by software measurement and the evaluation of the effects of the improvement provide at least two benefits. By making the outcome visible, it motivates and justifies the effort put into the initiative. Furthermore, it enables assessment of SPI strategies and tactics [67]. However, at the same time, it is difficult to establish and implement a measurement program which provides relevant and valid information on which decisions can be based [17], [67]. There is little agreement on what should be measured, and the absence of a systematic and reliable measurement approach is regarded as a factor that contributes to the high failure rate of improvement initiatives [186]. Regardless of these problems in evaluating SPI initiatives, a plethora of evidence exists

- *M. Unterkalmsteiner, T. Gorschek and R. Feldt are with the Software Engineering Research Lab, School of Computing, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden. E-mail: {mun, tgo, rfd}@bth.se.*
- *A. K. M. Moinul Islam is with the Software Engineering: Process and Measurement Research Group, Department of Computer Science, University of Kaiserslautern, PO Box 3049, 67653 Kaiserslautern, Germany. E-mail: moinul.islam@cs.uni-kl.de.*
- *C. K. Cheng is with General Electrics Healthcare, Healthcare IT, Munzinger Straße 5, 79111 Freiburg, Germany. E-mail: ChowKian.Cheng@ge.com.*
- *R. B. Permadi is with Amadeus S. A. S., Product Marketing and Development, 485 Route du Pin Montard, Boite Postale 69, 06902 Sophia Antipolis Cedex, France. E-mail: rahadian-bayu.permadi@amadeus.com.*

Manuscript received May 11, 2010; revised February 8, 2011; accepted February 15, 2011

Digital Object Indentifier 10.1109/TSE.2011.26 0098-5589/11/$26.00 © 2011 IEEE

to show that improvement efforts provide the expected benefits [46], [71], [121], [137], [161], [184], [213], [224], [240], [248].

An interesting question that arises from that is how these benefits are actually assessed. A similar question was raised by Gorschek and Davis [167], where it was criticized how changes / improvements in requirements engineering processes are evaluated for their success. Inspired by the search for dependent variables [167], we conducted a Systematic Literature Review (SLR) to explore how the success of SPI initiatives is determined, and if the approach is different depending on the particular initiative. Furthermore, we investigated which types of measures are used and, based on the categorization by Gorschek and Davis [167], which perspectives (project, product or organization) are used to assess improvement initiatives. Following the idea of Evidence-Based Software Engineering (EBSE) [33] we collect and analyze knowledge from both research and practical experience. To this end we adopted the approach for conducting SLR's proposed by Kitchenham [61].

This paper is organized as follows. Background and related work is presented in Section 2 and our research methodology is presented in Section 3. In Section 4 we describe the results and answer our four major research questions. We present our conclusions in Section 5.

2 Background and Related Work

2.1 Software process improvement

Software process research is motivated by the common assumption that process quality is directly related with the quality of the developed software [28], [44], [62]. The aim of software process improvement is therefore to increase product quality, but also to reduce time-to-market and production costs [28]. The mantra of many software process improvement frameworks and models originates in the Shewhart-Deming cycle [31]: establish an improvement plan, implement the new process, measure the changed process, and analyze the effect of the implemented changes [43], [49], [56], [80].

The Capability Maturity Model (CMM) [76] is an early attempt to guide organizations to increase their software development capability and process maturity [15]. Although software and process measurement is an integral part of the lower maturity levels (repeatable and defined) and central for the managed level [75], the model only *suggests* concrete measurements since the diversity of project environments may evoke varying measurement needs [74]. Similarly, the Capability Maturity Model Integration (CMMI) [2], [113], [114] and ISO / IEC 15504 [35], [111] (also known as SPICE), propose various measurements. The CMMI reference documentation, both for the staged and the continuous representation [113], [114], provides measurement suggestions for each process area as an informative supplement to the required components of the model. The ISO / IEC 15504 standard documentation [112], on the other hand, prescribes that the process improvement has to be confirmed and defines a process measurement framework. The informative part of the ISO standard provides some rather limited examples of process measures without showing how the measurement framework is applied in practice.

A common characteristic of the above-mentioned improvement initiatives is their approach to identify the to-be-improved processes: the actual processes are compared against a set of "best practice" processes. In case of significant divergences, improvement opportunities are identified and the elimination of the differences constitutes the actual process improvement [102]. This approach is commonly referred to as top-down [102] or prescriptive [80] improvement. In conceptual opposition to this idea are the bottom-up [102] or inductive [80] approaches to process improvement. The main principle of bottom-up improvement is a process change driven by the knowledge of the development organization and not by a set of generalized "best practices" [102]. The Quality Improvement Paradigm (QIP) / Experience Factory [7], [8] is one instance in this category of improvement initiatives. As in the prescriptive approaches, measurement to control process change and to confirm goal achievement is a central part of QIP.

2.2 Related work

Gorschek and Davis present a conceptual framework for assessing the impact of requirements process changes [167]. Their central idea is that the effect of a change in the requirements process can be observed and measured at different levels: (1) Effort and quality of requirements related activities and artifacts in the requirements phase, (2) project success in terms of meeting time, budget and scope constraints, (3) product success in terms of meeting both the customers' and the company's expectations, (4) company success in terms of product portfolio and market strategies, and (5) the influence on society.

Although these concepts are described from the perspective of requirements engineering, the essence to evaluate a process change on different levels to understand its impact more thoroughly, is conveyable to software process improvement in general.

By looking at the recent literature one can find several endeavors to systematically collect and analyze the current knowledge in software measurement.

Gomez et al. [48] conducted a SLR on measurement in software engineering. The study considered in total 78 publications and tried to answer three questions: "What to measure?", "How to Measure?' and "When to Measure?" The criteria for inclusion in the review were that the publication presents current and useful measurements. To answer the first question, the study accumulated the metrics based on entities where the measures are collected and the measured attributes. The most measured entity was "Product" (79%), followed by "Project" (12%) and "Process" (9%), and the most

measured attributes were "Complexity" (19%), "Size" (16%) and "Inheritance" (8%). The second question is answered by identifying metrics that have been validated empirically (46%), theoretically (26%) and both empirically / theoretically (28%). Furthermore the measurement focus, e. g. object-orientation, process, quality, was analyzed. The answer for the third question, when to measure, is presented by mapping the metrics onto the waterfall lifecycle phases. The identified product metrics are found in the design (42%), development (27%), maintenance (14%), testing (12%) and analysis (5%) phase.

Bellini et al. [10] systematically reviewed the literature in twenty Software Engineering and Information Systems journals with the aim of describing current and future trends in software measurement. The study identifies and discusses five key software measurement topics in the the reviewed literature: measurement theory, software metrics, development and identification of metrics, measure collection, and evaluation and analysis of measures. The authors conclude that, besides traditional software measures like code complexity and developer productivity, developments from organizational studies, marketing and human resources management are gaining interest in the area of Software Engineering / Information Systems due to the human-intensive nature of software development. Measures used in practice should be developed based upon a common agreement on the relationship between the empirical object of interest and its mathematical representation. Furthermore, for the practical analysis of measures, a more flexible interpretation of the admissible transformations of measurement scales is advocated.

Kitchenham [60] conducted a systematic mapping study to describe the state-of-the-art in software metrics research. The study assesses 103 papers published between 2000 and 2005 and includes an analysis on their influence (in terms of citation counts) on research. Kitchenham concludes that papers presenting empirical validations of metrics have the highest impact on metrics research although she has also identified several issues with this type of studies. For example, 5 out of 7 papers, which empirically validated the object oriented metrics proposed by Chidamber and Kemerer [26], included Lack of Cohesion (LCOM) in the validation. Kitchenham [60] pointed out that LCOM has been demonstrated theoretically invalid [53] and that continuous attempts to validate LCOM empirically seem therefore futile.

The aim of this SLR differs from the above reviews in two aspects. First, the focus of this review is on measurement of software process improvement initiatives, i. e. what to measure, and is therefore more specific than the reviews of Bellini et al. and Gomez et al. Second, this review investigates also how the measures are used to evaluate and analyze the process improvement. Given our different focus, only 1 ([185]) of our 148 reviewed papers was also covered by Bellini et al. [10]. Gomez et al. [48] did not report the reviewed papers which impedes a coverage assessment with our SLR.

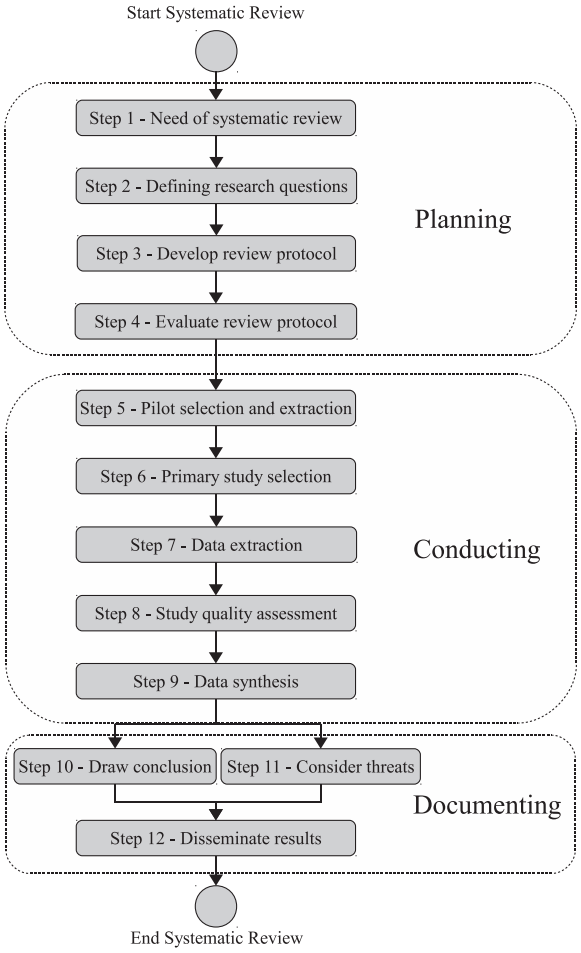

Figure 1. Systematic review steps (adapted from [61])

3 Research Methodology

In this section we describe the design and the execution of the SLR. Furthermore, we discuss threats to the validity of this review. Figure 1 outlines the research process we have used and its steps are described in detail in the following sub-sections.

The *need for this systematic review (Step 1, Figure 1)* was motivated in the introduction of this paper. In order to determine if similar work had already been performed, we searched the Compendex, Inspec and Google Scholar digital libraries[1]. We used the following search string to search within keywords, title and abstracts, using synonyms for "systematic review" defined by Biolchini et al. [30]:

((SPI *OR* "Software process improvement") *AND* ("systematic review" *OR* "research review" *OR* "research synthesis" *OR* "research integration" *OR* "systematic overview" *OR* "systematic research synthesis" *OR* "integrative research review" *OR* "integrative review"))

1. performed on 2008/11/20

Table 1
Research questions for the systematic review

ID	Question	Aim
RQ1	What types of evaluation strategies are used to evaluate SPI initiatives?	To identify which concrete evaluation strategies are used and how they are applied in practice to assess SPI initiatives.
RQ2	What are the reported metrics for evaluating the SPI initiatives?	To identify the metrics which are commonly collected and used to evaluate SPI initiatives.
RQ3	What measurement perspectives are used in the evaluation and to what extent are they associated with the identified SPI initiatives?	To determine from which measurement perspective SPI initiatives are evaluated. Furthermore, to analyze any relationship between SPI initiatives and measurement perspectives.
RQ4	What are the confounding factors in relation to the identified evaluation strategies?	To identify the reported factors that can distort and hence limit the validity of the results of the SPI evaluation. To determine if these issues are addressed and to identify possible remedies.

None of the retrieved publications (see [104]) were related to our objectives which are expressed in the *research questions (Step 2)*. The research questions (Table 1) define what should be extracted from the selected publications (see Section 3.4). For example, RQ1 pertains to how the success (or failure) of SPI initiatives is evaluated, that is, to methods which show the impact of the initiative. Note that with "evaluation strategy" we do not refer to SPI appraisals, such as CBA-IPI [32], SCE [21] or SCAMPI [116], where the organizations maturity is assessed by its conformity to a certain industrial standard [47]. We rather aim to identify the evaluation strategies which are used to effectively show the impact of a process change.

RQ3 investigates the measurement perspectives from which SPI initiatives are evaluated. The perspectives (project, product and organization) are an abstraction based on the identified metrics from RQ2. Finally, RQ4 aims to elicit factors which may impede an accurate evaluation of the initiative.

The aim of the *review protocol (Step 3)* is to reduce potential researcher bias and to permit a replication of the review in the future [61]. The *protocol was evaluated (Step 4)* by an independent researcher with experience in conducting systematic reviews. According to his feedback and our own gathered experiences during the process, we iteratively improved the design of the review. A summary of the final protocol is given in Sections 3.1 to 3.5.

3.1 Search strategy

We followed the process depicted in Figure 2 for the identification of papers. Figure 3 shows the selected databases and the respective number of publications that we retrieved from each.

From our research questions we derived the keywords for the search. The search string is composed by the terms representing the population AND intervention (Table 2).

In order to verify the quality of the search string, we conducted a trial search on Inspec and Compendex.

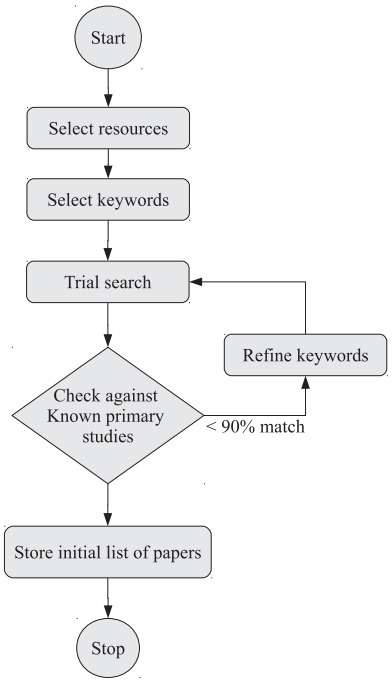

Figure 2. Search strategy

Table 2
Search keywords

Population	Intervention
"process improvement" OR "process enhancement" OR "process innovation"OR SPI	measur* OR metric* OR success* OR evaluat* OR assess* OR roi OR investment* OR value* OR cost* OR effect* OR goal* OR result*

We manually identified relevant publications from the journal "Software Process: Improvement and Practice" (SPIP) and compared them with the result-set of the trial search. The search string captured 24 out of 31 reference publications. Three papers were not in the result-set because Inspec and Compendex did not index at the time of the search[2] issues of SPIP prior to 1998. In order to capture the remaining four publications we added the

2. performed on 2008/11/28

term "result*" to the search string.

Due to the high number of publications we had to handle (10817, see Figure 3) we decided to use a reference management system. We opted for Zotero[3], mainly due to its integrated capability to share and synchronize references.

3.2 Study selection criteria

The main criterion for inclusion as primary study is the presentation of empirical data showing how the discussed SPI initiative is assessed and therefore answering the research questions (Table 1). Both studies conducted in industry and in an academic environment are included. Since the focus of this review is the measurement and evaluation of SPI (see our research questions in Table 1), general discussions on improvement approaches and comparisons of frameworks or models were excluded. For the same reason, descriptions of practices or tools without empirical evaluation of their application were also not considered. Furthermore we excluded reports of "lessons learned" and anecdotal evidence of SPI benefits, books, presentations, posters and non-English texts.

3.3 Study selection procedure

The systematic review *procedure was first piloted (Step 5)* in order to establish a homogeneous interpretation of the selection criteria among the four researchers which conducted the review. The selection criteria were applied on the title and abstract, and if necessary, on the introduction and conclusion of the publication. For the pilot, we assessed individually 50 randomly selected publications from a search conducted in Inspec and Compendex. The Fleiss' Kappa [40] value showed a very low agreement (0.2) among the review team. We conducted a post-mortem analysis to unveil the causes for the poor result. As a main reason we identified the imprecise definition of the selection criteria and research questions, on which the decision for inclusion was mainly based on. After a refinement of these definitions, we conducted a second pilot on 30 randomly selected publications from a search in SCOPUS. Furthermore, we introduced an "Unsure" category to classify publications that should be assessed by all researchers until a consensus was reached. Fleiss' Kappa increased to a moderate agreement (0.5), and, after that the "Unsure" publications were discussed, the inter-rater agreement improved to 0.7 (substantial agreement according to Landis and Koch [65]), which we considered as an acceptable level to start the selection procedure. Figure 3 illustrates in detail how the publications retrieved from the databases were *reduced to the final primary studies (Step 6)* on which we applied the data extraction.

As can be seen in Figure 3, from the 10817 retrieved papers, we first discarded duplicates (by ordering them alphabetically by their title and authors) and studies not published in the English language. After applying the inclusion / exclusion criteria, a total of 6321 papers were found not to be relevant and for 234 publications we were not able to obtain a copy of the text. This diminished the pool of papers for full-text reading to 362 papers. In the final pool of primary studies, 148 papers remained after filtering out studies that we found to be irrelevant after assessing the full-text and those that reported on the same industry case studies.

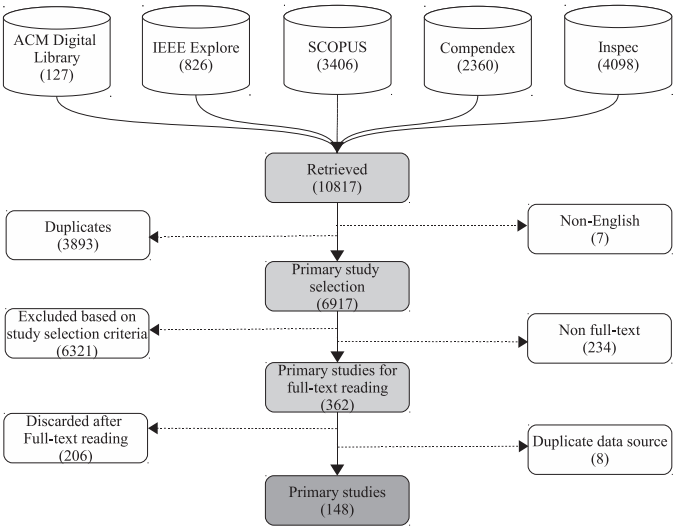

Figure 3. Primary studies selection

3.4 Data extraction

Similarly to the study selection, we distributed the workload among four researchers. The 148 publications accepted for *data extraction (Step 7)* were randomly assigned to the extraction team (37 publications for each member).

We performed the data extraction in an iterative manner. Based on the experiences reported by Staples and Niazi [99], we expected that it would be difficult to establish *a-priori* an exhaustive set of values for all the properties. We therefore prepared an initial extraction form with the properties listed in Table 3, which shows also the mapping to the respective research questions answered by the property. For properties P1, P2, P3 and P5 a list of expected values was established, whereas properties P4, P6 and P7 should be extracted from the studies. Before starting the second iteration, we reviewed the compiled extraction forms in joint meetings and consolidated the extracted data into the categorization given in Sections 3.4.1 to 3.4.7. In a second data extraction iteration we confirmed the established categorization and used it for data synthesis (*Step 9*) and drawing conclusions (*Step 10*).

3.4.1 Research method (P1)

We categorized the studies according to the applied research method. Our initial strategy for the categorization

3. http://www.zotero.org

Table 3
Extracted properties

ID	Property	Research question(s)
P1	Research method	Overview of the studies
P2	Context	Overview of the studies
P3	SPI initiative	RQ1, RQ2, RQ3
P4	Success indicator and metric	RQ2
P5	Measurement perspective	RQ3
P6	Evaluation strategy	RQ1, RQ4
P7	Confounding factors	RQ4

was simple and straightforward: extract the mentioned research method without interpreting the content of the study. However, we discovered two issues with this approach. First, the mentioned research methods were inconsistent, i. e. one study fulfilled our understanding of a "Case study" while another did not. Second, the research method was not mentioned at all in the paper.

Therefore, we defined the following categories and criteria to classify the studies consistently:

- **Case study** if one of the following criteria applies:
  1) The study declares one or more research questions which are answered (completely or partially) by applying a case study [34], [109].
  2) The study empirically evaluates a theoretical concept by applying it in a case study (without necessarily explicitly stating research questions, but having a clearly defined goal [109]).
- **Industry report** if the focus of the study is directed towards reporting industrial experiences without stating research questions or a theoretical concept which is then evaluated empirically. Usually these studies do not mention any research method explicitly. Therefore, instead of creating a category "N/A" (research method is not applicable), we added this category as it complies with the "Project monitoring" method described by Zelkowitz and Wallace [109].
- **Experiment** if the study conducts an experiment [34], [109] and clearly defines its design.
- **Survey** if the study collects quantitative and/or qualitative data by means of a questionnaire or interviews [34], [82], [94].
- **Action research** if the study states this research method explicitly [29], [34].
- **Not stated** if the study does not define the applied research method and it can not be derived or interpreted from reading the paper.

3.4.2 Context (P2)

We categorized the studies into industry and non-industry cases. The industry category contains studies in which the research was performed in collaboration or embedded within industry. The non-industry category is comprised of studies which were performed in an academic setting or for which the research environment was not properly described.

For industrial studies we extracted the company size following the European Recommendation 2003/361/EC [118], the customer type (internal or external to the company) of the developed product, the product type (pure software or embedded), and the application domain. Furthermore, the number of projects in which the SPI initiative was implemented and the staff-size was recorded.

Based on this information, we assessed the study quality from the perspective of the presented research context (see QA4 in Table 5 in Section 3.5).

3.4.3 SPI initiative (P3)

We categorized the studies according to the presented SPI initiative as follows:

- **Framework:** this group contains frameworks/models like CMM, international standards like ISO/IEC 15504 (SPICE) and business management strategies like Six Sigma. For the analysis, we further broke down this category into:
  - Established frameworks - CMM, CMMI, ISO/IEC 15504 (SPICE), Six-Sigma, PSP, TSP, QIP, TQM, IDEAL, PDCA.
  - Combined frameworks - two or more established frameworks are used in combination to implement the SPI initiative.
  - Derived frameworks - an established framework is extended or refined to fulfill the specific needs.
  - Own framework - the study proposes a new framework without reference to one of the established frameworks.
  - Limited framework - the framework targets only a specific process area.
- **Practices:** software engineering practices which can be applied in one or more phases of the software development life-cycle (e.g. inspections, test-driven development, etc.).
- **Tools:** software applications that support software engineering practices.

3.4.4 Success indicator and metric (P4)

From the inspected studies we extracted the metrics which were used to measure the described SPI initiative. In order to get an overview of what is actually measured, the metrics were categorized according to "success indicators". We did not define the classification scheme a-priori but it emerged and evolved during the data extraction (it was stabilized after the first iteration of the data extraction).

We use the term "success indicator" in order to describe the improvement context in which the measurement takes place. Therefore, a "success indicator" is an attribute of an entity (e. g. process, product, organization)

which can be used to evaluate the improvement of that entity. The categories of success indicators is shown in Section 4.3 (Table 8). The identified metrics were categorized as in the following example: (1) The metric "Number of defects found in peer reviews" is mapped to the "Process quality" category as it describes the effectiveness of the peer review process (e. g. [143], [160], [228]). (2) The metric "Number of defects identified after shipment / KLOC" (e. g. [140], [143], [229]) is mapped to the "Product quality" category as the object of study is the product itself and not the processes from which the product originates.

The categorization of the metric is dependent on the context of the study. The use of the metric is interpreted by understanding which attribute is actually measured and with which intention. In some cases this was not possible due to missing information in the description of the metric. For example the "Defects" category contains those defect metrics for which the given information could not be used to justify a classification into one of the predefined quality categories (neither product nor process).

3.4.5 Measurement perspective (P5)

We use the concept of "measurement perspective" to define and categorize how the improvement is being assessed. Concretely, a measurement perspective describes the view on the improvement, i.e. which entities are measured in order to make the change visible in either a quantitative or qualitative manner. We derived from which measurement perspective an initiative is evaluated by interpreting the metrics which were described in the study and from the attributes they are supposed to measure. We defined the following measurement perspectives, based on the five software entity types proposed by Buglione and Abran [20] (the entity types process, project and resources were bundled under the project perspective due to the difficulty to consistently interpret the measures identified in the reviewed studies and to avoid mis-categorization):

- **Project perspective**
  The measurement is conducted during the project where the SPI initiative takes place. Examples of metrics that are used to measure from this perspective are productivity during the development phase, defect rates per development phase, etc. These measures assess the entity types process, project and resources.
- **Product perspective**
  The evaluation of the SPI initiatives' impact is conducted by measuring the effect on the delivered products. An example of a metric that is used to measure from this perspective is the number of customer complaints after product release.
- **Organization perspective**
  The measurement and evaluation of the SPI initiatives' impact is conducted organization-wide. An example of a metric that is used to measure from this perspective is return on investment. Other qualitative measurements such as employee satisfaction and improved business opportunities are also measured from this perspective.

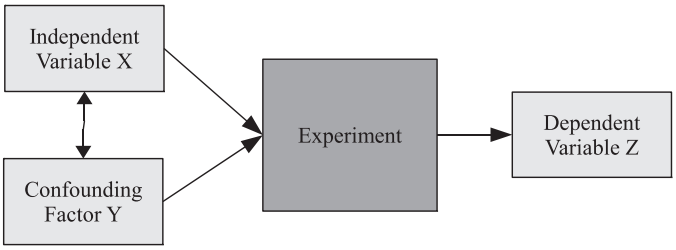

Figure 4. The influence of confounding factors

3.4.6 Evaluation strategy (P6)

During the first iteration of the data extraction we discovered that many publications do not describe or define the adopted evaluation strategy explicitly. To solve this problem, we established a categorization of evaluation strategies based on their common characteristics (see Section 4.2, Table 7). The categorization grew while extracting the data from the studies and was consolidated after the first iteration of the process. In some cases we could not identify an evaluation strategy and the publication was categorized as "Not Stated".

3.4.7 Confounding factors (P7)

In the context of experimental design, Wohlin et al. [108] defined confounding factors as "variables that may affect the dependent variables without the knowledge of the researcher". They represent a threat to the internal validity of the experiment and to the causal inferences that could be drawn since the effect of the treatment cannot be attributed solely to the independent variable. As shown in Figure 4, both independent variables (treatments) and confounding factors represent the input to the experiment and the assessment validity of the dependent variables (effects) is threatened [77].

Assuming that in the evaluation of software process improvements the change is assessed by comparing indicators which represent an attribute before and after the initiative has taken place, it is apparent that the problem of confounding factors, as it is encountered in an experimental setting, is also an issue in the evaluation of SPI initiatives. We argue therefore that it is of paramount importance to identify potential confounding factors in the field of software process improvement.

Kitchenham et al. [63] identified several confounding factors in the context of the the evaluation of software engineering methods and tools through case studies (see Table 4). Similarly, we extracted from the reviewed publications any discussion that addresses the concept of confounding factors in the context of SPI initiatives and, if given, the chosen remedies to control the issues.

Table 4
Confounding factors and remedies (adapted from [63])

Confounding factor	Description	Remedy
Learning bias	The initial effort to acquire the knowledge for using the method or tool interferes with the evaluation of its benefits.	Separate activities aimed at learning the new technology from those aimed at evaluating it.
Participant bias	Attitude of the case study participants towards the new technology (enthusiasm versus skepticism).	Select participants according to a standard staff-allocation method.
Project bias	The projects on which the technology is evaluated differ on application domains, e.g. embedded-systems and information systems.	Select projects within the same application domain.

3.5 Study quality assessment

The *study quality assessment (Step 8)* can be used to guide the interpretation of the synthesis findings and to determine the strength of the elaborated inferences [61]. However, as also experienced by Staples and Niazi [99], we found it difficult to assess to which extent the authors of the studies were actually able to address validity threats. Indeed, the quality assessment we have performed is a judgment of *reporting* rather than *study* quality. We answered the questions given in Table 5 for each publication during the data extraction process.

With QA1 we assessed if the authors of the study clearly state the aims and objectives of the carried out research. This question could be answered positively for all of the reviewed publications. With QA2 we asked if the study provides enough information (either directly or by referencing to the relevant literature) to give the presented research the appropriate context and background. For almost all publications (98%) this could be answered positively. QA3 was checked with "Yes" if validity threats were explicitly discussed, adopting the categorization proposed by Wohlin et al. [107]. The discussion on validity threats of an empirical study increases its credibility [78]. A conscious reflection on potential threats and an explicit reporting of validity threats from the researcher increases the trustworthiness of and the confidence in the reported results. Therefore, if the study just mentioned validity threats without properly explaining how they are identified or addressed, the question was answered with "Partially". The result of QA3 confirms the observation in [98] that in empirical studies the scope of validity is scarcely discussed. QA4 was answered with "Yes" if we could compile the data in the context property of the data extraction form to a major degree (see Section 3.4.2). As it was pointed out by Petersen and Wohlin [79], context has a large impact on the conclusions that are drawn from the evidence in industrial studies. However, 51.7% of the reviewed studies did not, or only partially, describe the context of the research. With QA5 we assessed if the outcome of the research was properly documented. As with QA1, this questions could be answered positively for all (except one) study.

3.6 Validity Threats

We *identified three potential threats to the validity (Step 11)* of the systematic review and its results.

3.6.1 Publication bias

Publication bias refers to the general problem that positive research outcomes are more likely to be published than negative ones [61]. We regard this threat as moderate, since the research questions in this review are not geared towards the performance of a specific software process improvement initiative for the purpose of a comparison. The same reasoning applies to the threat of sponsoring in which certain methods are promoted by influential organizations [61], and negative research outcomes regarding this method are not published. We did not restrict the sources of information to a certain publisher, journal or conference such that it can be assumed that the breadth of the field is covered sufficiently. However, we had to consider the trade-off of considering as much literature as possible and, at the same time, accumulating reliable information. Therefore we decided not to include grey literature (technical reports, work in progress, unpublished or not peer-reviewed publications) [61].

3.6.2 Threats to the identification of primary studies

The strategy to construct the search string aimed to retrieve as many documents as possible related to measurement and evaluation of software process improvements. Therefore, the main metric to decide about the quality of the search string should be the recall of the search result. Recall is expressed as the ratio of the retrieved relevant items and all existing relevant items [92]. Since it is impossible to know all existing relevant items, the recall of the search string was estimated by conducting a pilot search as described in Section 3.1. This showed showed an initial recall of 88%, and after a refinement of the search string, a recall of 100%. Although the search string was exercised on a journal (SPIP) of high relevance for this systematic review, the threat of missing relevant articles still exists. Inconsistent terminology, in particular in software measurement research [45], or use of different terminology with respect to the exercised search string (see Table 2) may have biased the identification of primary studies.

Table 5
Quality assessment

ID	Quality assessment question	Yes	Partially	No
QA1	Is the aim of the research sufficiently explained?	138 (93.2%)	10 (6.8%)	0 (0.0%)
QA2	Is the presented idea/approach clearly explained?	115 (77.7%)	30 (20.3%)	3 (2.0%)
QA3	Are threats to validity taken into consideration?	16 (10.8%)	19 (12.8%)	113 (76.4%)
QA4	Is it clear in which context the research was carried out?	73 (49.3%)	54 (36.5%)	21 (14.2%)
QA5	Are the findings of the research clearly stated?	117 (79.0%)	30 (20.3%)	1 (0.7%)

Precision, on the other hand, expresses how good the search identifies only relevant items. Precision is defined as the ratio of retrieved relevant items and all retrieved items [92]. We did not attempt to optimize the search string for precision. This is clearly reflected by the final, very low, precision of 2.2% (considering 6683 documents after the removal of duplicates and 148 selected primary studies). This is however an expected result since recall and precision are adversary goals, i. e. the optimization to retrieve more relevant items (increase recall) implies usually a retrieval of more irrelevant items too (decrease precision) [86]. The low precision itself represents a moderate threat to the validity of the systematic review since it induced a considerably higher effort in selecting the final primary studies. We addressed this threat as explained in Section 3.6.3.

We followed two additional strategies in order to further decrease the probability of missing relevant papers. First, during the testing of the search string (see Section 3.1), we discovered that the bibliographic databases (Inspec and Compendex) did not index studies published in "Software Process: Improvement and Practice" prior to 1998. Therefore we decided to include a third bibliographic database (SCOPUS) and also individual publishers in the data sources (IEEE Explore and ACM Digital Library). This led to a high number of duplicates (3893) which we could however reliably identify by sorting the documents alphabetically by their title and authors. Secondly, the systematic review design was assessed for completeness and soundness by an independent researcher with experience in conducting systematic literature reviews.

We could not retrieve the full-text for 234 studies within the scheduled time-frame for the systematic review. This however represents a minor threat since this set, recalling the retrieval precision of 2.2%, would have contained approximately only five relevant studies.

3.6.3 Threats to selection and data extraction consistency

Due to the scope of the systematic review, we had to develop efficient (in terms of execution time) and effective (in terms of selection and data extraction consistency) strategies. One of the main aims of defining a review protocol is to reduce researcher bias [61] by defining explicit inclusion/exclusion criteria and a data extraction strategy. A well defined protocol increases the consistency in selection of primary studies and in the following data extraction if the review is conducted by multiple researchers. One approach to further increase the validity of the review results is to conduct selection and data extraction in parallel by several researchers and cross-check the outcome after each phase. In the case of disagreements they should be discussed until a final decision is achieved. Due to the large amount of initially identified studies (10817) we found this strategy impossible to implement within the given time-frame. Therefore, as proposed by Brereton et al. [16] and illustrated in Section 3.3 and 3.4, we piloted the paper selection and data extraction and improved the consensus iteratively. By piloting we addressed two issues: first, the selection criteria and the data extraction form were tested for appropriateness, e. g. are the inclusion / exclusion criteria too restrictive or liberal, should fields be added or removed, are the provided options in the fields exhaustive? Second, the agreement between the researchers could be assessed and discrepancies streamlined, e. g. by increasing the precision of the definitions of terms. Although it can be argued that this strategy is weaker in terms of consistency than the previously mentioned cross-checking approach, it was a necessary trade-off in order to fulfill the schedule and the targeted breadth of the systematic review.

In order to assess data extraction consistency, we performed a second extraction on a randomly selected sample of the included primary studies. Each researcher extracted data from 15 papers, which is slightly more than 10% of the total number of included studies and approximately 50% of the studies each researcher was assigned in the first extraction.

Table 6 shows the Fleiss' Kappa [40] value of each property that was extracted from the primary studies. The inter-rater agreement denotes thereby the data extraction consistency between the researchers. The intra-rater agreement gives an indication of the repeatability of the process (the second extraction was performed eighteen months after the original one).

Landis and Koch [65] propose the following interpretation for Fleiss' Kappa: Almost excellent (1.0 - 0.81), Substantial (0.80 - 0.61), Moderate (0.60 - 0.41), Fair (0.40 - 0.21), Slight (0.20 - 0), and Poor (< 0).

The analysis shown in Table 6 indicates that in prop-

Table 6
Inter- and Intra-rater agreement

Property	Inter-rater	Intra-rater
Research method (P1)	0.56	0.78
Context (P2)	0.90	0.90
SPI initiative (P3)	0.83	0.91
Success indicator (P4)	0.54	0.58
Measurement perspective (P5)	0.06	0.25
Evaluation strategy (P6)	0.77	0.47
Confounding factors (P7)	-0.05	-0.1

erties P5 and P7 we achieved only slight respectively poor agreement in the data extraction validation. A potential reason for this result on property P7 may be that confounding factors are not explicitly mentioned in the selected primary studies and therefore difficult to identify. In rare cases, confounding factors are mentioned in the validity threats of the study (e. g. [140]) or, more frequently, in the results discussion (e. g. [146], [219]). A consistent extraction of property P7 is therefore rather challenging and may be biased.

We agreed however on the identified confounding factors (P7) and the measurement perspective (P5) categorization, as after the original data extraction, all involved researchers jointly discussed the results until a consensus was reached. Hence we are confident that the reported results in Section 4.4 and 4.5 are internally consistent.

4 Results and Analysis

A total of 148 studies discuss the measurement and evaluation of SPI initiatives. Prior to presenting the results and analysis for each research question we give a short overview of the general characteristics of the studies.

4.1 Overview of the studies

4.1.1 Publication year

The reviewed papers were published between 1991 and 2008. A first increased interest in evaluating SPI initiatives appears in the period between 1998 and 2000 (35, 24%). A second spike can be observed between 2005 and 2008 (55, 37%). This seems to indicate an increased interest in SPI and success measurement, pointing to the relevance of the area. In addition, as a substantial part of the publications fall within a period of four years before this review was conducted (2008), it increases the likelihood for the results of the studies being relevant, elevating the potential value obtained in this systematic review.

4.1.2 Research method

The inspected publications were classified according to the applied research methods as defined in Section 3.4.1. Case studies (66, 45%) and industry reports (53, 36%) constitute a clear majority of the studies, followed by experiments (8, 5%), surveys (7, 4%), action research (1, 1%) and a combination of action research and experiment (1, 1%). Also interesting to observe is that the lack of an adequate description of the applied research methodology prevented a categorization (12, 8%).

4.1.3 Study context

The study settings were categorized in industry and non-industry cases (see Section 3.4.2). The majority of the papers (126, 85%) are situated in the industry category, indicating that the results obtained from this review are based on realistic settings.

Remarkably about 50% of the industry studies do not provide any information on the size of the organization where the research was carried out. The fact that considerable research effort exists to explore how to introduce software process improvement into small and medium sized companies [91], [115], suggests that company size and the available resources should be taken into account when choosing and embarking on an SPI initiative. Omitting that information therefore debilitates the judgment if such an initiative is feasible in a different setting [79]. In those studies which reported the organizations size, large (> 250 employees) organizations dominate (34, 27%) over medium (13, 10%) or small (< 50 employees) organizations (13, 10%). Many publications only provide the name of the company but they seldom provide its size in terms of the number of employees. For well-known organizations, this could be due to that the authors consider this information as obvious. Another reason could be that the information was not considered as important to report. Furthermore, confidentiality concerns are not a valid argument for omitting context information since it is possible to anonymize the published data [90]. Indeed there are several reasons why context information such as size, not only of the organization, but also of the unit under study can be considered as crucial. Consider for example "A practical view of software measurement and implementation experiences within Motorola" [143]. The paper does not mention the size of the company. Since Motorola is a well-known company, it is possible to get the information about Motorola's size (at the end of 2008 it had 64000 employees [117]). Even if the organizations' size at the publication date of the study (1992) would be known, it is still difficult to judge the scope of SPI implementation since the paper does not specify the size of, nor in which business units the SPI initiative was implemented.

In order to improve context documentation, future SPI research should consider to adopt the guidelines developed by Petersen and Wohlin [79].

4.1.4 Identified SPI initiatives

Figure 5 shows the distribution of the SPI initiatives according to the definition given in Section 3.4.3. A de-

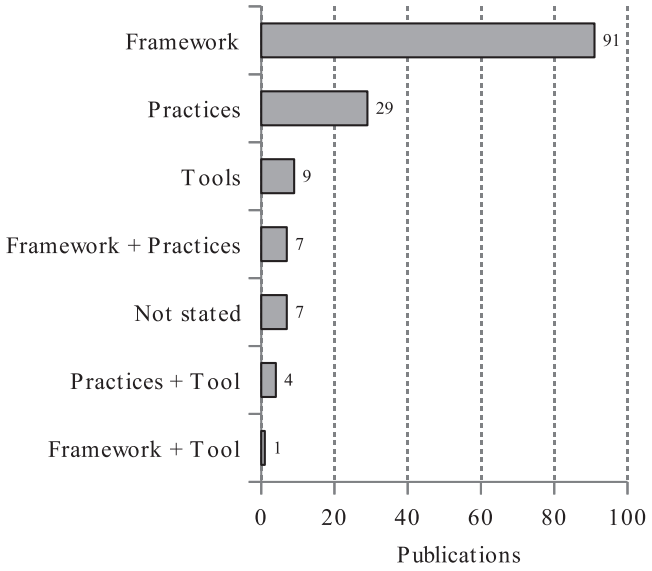

Figure 5. SPI initiative distribution of the publications

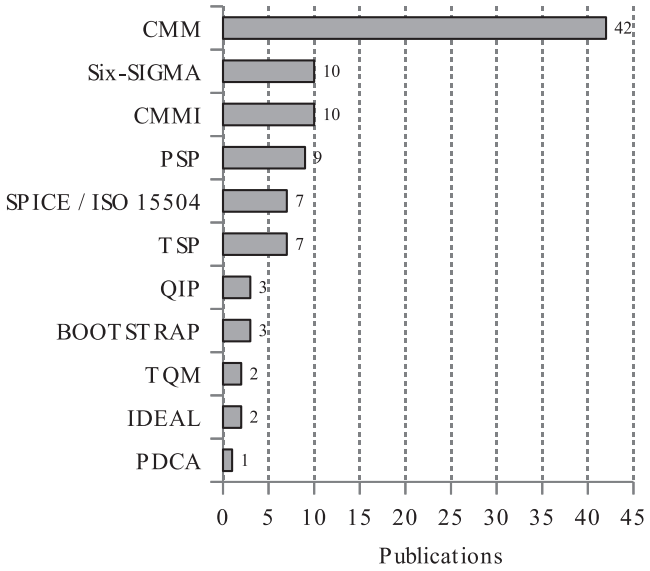

Figure 6. Established framework distribution of the publications

tailed list of all identified initiatives can be found in the extended material of the systematic review (see [104]). Combinations of SPI initiatives (e.g. a certain practice was applied in the context of a framework) are recorded explicitly. The "Framework" category is predominant (91, 61%), followed by "Practices" (29, 20%) and "Tools" (9, 6%).

The scope of this systematic review is to capture any kind of process improvement initiative and their respective approaches to evaluate it. The holistic approach is captured by the "Framework" category while the initiatives targeted at a limited or specific area of software development are represented by the "Practices" and "Tools" categories. Adding up the latter categories (i. e. the categories "Practices", "Tools" and "Practices + Tool" sum up to 42) shows that compared to frameworks (91 studies), they are underrepresented. This suggests that it is less common to measure and evaluate the impact of practices and tools in the context of software process improvement research.

Figure 6 shows the distribution of the established frameworks. It is of no surprise that CMM is the most reported framework (42, 44%) since it was introduced almost 20 years ago. The influence of the Software Engineering Institute (SEI) can be seen here, which is also the sponsor of the CMMI, Team and Personal Software Process (TSP/PSP) and IDEAL. SPICE (ISO/IEC 15504) and BOOTSTRAP, software process improvement and assessment proposals originating in Europe, are rather underrepresented. We extracted the geographic location from the papers where the authors explicitly stated were the study was conducted. Looking at the studies on CMM, North America is represented 15, and Europe 9 times. On the other hand, none of the studies on SPICE were conducted in North America. Considering that of all identified SPI initiatives 27 studies where located in North America and 38 in Europe, this may indicate the existence of a locality principle, i.e. that companies adopt SPI initiatives developed in their geographic vicinity.

However, since the focus of the research questions is to elicit evaluation strategies and measurements in SPI initiatives, the conclusion that SPICE is generally less commonly used in industry cannot be drawn from the picture; it rather means that the evaluation strategies and measurements used in SPICE are less frequently reported by the scientific literature.

In the following sections we answer the research questions stated in Table 1, Section 3.

4.2 Types of evaluation strategies used to evaluate SPI initiatives (RQ1)

4.2.1 Results

The purpose of this research question was to identify the evaluation strategies that are applied to assess the impact of an SPI initiative. As stated in Section 3.4.6, we categorized the strategies according to their common characteristics and established seven categories (see Table 7). The strategies are discussed in more detail in Section 4.2.2. The predominant evaluation strategy that we identified was "Pre-Post Comparison" (72, 49%), followed by "Statistical Analysis" (23, 15%). We encountered also papers where we could not identify an evaluation strategy (21, 14%). They were however included in the review as they provided data points relevant to the other research questions.

4.2.2 Analysis and Discussion

"Pre-Post Comparison" is the most common evaluation strategy. However, the validity of this strategy, in terms

Table 7
Evaluation strategies

Name	Studies	Frequency
Pre-Post Comparison	[119], [120], [122], [124], [125], [128], [129], [133], [135], [138], [139], [142], [144], [145], [147], [150], [152], [154]–[156], [160], [165], [168]–[170], [173], [179], [180], [182], [183], [185], [187], [188], [190]–[192], [195]–[197], [200], [201], [204], [209], [210], [215], [216], [219], [221], [225], [226], [228]–[230], [237]–[241], [244]–[247], [254], [256], [260]–[266]	72
Statistical Analysis	[127], [153], [158], [162], [164], [172], [174]–[176], [179], [211], [214], [222]–[224], [233], [234], [236], [242], [251], [253], [257], [259]	23
Pre-Post Comparison & Survey	[140], [141], [161], [181], [186], [189], [202], [207], [218], [232]	10
Statistical Process Control	[143], [163], [178], [203], [212], [213], [231], [255]	8
Cost-Benefit Analysis	[132], [151], [171], [227], [248]	5
Statistical Analysis & Survey	[217], [258]	2
Philip Crosby Associates' Approach	[148], [149]	2
Pre-Post Comparison & Cost-Benefit Analysis	[146], [243]	2
Survey	[235]	1
Software Productivity Analysis Method	[134]	1
Cost-Benefit Analysis & Survey	[252]	1
Not stated	[121], [123], [126], [131], [136], [137], [157], [159], [166], [167], [177], [184], [193], [194], [198], [205], [206], [208], [220], [249], [250]	21

of whether the assessed results are in causal relationship with the SPI initiative, is rarely discussed (see Section 3.4.7 for a more detailed discussion).

Most of the identified evaluation strategies are not specifically designed for evaluating the outcome of SPI initiatives. However, an exception is given by the Philip Crosby Associates' Approach, which suggests explicitly what to evaluate [27]. The majority of the found evaluation strategies are very generic in nature and different organizations applied those methods for measuring different success indicators based on the organizational needs and contexts. This indicates that there is a shortcoming in the used methods to evaluate the outcome of SPI initiative in a consistent and appropriate way, and supports the demand [186] for a comprehensive measurement framework for SPI.

Pre-Post Comparison: The outcome of SPI initiatives is evaluated by comparing the success indicators' values before and after the SPI initiatives took place. Hence, for the "Pre-Post Comparison" of success indicators it is necessary to setup a baseline from which the improvements can be measured [89]. The major difficulty here is to identify reasonable baseline values. One strategy could be to use the values from a very successful project or product (either internal or external to the organization) and benchmark the improvement against those. Accordingly, the baseline would represent the target that is aimed for in the improvement. Benchmarking in this way is useful if no historical data of successful projects or products is available. However, the performance of the improvement initiative cannot be deduced by comparing against a target baseline since the previous status is unknown and therefore the target may merely serve as an indication. Therefore, for evaluating the effect of improvement initiatives, historical data against which the actual performance can be compared is essential. An example that illustrates how a baseline for organizational performance can be constructed is given by Paulish and Carleton [219]. Organizations with an established measurement program will have less difficulty to establish a baseline than organizations with a newly instantiated or even not yet started program [219].

Baselines are also essential in statistical process control (SPC) where the variation of a specific process attribute relative to a baseline is interpreted as instability and therefore a possible cause of quality issues of the resulting product. Hollenbach and Smith [178] exemplify the establishment of baselines for SPC. Furthermore, the statistical techniques presented by Henry et al. [176] can be used to create baselines of quality and productivity measurements.

Statistical Analysis and Statistical Process Control (SPC): Statistical analysis includes descriptive statistics where data are summarized numerically (e. g. mean, median, mode) or graphically (e. g. charts and graphs). Statistical analysis can also be done by inferential statis-

tics by drawing inferences about the larger population through hypothesis testing, estimates of numerical characteristics (estimation), descriptions of association (correlation), or modeling of relationships (regression). One application of statistical techniques is to strengthen the validity of the collected measurements [88]. Another common application is found in SPC which aim is to measure and analyze the variation in processes. Time series analysis, as promoted by SPC, can provide information when an improvement should be carried out and determine the efficacy of the process changes [22].

As proposed by Henry et al. [176], several statistical techniques can be applied to evaluate the effectiveness of software process improvement in terms of increased estimation accuracy, product quality and customer satisfaction. The described methods are multiple regression, rank correlation and chi-square tests of independence in two-way contingency tables, which, when applied repeatedly over time can show the effectiveness of process improvements statistically [176]. However, care must be taken when applying these techniques since a single method alone may not show the true impact of the initiative and wrong conclusions could be drawn [176]. Furthermore Henry et al. [176] objected that in some cases the process improvement must be very effective in order to show significant alterations in the statistical evaluation results. Statistical methods are also used to assess process stability which is regarded as an important aspect of organizational capability [234]. In order to evaluate stability, the authors propose trend, change and shape metrics which can be used in the short- and long-term and are analyzed by visual inspection of the data summarized by descriptive statistics (e. g. histograms and trend diagrams).

Ramil and Lehman [223] discuss the assessment of process improvement from the viewpoint of software evolution. The authors propose a statistical technique to determine whether productivity (or any other process or product attribute) changes significantly over a long period of time. The aim of the presented CUSUM (cumulative sum) test is to systematically explore data points which highlight changes in the evolutionary behavior. Although this can also be done by visual inspection of trends (as it was proposed by Schneidewind [234]), a change detection algorithm is considered as less error-prone and is particularly useful when assessing the impact of process improvement initiatives and when analyzing whether the performance of processes has changed [223].

An interesting approach to address the issue of certain confounding factors using statistical techniques is presented by Schalken et al. [233]. The authors illustrate how Cost-Model Comparison, based on a linear regression equation, can account for the factor of project size when evaluating the effect of a process improvement on productivity (the same method is also proposed by Alagarsamy et al. [124]). A second issue, namely the comparison of projects from different departments to assess productivity improvement is addressed by the Hierarchical Model Approach. Projects originating from different departments in an organization are not directly comparable since they are either specialized on a group of products, a specific technology or have employees with different skills [233]. Both the Cost-Model Comparison and the Hierarchical Model Approach can be used to prevent erroneous conclusions about the impact of the process improvement initiative by considering context. Unfortunately, as we have shown in Section 4.1.3, the context in which the improvement initiatives are evaluated, is seldom presented completely. It is therefore difficult to judge in such cases if the reported improvement can be attributed to the initiative.

Survey: In the context of this work, a survey is defined as any method to collect, compare and evaluate quantitative or qualitative data from human subjects. A survey can be conducted by interviews or questionnaires, targeting employees affected by the process improvement initiative or customers of the organization. Surveys can be an effective mean to assess the changes introduced in an improvement effort since after all, the development of software is a human-intensive task. The feedback provided by employees can therefore be used to improve the understanding of the effects caused by the introduced changes and to steer future improvements. Gathering information from customers, on the other hand, can provide insight how the improvement affects the quality of products or services as perceived by their respective users. This can be valuable to assess external quality characteristics, such as integrity, reliability, usability, correctness, efficiency and interoperability [39], which otherwise would be difficult to evaluate. The analysis of the improvement participants' feedback can be valuable if historical data for comparison is not available or if its quality / completeness limits the evaluability of the improvement. A systematic method to assess the effects caused by an improvement initiative is described by Pettersson [80]. The approach can be useful if no or only limited historical data is available to construct a baseline which can serve as a reference point for the improvement evaluation. The post-evaluation is based on the expert opinion of the directly involved personnel which compares the improved process with the previous one. This lightweight process improves the visibility on the effects of the undertaken improvement initiative and provides also information on how the change was experienced by the involved roles. The method could be enhanced by integrating the concept of "contribution percentages" as it was proposed by van Solingen [252]. The idea is to let the experts assess how much the initiative actually contributed to the improvement, i. e. provide the possibility to express that only a fraction of the change is attributable to the initiative and other factors have also contributed to the enhancement. Such an approach could also support the identification of potential confounding factors (see Section 4.5).

Besides by the expert opinion of employees, it is also

possible to evaluate the effects of the improvement by querying customers. Quality of service surveys could be sent periodically to customers, illustrating the effects of the adapted or new process from the customer perspective [207].

Cost-Benefit Analysis: Evaluating an improvement initiative with a cost-benefit measure is important since the allocated budget for the program must be justifiable in order not to risk its continuation [62], [252]. Furthermore, it is necessary to avoid loss of money and to identify the most efficient investment opportunities [252]. When assessing cost, organizations should also consider other resources than pure effort (which can be relatively easily measured), e. g. office space, travel, computer infrastructure [252], training, coaching, additional metrics, additional management activities, process maintenance [151]. Activity Based Costing helps to relate certain activities with the actual spent effort [151]. Since cost and effort data can be collected in projects, they must not be estimated [151]. On the other hand, the thereby obtained values are still an approximation and estimations of both costs and benefits are inevitable [252]. Since it is usually enough to know the ROI's relative value (positive, balanced or negative), perfect accuracy is not required as long as the involved stakeholders agree on the procedure how to assess it [252]. Direct benefits and especially indirect and intangible benefits are best assessed by multiple stakeholders [252]; some of the difficult to quantify benefits are: customer satisfaction, improved market share due to improved quality, reduced time-to-deliver and accuracy, feature-cost reduction, opportunity costs, reduced maintenance in follow-up projects, better reusability, employee satisfaction, increased resource availability [151]. A useful technique to support the estimation is the so-called "what-if-not" analysis [252]. Project managers could be asked to estimate how much effort was saved due to the implemented improvement in follow-up projects. The saved effort would then be accounted as a benefit. Another strategy would be to estimate the "worth" of a certain improvement, e. g. asking managers how many training days would they invest to increase employee motivation and quantify the cost of such a training program [252].

Philip Crosby Associates' Approach: This method is derived from Philip Crosby's *Cost of Quality* idea [27]. It is based on distinguishing the cost of doing it right the first time (performance costs) from the cost of rework (non-conformance costs). The cost of quality is determined by the sum of appraisal, prevention and rework costs [148]. The improvement is evaluated by a reduction of rework costs over a longer period of time (several years, as shown in [148] and [149]). This method is similar to Cost-Benefit Analysis but particularly tailored to software process improvement evaluation.

Software Productivity Analysis Method (SPAM): SPAM provides a way of defining productivity models and evaluation algorithms to calculate the productivity of all possible combinations of an observed phenomenon (process, project size, technology etc.) [18], [134].

4.3 Reported metrics for evaluating the SPI initiatives (RQ2)

4.3.1 Results

The purpose of this research was to identify the used metrics and success indicators (see Section 3.4.4) in SPI evaluations.

Table 8 and Table 9 show the frequency of the identified success indicators in the inspected studies. "Process Quality" (57, 39%) was the most observed success indicator, followed by "Estimation Accuracy" (56, 38%), "Productivity" (52, 35%) and "Product Quality" (in total 47 papers, 32%, considering also those from Table 10).

We differentiated the "Product Quality" success indicators based on the ISO 9126-1 standard. The identified studies are shown in Table 10. Two points have to be noted. First, we added "Reusability", which is not defined as a product quality attribute by ISO 9126-1, to the quality attributes. Furthermore, if the study did not explicitly state, or sufficiently describe, which quality attribute is measured, we mapped the study to the general "Product Quality" category (see Table 8).

"Reliability" was the most observed success indicator for the product quality characteristics, followed by "Maintainability" and "Reusability".

Table 11 shows the categorization of estimation accuracy indicators. The "Others" category contains again estimation accuracy metrics which could not be mapped to the specific categories. "Schedule" (37, 25%) is by far the most observed success indicator for estimation accuracy. On the other hand, assuming that "Cost" can be expressed in terms of "Effort" and vice versa, combining them shows that their number of observations (35, 24%) is comparable to that one of "Schedule". "Size" (10, 7%), "Productivity" and "Quality" (2 papers each, 1%) fall behind.

We also distinguished how customer satisfaction is assessed (Table 9). Qualitative customer satisfaction is largely assessed by questionnaires, while quantitative customer satisfaction is recorded by objective measures (e.g. New open problems = total new post-release problems opened during the month).

The "Other Qualitative/Quantitative Success Indicator" categories contain indicators such as "Team morale", "Employee motivation" or "Innovation" which were explicitly mentioned in the studies as indicators for improvement but could not be mapped into the classification.

4.3.2 Analysis and Discussion

The main incentive behind the embarkment of an SPI initiative is to increase quality and to decrease cost and schedule [85], [100], [105]. In order to evaluate the success of such an initiative it is crucial to assess the improvement's effects. Table 8 and Table 9 list the success indicators we identified in this systematic review. We

Table 8
Success indicators

Success indicator	Description	Studies	Frequency
Process Quality	The quality that indicates process performance and is not related to the product. The metrics for this success indicator are dependent on the type of process they measure.	[120], [127]–[129], [131], [137], [138], [141]–[143], [146], [152], [155], [159], [160], [163], [165], [167], [168], [175], [176], [181], [187], [192], [195], [197]–[199], [201]–[204], [211]–[214], [218], [224], [228], [231], [232], [234], [237], [240], [241], [248], [249], [251], [253], [256], [257], [259]–[261], [265], [266]	57
Estimation Accuracy	The deviation between the actual and planned values of other attributes measurements. Examples of attributes commonly estimated are schedule, effort, size and productivity.	[119]–[122], [125], [136], [139]–[141], [143], [144], [149], [152]–[154], [157], [159]–[161], [163], [167], [169], [170], [172], [176]–[178], [183], [185]–[187], [189], [191], [194], [195], [204]–[206], [211], [219], [221], [222], [225]–[227], [229], [237], [238], [240], [243], [255]–[257], [262], [265], [266]	56
Productivity	The performance of the development team in terms of its efficiency in delivering the required output.	[122]–[124], [127], [130], [134], [135], [139]–[141], [143], [146], [149], [157], [161], [163], [168]–[170], [172], [177], [182], [183], [185]–[187], [191], [196], [199], [200], [204]–[206], [208], [216], [217], [219], [220], [222], [223], [227], [229], [233], [238], [239], [245], [247], [250], [255], [262], [264], [265]	52
Product Quality	This list shows studies in which we identified measures for product quality in general. Studies, in which a quality attribute according to the ISO 9126-1 standard was mentioned, are shown in Table 10.	[119], [120], [127], [140], [147], [152], [154], [158], [161], [166], [170], [177], [196], [197], [200], [202], [206], [217], [218], [225], [229], [237], [244], [245], [247], [256], [262], [263]	47 (28 + Table 10
Effort	The effort of the development team in developing the product.	[121], [127], [128], [136], [145], [147], [150], [152], [159], [165], [167], [169], [174], [180], [183], [188], [190], [191], [195], [197], [204], [208], [216], [219], [227], [229]–[231], [240], [242], [244], [247]–[249], [251], [253], [256], [257], [259], [261], [265]	41
Defects	This success indicator is to group metrics that are solely intended to measure the defects without relating them to quality.	[121], [135], [139], [143], [144], [150], [152], [156], [159], [162]–[164], [168], [169], [173], [179], [181], [182], [186], [190], [205], [208], [212], [215], [219], [221], [224], [227], [231], [239], [242], [246], [251], [257], [265]	35

mapped the improvement goals of quality, cost and schedule with these success indicators:

- Quality ("Process Quality" & "Product Quality" & "Other Quality Attributes") was found in 92 papers, 62%
- Cost ("Effort" & "Cost") was found in 61 papers, 41%
- Schedule ("Time-to-market") was found in 27 papers, 18%

This shows that quality is the most measured attribute, followed by cost and schedule. Drawing an analogy with the time-cost-performance triangle [4], [59], which reflects that the three properties are interrelated and it is not possible to optimize all three at the same time, the unbalanced number in the identified success indicators suggests that this is also true for what is actually measured in SPI initiatives.

Furthermore, in order to accurately calculate the financial benefits of an SPI initiative, it is necessary to take all three attributes into account [85]. The low occurrence of "Return-on-investment" (22, 15%) as success indicator suggests that it is seldom used to increase the visibility of the improvement efforts. It has been shown, however, that "Return-on-investment" can be used to communicate the results of an SPI initiative to the various stakeholders [87] (see Section 4.2.2, Cost-Benefit Analysis for are more in-depth discussion about "Return-on-Investment").

Product Quality: As shown in Table 10, we categorized success indicators according to ISO 9126-1 product quality attributes. The main incentive to analyze the success indicators from this perspective is that those attributes may have a different weight, depending on the stakeholder. A developer may rate "Maintainability", "Reusability" and "Portability" (internal quality attributes) higher than the products customer. "Reliability", "Usability", "Functionality" and "Efficiency" on the other hand are the external quality attributes of the product which are potentially more important to the customer [12], [101]. The measurement of internal quality attributes can be applied efficiently, with a low error frequency and cost [101]. It is therefore of no surprise that internal attributes are measured more frequently than external ones (see Table 10). Interestingly, "Reliability" is measured far more often as compared to the other three external attributes. This is explained by looking at the

Table 9
Success Indicators (Continued)

Success indicator	Description	Studies	Frequency
Cost	The cost in terms of the resources that is required in developing the product (monetary expenses).	[121], [123], [128], [129], [134], [135], [142], [143], [148], [149], [156], [166], [167], [170], [196], [202], [207], [208], [210], [212], [214], [215], [243], [244], [256], [260], [261]	27
Time-to-Market	The time that it takes to deliver a product to the market from its conception time.	[121], [135], [146], [150], [152], [154]–[156], [166], [179], [180], [188], [191], [196], [202], [205], [206], [208], [209], [219], [238], [240], [249], [258], [260], [262], [266]	27
Other Qualitative Success Indicators	Examples are: staff morale, employee satisfaction, quality awareness	[121], [126], [133], [136], [137], [140], [149], [157], [177], [184], [186], [193], [202], [205], [209], [226], [227], [235], [238], [239], [248], [257], [263]	24
Return-On-Investment	The value quantified by considering the benefit and cost of software process improvement.	[132], [133], [142], [146], [151], [161], [166], [167], [180], [193], [195], [197], [198], [208], [209], [212], [216], [219], [226], [239], [243], [252]	22
Customer Satisfaction (Qualitative)	The level of customer expectation fulfillment by the organization's product and service. Customer satisfaction measurement is divided into two types, qualitative and quantitative.	[121], [157], [158], [161], [166], [168], [177], [185], [186], [189], [194], [202], [207]–[209], [211], [217], [232], [238], [257]	20
Customer Satisfaction (Quantitative)		[129], [143], [145], [183], [207], [211], [262]	7
Other Quantitative Success Indicators	This success indicator is to group metrics that measure context-specific attributes which are not part of any of the above success indicators (e.g. employee satisfaction, innovation)	[159], [171], [202], [232], [242], [258]	6
Other Quality Attributes		[185], [236], [243], [262]	4

used measures in these studies to express "Reliability", which in the majority are based on product failures reported by the customer and therefore relatively easy to collect and evaluate. On the other hand, "Usability" which is considered as difficult to measure [13], [69], [95], is also seldom assessed in the context of process improvement (see Table 10).

Table 10
ISO-9126-1 Product Quality Attributes

Quality attribute	Studies	Frequency (abs/rel)
Reliability	[128], [143], [176], [178], [194], [222], [224], [226], [249]	9/0.47
Maintainability	[163], [174], [188], [194], [214], [242]	6/0.32
Reusability	[128], [136], [214], [220], [238], [246]	6/0.32
Usability	[194], [238]	2/0.10
Portability	[194], [239]	2/0.10
Efficiency	[194]	1/0.05
Functionality	[194]	1/0.05

Customer Satisfaction: Customer satisfaction can be used to determine software quality [84] since it is commonly considered as an accomplishment of quality management [54]. An increased product quality could therefore also be assessed by examining customer satisfaction. Nevertheless, we identified only few papers (20, 14%) which use qualitative means, and even fewer papers (7, 5%) in which quantitative means are described to determine a change in customer satisfaction (see Table 9). Although measuring customer satisfaction by a questionnaire can provide a more complete view on software quality, it is an intrusive measurement that needs the involvement and cooperation of the customer [68]. On the other hand, quantitative measurements as the number of customer reported failures need to be put into relation with other, possibly unknown, variables in order to be a valid measure for software quality. A decrease in product sales, an increased knowledge of the customer on how to circumvent problems or a shift in the user base can all cause a reduction in reported failures, making the measurement of software quality from this angle more complex [70].

Estimation accuracy: In Table 11 the success indicators for estimation accuracy are shown. It is interesting that *estimating* quality seems very uncommon although the improvement of quality is one of the main interests of

Table 11
Estimation Accuracy Success Indicators

Success Indicator	Studies	Frequency (abs/rel)
Schedule	[121], [122], [143], [144], [149], [153], [154], [157], [159], [161], [163], [167], [169], [172], [177], [178], [183], [185], [186], [189], [191], [194], [195], [211], [219], [221], [225], [226], [237], [238], [240], [251], [256], [257], [262], [265], [266]	37/0.66
Cost	[121], [149], [157], [160], [161], [167], [170], [177], [178], [183], [185], [186], [189], [194], [204], [211], [226], [238]	18/0.32
Effort	[119], [120], [136], [139]–[141], [143], [144], [152], [159], [176], [183], [206], [222], [229], [265], [266]	17/0.30
Size	[119], [125], [139], [144], [152], [169], [187], [222], [243], [265]	10/0.18
Others	[183], [195], [237]	3/0.05
Productivity	[222], [237]	2/0.04

SPI initiatives [50], [62], where quality is found to be the most measured success indicator (Table 8). The identified quality estimation metric instances cover process quality, e. g. actual/estimated number of Quality Assurance reviews ([205]) and actual/estimated number of defects removed per development phase ([237]). Quality estimation metrics should be given equal importance as the other estimation metrics as they can be used to assess the stability of the software process. On the other hand, "Schedule" (37, 25%) and "Cost and Effort" (34, 24%) represent the bulk of the estimation accuracy measures. These two factors may be presumed as important constraints during the project planning [59] and are therefore preferably selected for estimation.

Validity of measurements: Overall we extracted an overwhelming list of metric instances from the publications. However, many of the metric instances are actually measuring the same attribute but in different measurement units, e. g. defect density which is measured by taking the number of defects over size, where size can be expressed in either LOC, FP, etc. Even more interesting is that the definition of basic measures deviates considerably. For the success indicator "Productivity" there are examples where the metric was defined as the ratio of effort over size ([206], [220]), and reversely, as the ratio of size over effort ([170], [262]). Another example can be found for the metric "Defect Density", that is interpreted as "Process Quality" ([261]) but classified as "Defect" in [227], [231].

A potential reason for these inconsistencies can be the lack of a by researchers and practitioners acknowledged reference terminology for software measurement [45]. Imprecise terminology can lead to inadequate assessment, comparison and reporting of measurement results and impede learning [51] and therefore improvement. Besides the lack of agreement on measurement terminology and concepts, there exist doubts on the validity of certain measures. The poor definition of measures leads to broad margins of interpretation as, for example, shown by Kaner and Bond [58] for the reliability metric mean time to failure (MTTF). As pointed out by Carbone et al. [24] it is necessary to understand better the abstract concepts behind the measured quantities and to construct precise operational definitions in order to improve the validity of measurements.

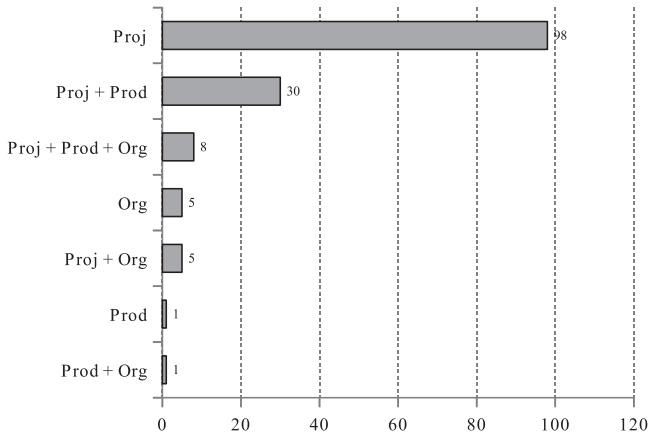

Figure 7. Measurement perspective

4.4 Identified measurement perspectives in the evaluation of SPI initiatives (RQ3)

4.4.1 Results

The purpose of this research question was to assess from which measurement perspective (project, product or organization) SPI initiatives are evaluated (see Section 3.4.5 for the definition of the perspectives). Figure 7 shows the frequencies of the identified measurement perspectives. The "Project" perspective (98, 66%) represents the majority, followed by the "Project and Product" perspective (30, 20%) and the "Project, Product and Organization" perspective (8, 5%). These numbers show that measurement and evaluation at the project level is the most common approach to assess SPI initiatives

The SPI initiatives and the corresponding measurement perspectives are mapped in Table 12 and Table 13 respectively.

We identified the organizational measurement perspective mostly in studies with a CMM-based initiative (row A in Table 12). We did not identify any study with the product perspective alone within the established SPI framework category; however rows A, B, E, F and G in Table 12 show that it is common to combine the project and product perspectives.

Table 12
Measurement perspectives identified in established frameworks

ID	SPI initiatives	Project (Prj)	Product (Prd)	Organization (Org)	Prj & Prd	Prj & Org	Prd & Org	Prj & Prd & Org
A	CMM	[130], [135], [146], [148], [151], [152], [169], [170], [204], [206], [227], [229], [231], [233], [234]	-	[184]	[121], [129], [145], [189], [207], [249], [262]	[186]	-	[177], [193], [219]
B	CMMI	[178], [265]	-	-	[245]	-	-	-
C	SPICE	[157], [183]	-	-	-	-	-	-
D	PSP	[119], [120], [182], [222], [261]	-	-	-	[243]	-	-
E	TSP	[144]	-	-	[237]	-	-	-
F	Six-Sigma	[164], [203]	-	-	[224], [263]	-	-	-
G	QIP	[128]	-	-	-	-	-	-
H	TQM	-	-	-	[238]	-	-	-
I	IDEAL	[200]	-	-	-	-	-	-
J	PDCA	[192]	-	-	-	-	-	-
	Frequency	30	0	1	12	2	0	3

4.4.2 Analysis and Discussion

A considerable amount (98, 66%) of the total 148 papers in this review reported only measurements for the project perspective. This indicates that the measurement perspective to evaluate the SPI initiatives' outcome is strongly biased towards the project perspective. The dominance of project perspective and the very low number of organization perspective may indicate a potential problem to communicate the evaluation results of the SPI initiatives to all the organization's stakeholders, assuming that they have different information needs. On the other hand, it can be argued that measuring the project is easier as probably less confounding factors are involved [167].

At the corporate level, business benefits realized by the improvement initiative need to be visible, whereas the initiatives' impact on a certain project is of more relevance for the involved developers, project or product managers [1]. Hence, it may be beneficial to consider and assess information quality of software measurements in terms of their fitness of purpose [11].

It can also be observed that, whenever the product perspective is considered it is often accompanied by the project perspective. The combination of these measurement perspectives seems reasonable, especially when considering the project success definition by Baccarini [6]: overall project success is the combination of project management success and project product success.

Relying exclusively on the project perspective can raise the difficulty to span the evaluation over several projects, thus not only focusing on attaining goals of a single project [167]. For example, Babar and Gorton [5] have observed in a survey among practitioners that software architecture reviews are often performed in an ad-hoc manner, without a dedicated role or team responsible for the review. As such, this initiative may be beneficial for the current project, but fail to provide the expected financial benefits in the long-term [5]. That would however stay unobserved if the improvement initiative is only evaluated from the project perspective. It is therefore important to assess the effect of SPI initiatives from perspectives beyond the project, that is, consider also the impact on the product and the organization [167].

Looking at Table 12 and rows K to N in Table 13, it can be seen that 77 out of 91 (85%) initiatives that are supported by a framework are evaluated from the project and/or product perspective. This indicates a discrepancy of the initiatives aim, i. e. to establish an organization-wide improvement (e. g. at CMM level 3 the improvement is extended to organizational issues [38]), and how the achievement of this aim is assessed. From the indications gathered in this review, the organizational

Table 13
Measurement perspectives identified in framework variations, practices and tools initiatives

ID	SPI initiatives	Project (Prj)	Product (Prd)	Organization (Org)	Prj & Prd	Prj & Org	Prd & Org	Prj & Prd & Org
K	Two or more SPI frameworks	[160], [187], [188], [195], [221], [228], [235], [240], [241], [255], [264]	-	[132], [252]	[143], [161], [194], [211], [212]	[149], [205]	-	[208], [232]
L	Derived SPI framework	[124], [138], [139], [213], [218], [257]	-	-	-	-	-	[166]
M	Own SPI framework	[137], [244], [248], [258]	-	-	[185], [256]	-	-	-
N	Limited framework	[174], [190], [198], [242], [253], [266]	-	-	[225]	-	[202]	-
O	SPI framework & Practice	[172], [199], [215], [220], [246]	-	-	[179], [196]	-	-	-
P	SPI framework & Tool	-	-	-	[159]	-	-	-
Q	Practices	[122], [123], [131], [141], [142], [153], [154], [156], [163], [173], [175], [181], [191], [201], [214], [216], [223], [230], [251], [259]	-	[133]	[127], [140], [150], [168], [210], [217]	[239]	-	[226]
R	Tool	[125], [126], [134], [155], [165], [180], [254]	[209]	-	[247]	-	-	-
S	Practices & Tool	[136], [147], [197], [260]	-	-	-	-	-	-
T	Not stated	[158], [162], [176], [236], [250]	-	[171]	-	-	-	[167]
	Frequency	68	1	4	18	3	1	5

measurement perspective is the least reported one.

SPI initiatives that involve Six Sigma are mostly focused on the "Project" and "Project & Product" perspective. In 9 out of 10 studies ([164], [203], [224], [263] from Table 12 and [143], [172], [212], [213], [264] from Table 13) these perspectives are considered while only [166] from Table 13 covers the organizational measurement perspective. This could be ascribed to the emphasis given by Six Sigma on product quality [14] and the implied focus on evaluating the impact on the project and on the produced goods.

Finally, if we look at the measurement perspectives identified in the tools and practices category (Table 13, rows Q, R and S), we can identify some interesting patterns. Only [133], [226], [239] consider the organizational measurement perspective. In particular, SPI initiatives in the "Tools" and "Practices + Tools" categories do not consider the organization perspective in the measurement. A potential explanation can be that tools and practices are mostly applied on project or product levels and not on the organization level. For the "Practice" category, the most prominent measurement perspective is the project perspective. The reason is that these initiatives are mostly addressing the project level. The introduction of a tool as an SPI initiative can however have far-reaching consequences, that is for the project [42], [134], but also both for the product quality [57], [63], [66], [73] and the organization [55], [72], [83], [96].

4.5 Confounding factors in evaluating SPI initiatives (RQ4)

4.5.1 Result

The purpose of this research question was to determine which confounding factors (see Section 3.4.7) need to be taken into consideration when evaluating SPI initiatives. As Table 14 shows, we could identify only a few hints regarding these factors. This might indicate that confounding factors are seldom explicitly taken into consideration when evaluating process improvement.

4.5.2 Analysis and Discussion

From the results presented above we can identify several issues regarding confounding factors and their role in evaluating SPI initiatives. The first is that we could only identify 19 studies (out of 148) which discuss potential validity problems when evaluating SPI initiatives. It is therefore difficult to generalize assumptions or to relate a finding to a certain evaluation strategy. Second, the authors of the publications seldom use the term "confounding factor" or "confounding variable"; often we

Table 14
Identified confounding factors

Study	Confounding factors	Proposed solutions proposed in the study
[190]	When choosing projects for evaluation, "it is impossible to find identical projects as they are always of differing size and nature".	Selection of similar projects in size and nature for evaluation.
[178]	Development phase, Measurement unit, Data collection process	Group projects according to project categories and evaluate categories individually.
[183]	In consecutive assessments, projects may have different application domains and scope.	When comparing two evaluation results, select projects with similar application domain and scope.
[219]	Environmental influences like staff size and turnover, capability maturity level, staff morale.	Collect those environmental data (influences) which help to identify and understand influences on performance.
[243]	The result of measuring the impact of personal software process training depends on the length of project time being measured and number of sample data used to measure improvement.	No solution provided.
[165]	The authors identify seven state variables which can influence the result of their study: programming and testing experience of the developers, application domain of the tested component, functional area of the classes involved in the tested component, familiarity of the developers with other tools, scale of the project, size of the project team, and number of iterations previously completed.	Besides the statement that these variables have to be taken into consideration when interpreting the result of their study, no solution is provided.
[146]	Project domain, Project size, Technology changes, code reuse	Project domain - Select similar projects for cycle-time baselining. Project size - Normalize size to "assembly-equivalent lines of code".
[156]	The authors mention uncontrolled independent variables and the Hawthorne effect [9], [23].	Evaluated projects are grouped according to potential confounding factors (cultural differences, skill background of employees) in non-overlapping sets.
[233]	Projects of different size and from different departments.	Linear regression models and hierarchical linear models.
[140]	Organizational changes (management), product maturity, process changes unrelated to the evaluated improvement initiative, and the Hawthorne effect.	No solution provided except a reasoning why these factors are minor threats to the internal validity of the study.
[196]	Staff size, staff training / learning curve, fixed ("overhead") costs as program management, and configuration management and regression testing for multi-platform development.	Staff size - Production rates normalized to staff size. Fixed ("overhead") costs – these cost need to be considered in cost reduction improvement.
[197]	Changes in the environment that might influence the experiment results: company restructuring, change of development platform, changes in product release frequency	No solution provided.
[169]	Project nature (sustainment or new development), manual data collection, different programming languages, and employee education an experience level	Project nature – Group projects according to project categories and evaluate the categories individually.
[186]	"Conflicts about measurement goals can often influence perceptions of success or failure on SPI initiatives".	No solution provided.
[248]	The authors state that measuring the effect of a specific process action on product quality is possible. However, the missing knowledge on relationships between process actions and product quality makes the measurement unreliable and therefore it cannot be generalized to all situations.	No solution provided.
[168]	The authors state that the availability and quality of historical data can affect the result of applying their method of defect-related measurement (BiDefect).	Data from a stable process is required if no high quality historical data is available.
[161]	Several factors that can influence the productivity values such as language, project size, tools and technical issues.	Measuring projects that use same language, tools, development environment and normalizing the productivity by size (function points) can help to reduce the influence of those factors.
[251]	"The preparation rate is known to be a main independent variable for inspection quality."	Measure preparation rates in software inspections and take them into account when evaluating the efficiency and effectiveness of software inspections.
[150]	Development team / Test (QA) team, technology, customer	Staffing, technology and used platform in all projects is similar. Customer is the same organizational division.

had to interpret the descriptions of study designs, executions and results to discover if the authors considered confounding factors. We identified several synonyms instead: "influencing factors" [161], "influences" [169], "state variables" [165], "uncontrolled independent variables" [156] and "environmental influences" [197], [219].

What can be learned from the identified studies is that the identification, characterization and control of confounding factors is a challenging endeavor. In [186], [197], [243], [248] they are described in an abstract and general way without discussing remedies to overcome them. The authors in [248] pointed out that it is possible to measure product quality improvement effected by specific process actions. They also cautioned that it is necessary to study the conditions under which the relationship between process action and improvement are observed in order to increase the knowledge on these relationships. Unfortunately in many cases the context, in which the improvement is evaluated, is described unsatisfactorily (see Section 4.1.3), and an identification of confounding factors is therefore aggravated.

Generally, the effect of confounding factors on the dependent variable can be controlled by designing the study appropriately, e. g. by a random allocation of the treatment and control groups [3]. The fundamental assumption by such a design is that the confounding variables are equally distributed in each group, i. e. that the probability is high that the groups have similar properties. Therefore, if the distribution of the dependent variable is similar in both the control and treatment group, it can be concluded that the treatment has no effect.

The concept of randomization is also discussed in [64], [81], [108] in the context of software engineering experiments. Pfleeger [81], points out that the major difference between experiments and case studies is the degree of control. In order to control a potential confounding variable, the experiment can be designed in such a way that the experimental units within the distinct groups are homogeneous (blocking). Additionally, if the number of experimental units is the same in each group, the design is balanced.

Unfortunately, random sampling of projects or subjects is seldom an option in the evaluation of improvement initiatives and therefore knowing of the existence of potential confounding factors is however needed in order to be able to apply certain techniques to compensate confounding effects [3]. The matching technique, for example, leads to an evaluation design that satisfies the ceteris paribus condition by selecting groups with similar properties with respect to confounding factors [3]. By looking at the proposed solutions, several studies apply some sort of matching, e. g. by selecting similar projects in terms of size and application domain, technology or staff size (see [146], [150], [156], [161], [169], [178], [183], [190] in Table 14).

There exists no systematic way to identify confounding variables [77] and as shown by the examples above, their identification depends on the context in which the study is conducted and on the background knowledge of the researcher. It is therefore difficult to assure that all confounding variables are eliminated or controlled, since their determination relies on assumptions and sound logical reasoning. An interesting discussion about the identification of a confounding factor can be found in the comments by Evanco [37], which refers to the validity of the assumption by El Emam et al. that size is a confounding variable for object oriented metrics [36]. El Emam et al. demonstrate empirically that class size confounds the validity of object oriented metrics as indicators of the fault-proneness of a class. The comments [37], however, show that the identification and recognition of certain confounding factors is still disputed [19], [110].

5 Conclusion

This paper presents a systematic literature review that investigates how the impact of software process improvement initiatives (as defined in Section 3.4.3) is measured and evaluated. The aim is to identify and characterize the different approaches used in realistic settings, i. e. to provide a comprehensive outline and discussion of evaluation strategies and measurements used in the field to assess improvement initiatives. The major findings of this review and their implications for research are:

- **Incomplete context descriptions:** Seventy-five out of 148 studies did not or only partially describe the context in which the study was carried out (see Section 3.5). In the area of process improvement it is however critical to describe the process change and its environment in order to provide results which have the potential to be reused or to be transferred into different settings. Since a considerable body of knowledge on the impact of improvement initiatives is provided by industry reports (53, 36%), a precise and informative context description would be beneficial for both practitioners and researchers.
- **Evaluation validity:** In more than 50% of the studies in which improvement initiatives are evaluated, "Pre-Post Comparison" is used individually or in combination with another method (see Section 4.2). Considering that confounding factors are rarely discussed (19 out of 148 studies, see Section 4.5), the accuracy of the evaluation results can be questioned. The severity of confounding is even increased by unsatisfactory context descriptions. A grounded judgment by the reader on the validity of the evaluation is prohibited by the absence of essential information.
- **Measurement validity:** Kaner and Bond [58] illustrated how important it is to define exactly the semantics of a metric and the pitfalls that arise if it is not commonly agreed what the metric actually means, i. e. which attribute it actually measures. This issue is related with farther reaching questions than

process improvement measurement and evaluation, and concerns fundamental problems of software measurement validity. Nevertheless, measurement definition inconsistencies, as shown in Section 4.3.2, inhibit the process of improvement itself since the comparison and communication of results is aggravated. The implication for research is that it is difficult to identify and use the appropriate measures for improvement evaluation. A better support for defining, selecting and validating measures could enable a comparable and meaningful evaluation of SPI initiatives.

- **Measurement scope:** The analysis on what is actually measured during or after an improvement initiative shows a focus on process and product quality (see Section 4.3). From the software *process* improvement perspective this measurement goal might be adequate and sufficient. It is however crucial to push the event horizon of improvement measurement beyond the level of projects (see Section 4.4) in order to confirm the relatively short-dated measurements at the project or product level. Since the information needs for the different stakeholders vary, appropriate improvement indicators need to be implemented. At the corporate level for example, business benefits realized by projects which encompass a wider scope than pilot improvement implementations are of interest.
  Indicators for these long-term effects can be customer satisfaction, to assess quality improvement, and return on investment to evaluate the economic benefits of improvement. The data presented in this review (see Section 4.3.2) suggests that these indicators tend to be less used in the evaluation of process improvement as other, easier to collect, indicators. The implication for research is to integrate the success indicators into a faceted view on process improvement which captures its short- and long-term impact.
- **Confounding factors:** In a majority (129, 87%) of the reviewed studies we could not identify a discussion on confounding factors that might affect the performance of SPI initiatives and thus their evaluation. Since process improvement affects many aspects of a development project, its results and effect on the organization, there are many potential such confounding factors that threaten validity. Even though study design can often be used to limit the effects it is often not practical to fully control the studied context. Thus future research on SPI should always consider and discuss confounding factors. However, we note that no good conceptual model or framework for such a discussion is currently available.

The results of this review encourage further research on the evaluation of process improvement, particularly on the conception of structured guidelines which support practitioners in the endeavor of measuring, evaluating and communicating the impact of improvement initiatives.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers whose detailed and judicious comments improved the paper considerably. This work was partially funded by the Industrial Excellence Center EASE - Embedded Applications Software Engineering (http://ease.cs.lth.se).

REFERENCES

[1] P. Abrahamsson, "Measuring the success of software process improvement: The dimensions," in *Proceedings European Software Process Improvement (EuroSPI2000) Conference*, Copenhagen, Denmark, 2000. [Online]. Available: http://www.iscn.at/select_newspaper/measurement/oulu.html

[2] D. M. Ahern, R. Turner, and A. Clouse, *CMMI(SM) Distilled: A Practical Introduction to Integrated Process Improvement*. Boston: Addison-Wesley, 2001.

[3] S. Anderson, A. Auquier, W. W. Hauck, D. Oakes, W. Vandaele, and H. I. Weisberg, *Statistical Methods for Comparative Studies: Techniques for Bias Reduction*. New York: John Wiley, 1980.

[4] R. Atkinson, "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*, vol. 17, no. 6, pp. 337–342, Dec. 1999.

[5] M. A. Babar and I. Gorton, "Software architecture review: The state of practice," *Computer*, vol. 42, no. 7, pp. 26–32, Jul. 2009.

[6] D. Baccarini, "The logical framework method for defining project success," *Project Management Journal*, vol. 30, no. 4, pp. 25–32, Dec. 1999.

[7] V. Basili, "The experience factory and its relationship to other improvement paradigms," in *Software Engineering - ESEC '93*, ser. Lecure Notes in Computer Science. London, UK: Springer, 1993, vol. 717, pp. 68–83.

[8] V. Basili and G. Caldiera, "Improve software quality by reusing knowledge and experience," *Sloan Management Review*, vol. 37, no. 1, pp. 55–64, Oct. 1995.

[9] V. Basili and D. Weiss, "A methodology for collecting valid software engineering data," *IEEE Trans. Softw. Eng.*, vol. 10, no. 6, pp. 728–738, Nov. 1984.

[10] C. G. P. Bellini, R. C. F. Pereira, and J. L. Becker, "Measurement in software engineering: From the roadmap to the crossroads," *International Journal of Software Engineering and Knowledge Engineering*, vol. 18, no. 1, pp. 37–64, Feb. 2008.

[11] M. Berry, R. Jeffery, and A. Aurum, "Assessment of software measurement: an information quality study," in *Proceedings 10th International Symposium on Software Metrics (METRICS)*, Chicago, 2004, pp. 314–325.

[12] N. Bevan, "Quality in use: Meeting user needs for quality," *Journal of Systems and Software*, vol. 49, no. 1, pp. 89–96, Dec. 1999.

[13] N. Bevan and M. MacLeod, "Usability measurement in context," *Behaviour & Information Technology*, vol. 13, no. 1, pp. 132–45, 1994.

[14] R. Biehl, "Six sigma for software," *IEEE Softw.*, vol. 21, no. 2, pp. 68–70, Mar. 2004.

[15] B. Boehm, "A view of 20th and 21st century software engineering," in *Proceedings 28th International Conference on Software Engineering (ICSE)*, Shanghai, China, 2006, pp. 12–29.

[16] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

[17] M. Brown and D. Goldenson, "Measurement and analysis: What can and does go wrong?" in *Proceedings 10th International Symposium on Software Metrics (METRICS)*, Chicago, 2004, pp. 131–138.

[18] T. Bruckhaus, "A quantitative approach for analyzing the impact of tools on software productivity," Ph.D. dissertation, McGill University, 1997.

[19] M. Bruntink and A. van Deursen, "An empirical study into class testability," *Journal of Systems and Software*, vol. 79, no. 9, pp. 1219–1232, Sep. 2006.

[20] L. Buglione and A. Abran, "ICEBERG: a different look at software project management," in *Proceedings 12th International Workshop on Software Measurement (IWSM)*, Magdeburg, Germany, 2002, pp. 153–167.

[21] P. Byrnes and M. Phillips, "Software capability evaluation version 3.0 method description," Software Engineering Institute, Carnegie Mellon, Technical Report CMU/SEI-96-TR-002, 1996. [Online]. Available: ftp://ftp.sei.cmu.edu/public/documents/96.reports/pdf/tr002.96.pdf

[22] D. Caivano, "Continuous software process improvement through statistical process control," in *Proceedings 9th European Conference on Software Maintenance and Reengineering (CSMR)*, Manchester, UK, 2005, pp. 288–293.

[23] J. P. Campbell, V. A. Maxey, and W. A. Watson, "Hawthorne effect: Implications for prehospital research," *Annals of Emergency Medicine*, vol. 26, no. 5, pp. 590–594, Nov. 1995.

[24] P. Carbone, L. Buglione, L. Mari, and D. Petri, "A comparison between foundations of metrology and software measurement," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 2, pp. 235–241, Feb. 2008.

[25] D. N. Card, "Research directions in software process improvement," in *Proceedings 28th Annual International Computer Software and Applications Conference (COMPSAC)*, Hong Kong, China, 2004, p. 238.

[26] S. Chidamber and C. Kemerer, "A metrics suite for object oriented design," *IEEE Trans. Softw. Eng.*, vol. 20, no. 6, pp. 476–493, Jun. 1994.

[27] P. B. Crosby, *Quality Without Tears*. New York: McGraw-Hill, 1984.

[28] G. Cugola and C. Ghezzi, "Software processes: a retrospective and a path to the future," *Software Process: Improvement and Practice*, vol. 4, no. 3, pp. 101–123, Sep. 1998.

[29] R. Davison, M. G. Martinsons, and N. Kock, "Principles of canonical action research," *Information Systems Journal*, vol. 14, no. 1, pp. 65–86, Jan. 2004.

[30] J. C. de Almeida Biolchini, P. G. Mian, A. C. C. Natali, T. U. Conte, and G. H. Travassos, "Scientific research ontology to support systematic review in software engineering," *Advanced Engineering Informatics*, vol. 21, no. 2, pp. 133–151, Apr. 2007.

[31] W. E. Deming, *Out of the crisis*. Cambridge: MIT Press, 1986.

[32] D. K. Dunaway and S. Masters, "CMM®-Based appraisal for internal process improvement (CBA IPI) version 1.2 method description," Software Engineering Institute, Carnegie Mellon, Technical Report CMU/SEI-2001-TR-033, 2001. [Online]. Available: http://www.sei.cmu.edu/reports/01tr033.pdf

[33] T. Dyba, "An empirical investigation of the key factors for success in software process improvement," *IEEE Trans. Softw. Eng.*, vol. 31, no. 5, pp. 410–424, May 2005.

[34] S. Easterbrook, J. Singer, M. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to Advanced Empirical Software Engineering*. London, UK: Springer, 2008, pp. 285–311.

[35] K. El Emam, J. Drouin, and W. Melo, *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. Los Alamitos: IEEE Comput. Soc., 1998.

[36] K. El Emam, S. Benlarbi, N. Goel, and S. N. Rai, "The confounding effect of class size on the validity of Object-Oriented metrics," *IEEE Trans. Softw. Eng.*, vol. 27, no. 7, pp. 630–650, Jul. 2001.

[37] W. Evanco, "Comments on "The confounding effect of class size on the validity of object-oriented metrics"," *IEEE Trans. Softw. Eng.*, vol. 29, no. 7, pp. 670–672, Jul. 2003.

[38] B. Fitzgerald and T. O'Kane, "A longitudinal study of software process improvement," *IEEE Softw.*, vol. 16, no. 3, pp. 37–45, May 1999.

[39] R. Fitzpatrick and C. Higgins, "Usable software and its attributes: A synthesis of software quality," in *Proceedings of HCI on People and Computers XIII*, Sheffield, UK, 1998, pp. 3–21.

[40] J. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, Nov. 1971.

[41] W. A. Florac and A. D. Carleton, *Measuring the software process*. Boston: Addison-Wesley, 1999.

[42] D. Flynn, J. Vagner, and O. D. Vecchio, "Is CASE technology improving quality and productivity in software development?" *Logistics Information Management*, vol. 8, no. 2, p. 8–21, 1995.

[43] C. Fox and W. Frakes, "The quality approach: is it delivering?" *Comm. ACM*, vol. 40, no. 6, pp. 24–29, Jun. 1997.

[44] A. Fuggetta, "Software process: a roadmap," in *Proceedings Conference on The Future of Software Engineering*, Limerick, Ireland, 2000, pp. 25–34.

[45] F. Garcia, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruiz, M. Piattini, and M. Genero, "Towards a consistent terminology for software measurement," *Information and Software Technology*, vol. 48, no. 8, pp. 631–644, Aug. 2006.

[46] D. Goldenson and D. Gibson, "Demonstrating the impact and benefits of CMMI: an update and preliminary results," Software Engineering Institute, Tech. Rep. CMU/SEI-2003-SR-009, 2003. [Online]. Available: http://www.sei.cmu.edu/library/abstracts/reports/03sr009.cfm

[47] D. Goldenson, K. E. Emam, J. Herbsleb, and C. Deephouse, "Empirical studies of software process assessment methods," Kaiserslautern: Fraunhofer - Institute for Experimental Software Engineering, Tech. Rep. ISERN-97-09, 1996. [Online]. Available: http://www.ehealthinformation.ca/documents/isern-97-09.pdf

[48] O. Gómez, H. Oktaba, M. Piattini, and F. Garcia, "A systematic review measurement in software engineering: State-of-the-Art in measures," in *Software and Data Technologies*, ser. Communications in Computer and Information Science. Berlin, Germany: Springer, 2008, vol. 10, pp. 165–176.

[49] T. Gorschek and C. Wohlin, "Packaging software process improvement issues: a method and a case study," *Software: Practice and Experience*, vol. 34, no. 14, pp. 1311–1344, Nov. 2004.

[50] E. Gray and W. Smith, "On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement," *Software Quality Journal*, vol. 7, no. 1, pp. 21–34, Mar. 1998.

[51] S. Grimstad, M. Jørgensen, and K. Moløkken-Østvold, "Software effort estimation terminology: The tower of babel," *Information and Software Technology*, vol. 48, no. 4, pp. 302–310, Apr. 2006.

[52] T. Hall, N. Baddoo, and D. Wilson, "Measurement in software process improvement programmes: An empirical study," in *New Approaches in Software Measurement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2001, vol. 2006, pp. 73–82.

[53] M. Hitz and B. Montazeri, "Chidamber and Kemerer's metrics suite: a measurement theory perspective," *IEEE Trans. Softw. Eng.*, vol. 22, no. 4, pp. 267–271, Apr. 1996.

[54] J. Ho-Won, K. Seung-Gweon, and C. Chang-Shin, "Measuring software product quality: a survey of ISO/IEC 9126," *IEEE Softw.*, vol. 21, no. 5, pp. 88–92, Sep. 2004.

[55] W. Humphrey, "CASE planning and the software process," *Journal of Systems Integration*, vol. 1, no. 3, pp. 321–337, Nov. 1991.

[56] W. S. Humphrey, "Introduction to software process improvement," Software Engineering Institute, Tech. Rep. CMU/SEI-92-TR7, 1993. [Online]. Available: ftp://ftp.sei.cmu.edu/public/documents/92.reports/pdf/tr07.92.pdf

[57] S. Jarzabek and R. Huang, "The case for user-centered CASE tools," *Comm. ACM*, vol. 41, no. 8, pp. 93–99, Aug. 1998.

[58] C. Kaner and W. P. Bond, "Software engineering metrics: What do they measure and how do we know," in *Proceedings 10th International Software Metrics Symposium (METRICS)*, Chicago, 2004.

[59] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 10th ed. Hoboken: John Wiley, 2009.

[60] B. Kitchenham, "What's up with software metrics? - a preliminary mapping study," *Journal of Systems and Software*, vol. 83, no. 1, pp. 37–51, Jan. 2010.

[61] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Software Engineering Group, Keele University and Department of Computer Science, University of Durham, United Kingdom, Technical Report EBSE-2007-01, 2007.

[62] B. Kitchenham and S. Pfleeger, "Software quality: the elusive target," *IEEE Softw.*, vol. 13, no. 1, pp. 12–21, Jan. 1996.

[63] B. Kitchenham, L. Pickard, and S. Pfleeger, "Case studies for method and tool evaluation," *IEEE Softw.*, vol. 12, no. 4, pp. 52–62, Jul. 1995.

[64] B. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," *IEEE Trans. Softw. Eng.*, vol. 28, no. 8, pp. 721–734, Aug. 2002.

[65] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, Mar. 1977.

[66] G. Low and V. Leenanuraksa, "Software quality and CASE tools," in *Proceedings on Software Technology and Engineering Practice (STEP)*, Pittsburgh, 1999, pp. 142–150.

[67] L. Mathiassen, O. Ngwenyama, and I. Aaen, "Managing change in software process improvement," *IEEE Softw.*, vol. 22, no. 6, pp. 84–91, Nov. 2005.

[68] J. McColl-Kennedy and U. Schneider, "Measuring customer satisfaction: why, what and how," *Total Quality Management*, vol. 11, no. 7, pp. 883–896, Sep. 2000.

[69] N. McNamara and J. Kirakowski, "Functionality, usability, and user experience: three areas of concern," *Interactions*, vol. 13, no. 6, pp. 26–28, Nov. 2006.

[70] A. Mockus, P. Zhang, and P. L. Li, "Predictors of customer perceived software quality," in *Proceedings 27th International Conference on Software Engineering (ICSE)*, St. Louis, 2005, pp. 225–233.

[71] P. Mohagheghi and R. Conradi, "An empirical investigation of software reuse benefits in a large telecom product," *ACM Transactions on Software Engineering and Methodology*, vol. 17, no. 3, pp. 1–31, Jun. 2008.

[72] W. J. Orlikowski, "CASE tools as organizational change: Investigating incremental and radical changes in systems development," *MIS Quarterly*, vol. 17, no. 3, pp. 309–340, Sep. 1993.

[73] R. Patnayakuni and A. Rai, "Development infrastructure characteristics and process capability," *Comm. ACM*, vol. 45, no. 4, pp. 201–210, Apr. 2002.

[74] M. C. Paulk, C. V. Weber, S. M. Garcia, M. B. Chrissis, and M. Bush, "Key practices of the capability maturity model SM, version 1.1," Software Engineering Institute, Tech. Rep. CMU/SEI-93-TR-025, 1993. [Online]. Available: ftp://ftp.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf

[75] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "Capability maturity model for software version 1.1," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA, Technical Report CMU/SEI-93-TR-024, 1993. [Online]. Available: ftp://ftp.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf

[76] M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Boston: Addison-Wesley, 1995.

[77] J. Pearl, "Why there is no statistical test for confounding, why many think there is, and why they are almost right," *Department of Statistics, UCLA*, Jul. 1998.

[78] D. E. Perry, A. A. Porter, and L. G. Votta, "Empirical studies of software engineering: a roadmap," in *Proceedings Conference on The Future of Software Engineering*, Limerick, Ireland, 2000, pp. 345–355.

[79] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *Proceedings 3rd International Symposium on Empirical Software Engineering and Measurement*, Orlando, 2009, pp. 401–404.

[80] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Öhman, "A practitioner's guide to light weight software process assessment and improvement planning," *The Journal of Systems and Software*, vol. 81, no. 6, pp. 972–995, Jun. 2008.

[81] S. Pfleeger, "Experimentation in software engineering," in *Advances in Computers*. San Diego: Academic Press, 1997, vol. 44, pp. 127–167.

[82] S. L. Pfleeger and B. Kitchenham, "Principles of survey research: part 1: turning lemons into lemonade," *ACM SIGSOFT Software Engineering Notes*, vol. 26, no. 6, pp. 16–18, Nov. 2001.

[83] G. Premkumar and M. Potter, "Adoption of computer aided software engineering (CASE) technology: an innovation adoption perspective," *ACM SIGMIS Database*, vol. 26, no. 2-3, pp. 105–124, May 1995.

[84] R. S. Pressman, *Software engineering: a practitioner's approach*, 5th ed. New York: McGraw-Hill, 2001.

[85] D. Raffo, "The role of process improvement in delivering customer and financial value," in *Portland International Conference on Management and Technology (PICMET)*, Portland, 1997, pp. 589–592.

[86] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Transactions on Information Systems*, vol. 7, no. 3, pp. 205–229, Jul. 1989.

[87] D. F. Rico, *ROI of software process improvement*. Fort Lauderdale: J. Ross Publishing, 2004.

[88] D. J. Rocha, "Strengthening the validity of software process improvement measurements through statistical analysis: A case study at Ericsson AB," http://hdl.handle.net/2077/10529. [Online]. Available: http://hdl.handle.net/2077/10529

[89] J. A. Rozum, "Concepts on measuring the benefits of software process improvements," Software Engineering Institute, Tech. Rep. CMU/SEI-93-TR-009, 1993. [Online]. Available: http://www.sei.cmu.edu/reports/93tr009.pdf

[90] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, Apr. 2009.

[91] G. Santos, M. Montoni, J. Vasconcellos, S. Figueiredo, R. Cabral, C. Cerdeiral, A. Katsurayama, P. Lupo, D. Zanetti, and A. Rocha, "Implementing software process improvement initiatives in small and medium-size enterprises in Brazil," in *6th International Conference on the Quality of Information and Communications Technology (QUATIC)*, Lisbon, Portugal, 2007, pp. 187–198.

[92] T. Saracevic, "Evaluation of evaluation in information retrieval," in *Proceedings 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, 1995, pp. 138–146.

[93] W. Scacchi, "Process models in software engineering," *Encyclopedia of Software Engineering*, pp. 993–1005, 2001.

[94] C. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Trans. Softw. Eng.*, vol. 25, no. 4, pp. 557–72, Jul. 1999.

[95] A. Seffah, M. Donyaee, R. Kline, and H. Padda, "Usability measurement and metrics: A consolidated model," *Software Quality Journal*, vol. 14, no. 2, pp. 159–178, Jun. 2006.

[96] S. Sharma and A. Rai, "CASE deployment in IS organizations," *Comm. ACM*, vol. 43, no. 1, pp. 80–88, Jan. 2000.

[97] M. Shaw, "Prospects for an engineering discipline of software," *IEEE Softw.*, vol. 7, no. 6, pp. 15–24, Nov. 1990.

[98] D. I. K. Sjoberg, T. Dyba, and M. Jorgensen, "The future of empirical methods in software engineering research," in *Future of Software Engineering (FOSE)*, Minneapolis, 2007, pp. 358–378.

[99] M. Staples and M. Niazi, "Experiences using systematic review guidelines," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1425–1437, Sep. 2007.

[100] ——, "Systematic review of organizational motivations for adopting CMM-based SPI," *Information and Software Technology*, vol. 50, no. 7-8, pp. 605–620, Jun. 2008.

[101] D. Stavrinoudis and M. Xenos, "Comparing internal and external software quality measurements," in *Proceedings 2008 Conference on Knowledge-Based Software Engineering*, Pireaus, Greece, 2008, pp. 115–124.

[102] M. Thomas and F. McGarry, "Top-down vs. bottom-up process improvement," *IEEE Softw.*, vol. 11, no. 4, pp. 12–13, Jul. 1994.

[103] J. Trienekens, R. Kusters, and R. van Solingen, "Product focused software process improvement: concepts and experiences from industry," *Software Quality Journal*, vol. 9, no. 4, pp. 269–81, Dec. 2001.

[104] M. Unterkalmsteiner, T. Gorschek, A. K. M. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, "Extended material to "Evaluation and measurement of software process improvement - a systematic literature review"," 2010. [Online]. Available: http://www.bth.se/com/mun.nsf/pages/spi-sysrev-material

[105] R. van Solingen, D. F. Rico, and M. V. Zelkowitz, "Calculating software process improvement's return on investment," in *Advances in Computers*. Elsevier, 2006, vol. 66, pp. 1–41.

[106] N. Wirth, "A brief history of software engineering," *IEEE Ann. Hist. Comput.*, vol. 30, no. 3, pp. 32–39, Jul. 2008.

[107] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering: an introduction*. Norwell: Kluwer Academic Publishers, 2000.

[108] C. Wohlin, M. Höst, and K. Henningsson, "Empirical research methods in software engineering," in *Empirical Methods and Studies in Software Engineering*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2003, vol. 2765, pp. 7–23.

[109] M. V. Zelkowitz and D. Wallace, "Experimental validation in software engineering," *Information and Software Technology*, vol. 39, no. 11, pp. 735–743, 1997.

[110] Y. Zhou, H. Leung, and B. Xu, "Examining the potentially confounding effect of class size on the associations between Object-Oriented metrics and Change-Proneness," *IEEE Trans. Softw. Eng.*, vol. 35, no. 5, pp. 607–623, Sep. 2009.
[111] "ISO/IEC TR2 15504 - software process assessment - part 7: Guide for use in process improvement," ISO, Geneva, Switzerland, Technical Report ISO/IEC TR2 15504, 1998.
[112] "ISO/IEC TR2 15504 - software process assessment: Part 1 - part 9," ISO, Geneva, Switzerland, Technical Report ISO/IEC TR2 15504, 1998.
[113] "Capability maturity model integration (CMMI), version 1.1 (Continuous representation)," Software Engineering Institute, Technical Report CMU/SEI-2002-TR-011, 2002. [Online]. Available: http://www.sei.cmu.edu/library/abstracts/reports/02tr011.cfm
[114] "Capability maturity model integration (CMMI), version 1.1 (Staged representation)," Carnegie Mellon Software Engineering Institute, Pittsburgh, USA, Technical Report CMU/SEI-2002-TR-012, 2002. [Online]. Available: http://www.sei.cmu.edu/library/abstracts/reports/02tr012.cfm
[115] "Improving processes in small settings (IPSS) - a white paper," The International Process Research Consortium (IPRC), Pittsburgh, USA, Tech. Rep., 2006. [Online]. Available: http://www.sei.cmu.edu/iprc/ipss-white-paper-v1-1.pdf
[116] "Appraisal requirements for CMMI, version 1.2 (ARC, v1.2)," Software Engineering Institute, Carnegie Mellon, Pittsburgh, USA, Technical Report CMU/SEI-2006-TR-011, 2006. [Online]. Available: http://www.sei.cmu.edu/library/abstracts/reports/06tr011.cfm
[117] "2008 annual report," Motorola, Inc., Annual Report Motorola, Inc. 2008 Form 10-K, 2009. [Online]. Available: http://investor.motorola.com/annuals.cfm
[118] "Enterprise - SME definition," Aug. 2009. [Online]. Available: http://ec.europa.eu/enterprise/enterprise_policy/sme_definition/index_en.htm

SYSTEMATIC REVIEW REFERENCES

[119] P. Abrahamsson and K. Kautz, "The personal software process: Experiences from Denmark," in *Proceedings 28th Euromicro Conference (EUROMICRO)*, Dortmund, Germany, 2002, pp. 367–74.
[120] ——, "Personal software process: classroom experiences from Finland," in *Software Quality - ESCQ 2002*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2002, vol. 2349, pp. 175–85.
[121] R. Achatz and F. Paulisch, "Industrial strength software and quality: software and engineering at Siemens," in *Proceedings 3rd International Conference on Quality Software (QSIC)*, Dallas, 2003, pp. 321–6.
[122] A. Ahmed, M. Fraz, and F. Zahid, "Some results of experimentation with extreme programming paradigm," in *Proceedings 7th International Multi-Topic Conference (INMIC)*, Lahore, Pakistan, 2004, pp. 387–90.
[123] S. A. Ajila and D. Wu, "Empirical study of the effects of open source adoption on software development economics," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1517–1529, Sep. 2007.
[124] K. Alagarsamy, S. Justus, and K. Iyakutti, "The knowledge based software process improvement program: A rational analysis," in *International Conference on Software Engineering Advances (ICSEA)*, Cap Esterel, France, 2007, p. 61.
[125] B. Anda, E. Angelvik, and K. Ribu, "Improving estimation practices by applying use case models," in *Product Focused Software Process Improvement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2002, vol. 2559, pp. 383–97.
[126] J. Andrade, J. Ares, O. Dieste, R. Garcia, M. Lopez, S. Rodriguez, and L. Verde, "Creation of an automated management software requirements environment: A practical experience," in *Proceedings 10th International Workshop on Database and Expert Systems Applications (DEXA)*, Florence, Italy, 1999, pp. 328–35.
[127] M. T. Baldassarre, A. Bianchi, D. Caivano, and G. Visaggio, "An industrial case study on reuse oriented development," in *Proceedings 21st International Conference on Software Maintenance (ICSM)*, Budapest, Hungary, 2005, pp. 283–92.
[128] V. Basili, M. Zelkowitz, F. McGarry, J. Page, S. Waligora, and R. Pajerski, "SEL's software process improvement program," *IEEE Softw.*, vol. 12, no. 6, pp. 83–7, Nov. 1995.
[129] J. Batista and A. D. de Figueiredo, "SPI in a very small team: a case with CMM," *Software Process Improvement and Practice*, vol. 5, no. 4, pp. 243–50, Dec. 2000.
[130] E. Bellini and C. lo Storto, "CMM implementation and organizational learning: findings from a case study analysis," in *Proceedings PICMET 2006-Technology Management for the Global Future (PICMET)*, Istanbul, Turkey, 2006, pp. 1256–71.
[131] S. Biffl and M. Halling, "Software product improvement with inspection. a large-scale experiment on the influence of inspection processes on defect detection in software requirements documents," in *Proceedings 26th Euromicro Conference*, Maastricht, The Netherlands, 2000, pp. 262–9.
[132] A. Birk, P. Derks, D. Hamann, J. Hirvensalo, M. Oivo, E. Rodenbach, R. van Solingen, and J. Taramaa, "Applications of measurement in product-focused process improvement: a comparative industrial case study," in *Proceedings 5th International Software Metrics Symposium (METRICS)*, Bethesda, 1998, pp. 105–8.
[133] A. Borjesson, "Improve by improving software process improvers," *International Journal of Business Information Systems*, vol. 1, no. 3, pp. 310–38, Jan. 2006.
[134] T. Bruckhaus, N. H. Madhavii, I. Janssen, and J. Henshaw, "The impact of tools on software productivity," *IEEE Softw.*, vol. 13, no. 5, pp. 29–38, Sep. 1996.
[135] C. Buchman, "Software process improvement at AlliedSignal Aerospace," in *Proceedings 29th Hawaii International Conference on System Sciences (HICSS)*, Wailea, 1996, pp. 673–80.
[136] A. Calio, M. Autiero, and G. Bux, "Software process improvement by object technology (ESSI PIE 27785 - SPOT)," in *Proceedings 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, 2000, pp. 641–647.
[137] G. Canfora, F. Garcia, M. Piattini, F. Ruiz, and C. Visaggio, "Applying a framework for the improvement of software process maturity," *Software - Practice and Experience*, vol. 36, no. 3, pp. 283–304, Mar. 2006.
[138] A. Cater-Steel, M. Toleman, and T. Rout, "Process improvement for small firms: An evaluation of the RAPID assessment-based method," *Information and Software Technology*, vol. 48, no. 5, pp. 323–334, May 2006.
[139] G. Cuevas, J. C. Manzano, T. S. Feliu, J. Mejia, M. Munoz, and S. Bayona, "Impact of TSPi on software projects," in *4th Congress of Electronics, Robotics and Automotive Mechanics (CERMA)*, Cuernavaca, Mexico, 2007, pp. 706–711.
[140] D. Damian and J. Chisan, "An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management," *IEEE Trans. Softw. Eng.*, vol. 32, no. 7, pp. 433–453, Jul. 2006.
[141] D. Damian, J. Chisan, L. Vaidyanathasamy, and Y. Pal, "Requirements engineering and downstream software development: Findings from a case study," *Empirical Software Engineering*, vol. 10, no. 3, pp. 255–283, Jul. 2005.
[142] L. Damm and L. Lundberg, "Results from introducing component-level test automation and Test-Driven development," *Journal of Systems and Software*, vol. 79, no. 7, pp. 1001–14, Jul. 2006.
[143] M. K. Daskalantonakis, "A practical view of software measurement and implementation experiences within Motorola," *IEEE Trans. Softw. Eng.*, vol. 18, no. 11, pp. 998–1010, Nov. 1992.
[144] N. Davis, J. Mullaney, and D. Carrington, "Using measurement data in a TSPSM project," in *Software Process Improvement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2004, vol. 3281, pp. 91–101.
[145] C. Debou and A. Kuntzmann-Combelles, "Linking software process improvement to business strategies: experiences from industry," *Software Process Improvement and Practice*, vol. 5, no. 1, pp. 55–64, Mar. 2000.
[146] M. Diaz and J. Sligo, "How software process improvement helped Motorola," *IEEE Softw.*, vol. 14, no. 5, pp. 75–80, Sep. 1997.
[147] J. Dick and E. Woods, "Lessons learned from rigorous system software development," *Information and Software Technology*, vol. 39, no. 8, pp. 551–560, Aug. 1997.
[148] R. Dion, "Elements of a process-improvement program," *IEEE Softw.*, vol. 9, no. 4, pp. 83–5, Jul. 1992.
[149] ——, "Process improvement and the corporate balance sheet," *IEEE Softw.*, vol. 10, no. 4, pp. 28–35, Jul. 1993.

[150] F. Downey and G. Coleman, "Using SPI to achieve delivery objectives in e-commerce software development," *Software Process Improvement and Practice*, vol. 13, no. 4, pp. 327–333, Jul. 2008.
[151] C. Ebert, "The quest for technical controlling," *Software Process Improvement and Practice*, vol. 4, no. 1, pp. 21–31, Mar. 1998.
[152] ——, "Technical controlling and software process improvement," *Journal of Systems and Software*, vol. 46, no. 1, pp. 25–39, Apr. 1999.
[153] ——, "Understanding the product life cycle: four key requirements engineering techniques," *IEEE Softw.*, vol. 23, no. 3, pp. 19–25, May 2006.
[154] ——, "The impacts of software product management," *Journal of Systems and Software*, vol. 80, no. 6, pp. 850–861, Jun. 2007.
[155] C. Ebert and J. D. Man, "e-R&D - effectively managing process diversity," *Annals of Software Engineering*, vol. 14, no. 1-4, pp. 73–91, Dec. 2002.
[156] C. Ebert, C. H. Parro, R. Suttels, and H. Kolarczyk, "Improving validation activities in a global software development," in *Proceedings 23rd International Conference on Software Engineering (ICSE)*, Toronto, Canada, 2001, pp. 545–554.
[157] K. El Emam and A. Birk, "Validating the ISO/IEC 15504 measure of software requirements analysis process capability," *IEEE Trans. Softw. Eng.*, vol. 26, no. 6, pp. 541–66, Jun. 2000.
[158] K. El Emam and N. Madhavji, "Does organizational maturity improve quality?" *IEEE Softw.*, vol. 13, no. 5, pp. 109–10, Sep. 1996.
[159] D. Escala and M. Morisio, "A metric suite for a team PSP," in *Proceedings 5th International Software Metrics Symposium (METRICS)*, Bethesda, 1998, pp. 89–92.
[160] A. Ferreira, G. Santos, R. Cerqueira, M. Montoni, A. Barreto, A. S. Barreto, and A. Rocha, "Applying ISO 9001:2000, MPS.BR and CMMI to achieve software process maturity: BL Informatica's pathway," in *Proceedings 29th International Conference on Software Engineering (ICSE)*, Minneapolis, 2007, pp. 642–651.
[161] A. I. F. Ferreira, G. Santos, R. Cerqueira, M. Montoni, A. Barreto, A. R. Rocha, A. O. S. Barreto, and R. C. Silva, "ROI of software process improvement at BL Informatica: SPIdex is really worth it," *Software Process Improvement and Practice*, vol. 13, no. 4, pp. 311–318, Jul. 2008.
[162] B. Freimut, C. Denger, and M. Ketterer, "An industrial case study of implementing and validating defect classification for process improvement and quality management," in *Proceedings 11th International Software Metrics Symposium (METRICS)*, Como, Italy, 2005, pp. 165–174.
[163] V. French, "Applying software engineering and process improvement to legacy defence system maintenance: an experience report," in *Proceedings 11th International Conference on Software Maintenance (ICSM)*, Opio (Nice), France, 1995, pp. 337–43.
[164] T. Galinac and Z. Car, "Software verification process improvement proposal using six sigma," in *Product Focused Software Process Improvement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, vol. 4589, pp. 51–64.
[165] G. Giraudo and P. Tonella, "Designing and conducting an empirical study on test management automation," *Empirical Software Engineering*, vol. 8, no. 1, pp. 59–81, Mar. 2003.
[166] S. Golubic, "Influence of software development process capability on product quality," in *Proceedings 8th International Conference on Telecommunications (ConTEL)*, Zagreb, Croatia, 2005, pp. 457–63.
[167] T. Gorschek and A. Davis, "Requirements engineering: In search of the dependent variables," *Information and Software Technology*, vol. 50, no. 1-2, pp. 67–75, Jan. 2008.
[168] L. Gou, Q. Wang, J. Yuan, Y. Yang, M. Li, and N. Jiang, "Quantitatively managing defects for iterative projects: An industrial experience report in China," in *Making Globally Distributed Software Development a Success Story*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, vol. 5007, pp. 369–380.
[169] R. Grable, J. Jernigan, C. Pogue, and D. Divis, "Metrics for small projects: Experiences at the SED," *IEEE Softw.*, vol. 16, no. 2, pp. 21–9, Mar. 1999.
[170] T. J. Haley, "Software process improvement at Raytheon," *IEEE Softw.*, vol. 13, no. 6, pp. 33–41, Nov. 1996.
[171] W. Harrison, D. Raffo, J. Settle, and N. Eicklemann, "Technology review: adapting financial measures: making a business case for software process improvement," *Software Quality Journal*, vol. 8, no. 3, pp. 211–30, Nov. 1999.
[172] S. I. Hashmi and J. Baik, "Quantitative process improvement in XP using six sigma tools," in *Proceedings 7th International Conference on Computer and Information Science (ICIS)*, Portland, 2008, pp. 519–524.
[173] J. Haugh, "Never make the same mistake twice-using configuration control and error analysis to improve software quality," in *Proceedings 10th Digital Avionics Systems Conference*, Los Angeles, 1991, pp. 220–5.
[174] J. H. Hayes, N. Mohamed, and T. H. Gao, "Observe-mine-adopt (OMA): an agile way to enhance software maintainability," *Journal of Software Maintenance and Evolution*, vol. 15, no. 5, pp. 297–323, Sep. 2003.
[175] L. He and J. Carver, "PBR vs. checklist: A replication in the n-fold inspection context," in *Proceedings 5th International Symposium on Empirical Software Engineering*, Rio de Janeiro, Brazil, 2006, pp. 95–104.
[176] J. Henry, A. Rossman, and J. Snyder, "Quantitative evaluation of software process improvement," *Journal of Systems and Software*, vol. 28, no. 2, pp. 169–177, Feb. 1995.
[177] J. D. Herbsleb and D. R. Goldenson, "A systematic survey of CMM experience and results," in *Proceedings 18th International Conference on Software Engineering (ICSE)*, Berlin, Germany, 1996, pp. 323–330.
[178] C. Hollenbach and D. Smith, "A portrait of a CMMISM level 4 effort," *Systems Engineering*, vol. 5, no. 1, pp. 52–61, 2002.
[179] C. Hollenbach, R. Young, A. Pflugrad, and D. Smith, "Combining quality and software improvement," *Comm. ACM*, vol. 40, no. 6, pp. 41–5, Jun. 1997.
[180] J. Hössler, O. Kath, M. Soden, M. Born, and S. Saito, "Significant productivity enhancement through model driven techniques: a success story," in *Proceedings 10th International Enterprise Distributed Object Computing Conference (EDOC)*, Hong Kong, China, 2006, pp. 367–373.
[181] M. Höst and C. Johansson, "Evaluation of code review methods through interviews and experimentation," *Journal of Systems & Software*, vol. 52, no. 2-3, pp. 113–20, Jun. 2000.
[182] W. Humphrey, "Using a defined and measured personal software process," *IEEE Softw.*, vol. 13, no. 3, pp. 77–88, May 1996.
[183] S. Hwang and H. Kim, "A study on metrics for supporting the software process improvement based on SPICE," in *Software Engineering Research and Applications*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2005, vol. 3647, pp. 71–80.
[184] K. Hyde and D. Wilson, "Intangible benefits of CMM-based software process improvement," *Software Process Improvement and Practice*, vol. 9, no. 4, pp. 217–28, Oct. 2004.
[185] J. Iversen and L. Mathiassen, "Cultivation and engineering of a software metrics program," *Information Systems Journal*, vol. 13, no. 1, pp. 3–19, Jan. 2003.
[186] J. Iversen and O. Ngwenyama, "Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at Danske Data," *International Journal of Information Management*, vol. 26, no. 1, pp. 30–43, Feb. 2006.
[187] J. Jarvinen and R. van Solingen, "Establishing continuous assessment using measurements," in *Proceedings 1st Internation Conference on Product Focused Software Process Improvement (PROFES)*, Oulu, Finland, 1999, pp. 49–67.
[188] J. Jarvinen, D. Hamann, and R. van Solingen, "On integrating assessment and measurement: towards continuous assessment of software engineering processes," in *Proceedings 6th International Software Metrics Symposium (METRICS)*, Boca Raton, 1999, pp. 22–30.
[189] A. Johnson, "Software process improvement experience in the DP/MIS function," in *Proceedings 16th International Conference on Software Engineering (ICSE)*, Sorrento, Italy, 1994, pp. 323–9.
[190] D. Karlström, P. Runeson, and S. Norden, "A minimal test practice framework for emerging software organizations," *Software Testing, Verification and Reliability*, vol. 15, no. 3, pp. 145–66, Sep. 2005.
[191] K. Kautz, "Making sense of measurement for small organizations," *IEEE Softw.*, vol. 16, no. 2, pp. 14–20, Mar. 1999.
[192] H. Kihara, "Quality assurance activities in the software development center, Hitachi Ltd," in *Proceedings 16th Annual Pacific Northwest Software Quality Conference Joint ASQ Software Division's 8th International Conference on Software Quality*, Portland, 1998, pp. 372–84.
[193] H. Krasner and G. Scott, "Lessons learned from an initiative for improving software process, quality and reliability in a semiconductor equipment company," in *Proceedings 29th Annual*

Hawaii International Conference on System Sciences (HICSS), Maui, 1996, pp. 693–702.

[194] J. Kuilboer and N. Ashrafi, "Software process improvement deployment: an empirical perspective," *Journal of Information Technology Management*, vol. 10, no. 3-4, pp. 35–47, 1999.

[195] A. Kuntzmann-Combelles, "Quantitative approach to software process improvement," in *Objective Software Quality*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 1995, vol. 926, pp. 16–30.

[196] J. A. Lane and D. Zubrow, "Integrating measurement with improvement: An action-oriented approach," in *Proceedings 19th International Conference on Software Engineering (ICSE)*, Boston, 1997, pp. 380–389.

[197] J. Larsen and H. Roald, "Introducing ClearCase as a process improvement experiment," in *System Configuration Management*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 1998, vol. 1439, pp. 1–12.

[198] L. Lazic and N. Mastorakis, "Cost effective software test metrics," *WSEAS Transactions on Computers*, vol. 7, no. 6, pp. 599–619, Jun. 2008.

[199] E. Lee and M. Lee, "Development system security process of ISO/IEC TR 15504 and security considerations for software process improvement," in *Computational Science and its Applications*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2005, vol. 3481, pp. 363–372.

[200] J. W. Lee, S. H. Jung, S. C. Park, Y. J. Lee, and Y. C. Jang, "System based SQA and implementation of SPI for successful projects," in *Proceedings International Conference on Information Reuse and Integration (IRI)*, Las Vegas, 2005, pp. 494–499.

[201] H. Leung, "Improving defect removal effectiveness for software development," in *Proceedings 2nd Euromicro Conference on Software Maintenance and Reengineering (CSMR)*, Florence, Italy, 1998, pp. 157–64.

[202] B. List, R. Bruckner, and J. Kapaun, "Holistic software process performance measurement from the stakeholders' perspective," in *Proceedings 16th International Workshop on Database and Expert Systems Applications (DEXA)*, Copenhagen, Denmark, 2005, pp. 941–7.

[203] D. Macke and T. Galinac, "Optimized software process for fault handling in global software development," in *Making Globally Distributed Software Development a Success Story*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, vol. 5007, pp. 395–406.

[204] F. McGarry, "What is a level 5?" in *Proceedings 26th Annual NASA Goddard Software Engineering Workshop (SEW)*, Greenbelt, 2002, pp. 83–90.

[205] F. McGarry and B. Decker, "Attaining level 5 in CMM process maturity," *IEEE Softw.*, vol. 19, no. 6, pp. 87–96, Nov. 2002.

[206] F. McGarry, S. Burke, and B. Decker, "Measuring the impacts individual process maturity attributes have on software products," in *Proceedings 5th International Software Metrics Symposium (METRICS)*, Bethesda, 1998, pp. 52–60.

[207] K. A. McKeown and E. G. McGuire, "Evaluation of a metrics framework for product and process integrity," in *Proceedings 33rd Hawaii International Conference on System Sciences (HICSS)*, Maui, 2000, p. pp. 4046.

[208] P. Miller, "An SEI process improvement path to software quality," in *Proceedings 6th International Conference on the Quality of Information and Communication Technology (QUATIC)*, Lisbon, Portugal, 2007, pp. 12–18.

[209] J. Momoh and G. Ruhe, "Release planning process improvement - an industrial case study," *Software Process Improvement and Practice*, vol. 11, no. 3, pp. 295–307, May 2006.

[210] S. Morad and T. Kuflik, "Conventional and open source software reuse at Orbotech - an industrial experience," in *Proceedings International Conference on Software - Science, Technology and Engineering (SWSTE)*, Herzlia, Israel, 2005, pp. 110–17.

[211] B. Moreau, C. Lassudrie, B. Nicolas, O. I'Homme, C. d'Anterroches, and G. L. Gall, "Software quality improvement in France Telecom research center," *Software Process Improvement and Practice*, vol. 8, no. 3, pp. 135–44, Jul. 2003.

[212] M. Murugappan and G. Keeni, "Quality improvement-the six sigma way," in *Proceedings 1st Asia-Pacific Conference on Quality Software (APAQS)*, Hong Kong, China, 2000, pp. 248–57.

[213] ——, "Blending CMM and six sigma to meet business goals," *IEEE Softw.*, vol. 20, no. 2, pp. 42–8, Mar. 2003.

[214] K. Nelson and M. Ghods, "Evaluating the contributions of a structured software development and maintenance methodology," *Information Technology & Management*, vol. 3, no. 1-2, pp. 11–23, Jan. 2002.

[215] K. Nelson, M. Buche, and H. Nelson, "Structural change and change advocacy: a study in becoming a software engineering organization," in *Proceedings 34th Annual Hawaii International Conference on System Sciences (HICSS)*, Maui, 2001, p. 9 pp.

[216] T. Nishiyama, K. Ikeda, and T. Niwa, "Technology transfer macro-process. a practical guide for the effective introduction of technology," in *Proceedings 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, 2000, pp. 577–86.

[217] A. Nolan, "Learning from success," *IEEE Softw.*, vol. 16, no. 1, pp. 97–105, Jan. 1999.

[218] S. Otoya and N. Cerpa, "An experience: a small software company attempting to improve its process," in *Proceedings 9th International Workshop Software Technology and Engineering Practice (STEP)*, Pittsburgh, 1999, pp. 153–60.

[219] D. J. Paulish and A. D. Carleton, "Case studies of software-process-improvement measurement," *Computer*, vol. 27, no. 9, pp. 50–57, Sep. 1994.

[220] S. Pfleeger, "Maturity, models, and goals: how to build a metrics plan," *Journal of Systems and Software*, vol. 31, no. 2, pp. 143–55, Nov. 1995.

[221] L. Pracchia, "TheAV-8B team learns synergy of EVM and TSP accelerates software process improvement," *CrossTalk*, no. 1, pp. 20–22, 2004.

[222] L. Prechelt and B. Unger, "An experiment measuring the effects of personal software process (PSP) training," *IEEE Trans. Softw. Eng.*, vol. 27, no. 5, pp. 465–472, May 2001.

[223] J. Ramil and M. Lehman, "Defining and applying metrics in the context of continuing software evolution," in *Proceedings 7th International Software Metrics Symposium (METRICS)*, London, UK, 2000, pp. 199–209.

[224] C. Redzic and J. Baik, "Six sigma approach in software quality improvement," in *Proceedings 4th International Conference on Software Engineering Research, Management and Applications (SERA)*, Seattle, 2006, pp. 396–406.

[225] B. Regnell, P. Beremark, and O. Eklundh, "A market-driven requirements engineering process: results from an industrial process improvement programme," *Requirements Engineering*, vol. 3, no. 2, pp. 121–9, Jun. 1998.

[226] A. Roan and P. Hebrard, "A PIE one year after: APPLY," in *International Conference on Product Focused Software Process Improvement (VTT Symposium)*, Oulu, Finland, 1999, pp. 606–19.

[227] J. Rooijmans, H. Aerts, and M. van Genuchten, "Software quality in consumer electronics products," *IEEE Softw.*, vol. 13, no. 1, pp. 55–64, Jan. 1996.

[228] M. Russ and J. McGregor, "A software development process for small projects," *IEEE Softw.*, vol. 17, no. 5, pp. 96–101, Sep. 2000.

[229] K. Sakamoto, N. Niihara, T. Tanaka, K. Nakakoji, and K. Kishida, "Analysis of software process improvement experience using the project visibility index," in *Proceedings 3rd Asia-Pacific Software Engineering Conference (APSEC)*, Seoul, South Korea, 1996, pp. 139–48.

[230] O. Salo and P. Abrahamsson, "An iterative improvement process for agile software development," *Software Process Improvement and Practice*, vol. 12, no. 1, pp. 81–100, Jan. 2007.

[231] K. Sargut and O. Demirors, "Utilization of statistical process control (SPC) in emergent software organizations: pitfalls and suggestions," *Software Quality Journal*, vol. 14, no. 2, pp. 135–57, Jun. 2006.

[232] E. Savioja and M. Tukiainen, "Measurement practices in financial software industry," *Software Process Improvement and Practice*, vol. 12, no. 6, pp. 585–595, Nov. 2007.

[233] J. Schalken, S. Brinkkemper, and H. van Vliet, "Using linear regression models to analyse the effect of software process improvement," in *Product-Focused Software Process Improvement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2006, vol. 4034, pp. 234–248.

[234] N. Schneidewind, "Measuring and evaluating maintenance process using reliability, risk, and test metrics," *IEEE Trans. Softw. Eng.*, vol. 25, no. 6, pp. 769–81, Nov. 1999.

[235] L. Scott, R. Jeffery, L. Carvalho, J. D'Ambra, and P. Rutherford, "Practical software process improvement - the IMPACT project," in *Proceedings 13th Australian Software Engineering Conference (ASWEC)*, Canberra, Australia, 2001, pp. 182–9.

[236] R. Seacord, J. Elm, W. Goethert, G. Lewis, D. Plakosh, J. Robert, L. Wrage, and M. Lindvall, "Measuring software sustainability," in *Proceedings International Conference on Software Maintenance (ICSM)*, Amsterdam, The Netherlands, 2003, pp. 450–9.

[237] G. Seshagiri and S. Priya, "Walking the talk: building quality into the software quality management tool," in *Proceedings 3rd International Conference on Quality Software (QSIC)*, Dallas, 2003, pp. 67–74.

[238] S. Shah and J. Sutton, "Crafting a TQM-oriented software development lifecycle: program experience," in *Proceedings National Aerospace and Electronics Conference (NEACON)*, Dayton, 1992, pp. 643–9.

[239] B. Shen and D. Ju, "On the measurement of agility in software process," in *Software Process Dynamics and Agility*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, vol. 4470, pp. 25–36.

[240] I. Sommerville and J. Ransom, "An empirical study of industrial requirements engineering process assessment and improvement," *ACM Transactions on Software Engineering and Methodology*, vol. 14, no. 1, pp. 85–117, Jan. 2005.

[241] G. Spork and U. Pichler, "Establishment of a performance driven improvement programme," *Software Process Improvement and Practice*, vol. 13, no. 4, pp. 371–382, Jul. 2008.

[242] L. Suardi, "How to manage your software product life cycle with MAUI," *Comm. ACM*, vol. 47, no. 3, pp. 89–94, Mar. 2004.

[243] T. Lee, D. Baik, and H. In, "Cost benefit analysis of personal software process training program," in *Proceedings 8th International Conference on Computer and Information Technology Workshops (CIT-WORKSHOPS)*, Sydney, Australia, 2008, pp. 631–6.

[244] T. Tanaka, K. Sakamoto, S. Kusumoto, K. Matsumoto, and T. Kikuno, "Improvement of software process by process description and benefit estimation," in *Proceedings 17th International Conference on Software Engineering (ICSE)*, Seattle, 1995, pp. 123–32.

[245] K. Taneike, H. Okada, H. Ishigami, and H. Mukaiyama, "Quality assurance activities for enterprise application software packages," *Fujitsu Scientific and Technical Journal*, vol. 44, no. 2, pp. 106–113, Apr. 2008.

[246] C. Tischer, A. Müller, M. Ketterer, and L. Geyer, "Why does it take that long? Establishing product lines in the automotive domain," in *Proceedings 11th International Software Product Line Conference (SPLC)*, Kyoto, Japan, 2007, pp. 269–274.

[247] F. Titze, "Improvement of a configuration management system," in *Proceedings 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, 2000, pp. 618–25.

[248] J. Trienekens, R. Kusters, and R. van Solingen, "Product focused software process improvement: concepts and experiences from industry," *Software Quality Journal*, vol. 9, no. 4, pp. 269–81, Dec. 2001.

[249] J. Trienekens, R. Kusters, M. van Genuchten, and H. Aerts, "Targets, drivers and metrics in software process improvement: results of a survey in a multinational organization," *Software Quality Journal*, vol. 15, no. 2, pp. 135–53, Jun. 2007.

[250] J. D. Valett, "Practical use of empirical studies for maintenance process improvement," *Empirical Software Engineering*, vol. 2, no. 2, pp. 133–142, Jun. 1997.

[251] M. van Genuchten, C. van Dijk, H. Scholten, and D. Vogel, "Using group support systems for software inspections," *IEEE Softw.*, vol. 18, no. 3, pp. 60–5, May 2001.

[252] R. van Solingen, "Measuring the ROI of software process improvement," *IEEE Softw.*, vol. 21, no. 3, pp. 32–38, May 2004.

[253] G. Visaggio, P. Ardimento, M. Baldassarre, and D. Caivano, "Assessing multiview framework (MF) comprehensibility and efficiency: A replicated experiment," *Information and Software Technology*, vol. 48, no. 5, pp. 313–22, May 2006.

[254] M. Visconti and L. Guzman, "A measurement-based approach for implanting SQA and SCM practices," in *Proceedings 20th International Conference of the Chilean Computer Science Society (SCCC)*, Santiago, Chile, 2000, pp. 126–34.

[255] B. R. V. Konsky and M. Robey, "A case study: GQM and TSP in a software engineering capstone project," in *Proceedings 18th Software Engineering Education Conference (CSEET)*, Ottawa, Canada, 2005, pp. 215–222.

[256] C. von Wangenheim, S. Weber, J. Hauck, and G. Trentin, "Experiences on establishing software processes in small companies," *Information and Software Technology*, vol. 48, no. 9, pp. 890–900, Sep. 2006.

[257] Q. Wang and M. Li, "Measuring and improving software process in China," in *Proceedings International Symposium on Empirical Software Engineering*, Noosa Heads, Australia, 2005, pp. 183–192.

[258] D. Weiss, D. Bennett, J. Payseur, P. Tendick, and P. Zhang, "Goal-oriented software assessment," in *Proceedings 24th International Conference on Software Engineering (ICSE)*, Orlando, 2002, pp. 221–31.

[259] D. Winkler, B. Thurnher, and S. Biffl, "Early software product improvement with sequential inspection sessions: an empirical investigation of inspector capability and learning effects," in *Proceedings 33rd Euromicro Conference on Software Engineering and Advanced Applications (EUROMICRO)*, Lübeck, Germany, 2007, pp. 245–54.

[260] M. Winokur, A. Grinman, I. Yosha, and R. Gallant, "Measuring the effectiveness of introducing new methods in the software development process," in *Proceedings 24th EUROMICRO Conference (EUROMICRO)*, Västerås, Sweden, 1998, pp. 800–7.

[261] C. Wohlin and A. Wesslen, "Understanding software defect detection in the personal software process," in *Proceedings 9th International Symposium on Software Reliability Engineering (ISSRE)*, Paderborn, Germany, 1998, pp. 49–58.

[262] H. Wohlwend and S. Rosenbaum, "Schlumberger's software improvement program," *IEEE Trans. Softw. Eng.*, vol. 20, no. 11, pp. 833–9, Nov. 1994.

[263] Z. Xiaosong, H. Zhen, G. Fangfang, and Z. Shenqing, "Research on the application of six sigma in software process improvement," in *Proceedings 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Harbin, China, 2008, pp. 937–940.

[264] Z. Xiaosong, H. Zhen, ZhangMin, W. Jing, and Y. Dainuan, "Process integration of six sigma and CMMI," in *Proceedings 6th International Conference on Industrial Informatics (INDIN)*, Singapore, China, 2008, pp. 1650–1653.

[265] R. Xu, Y. Xue, P. Nie, Y. Zhang, and D. Li, "Research on CMMI-based software process metrics," in *Proceedings 1st International on Computer and Computational Sciences (IMSCCS)*, Hangzhou, China, 2006, pp. 391–397.

[266] J. Zettell, F. Maurer, J. Münch, and L. Wong, "LIPE: a lightweight process for e-business startup companies based on extreme programming," in *Product Focused Software Process Improvement*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2001, vol. 2188, pp. 255–70.

Michael Unterkalmsteiner is a PhD student at the Blekinge Institute of Technology (BTH) 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 co-optimization of requirements engineering and verification & validation processes. He received the B. Sc. degree in applied computer science from the Free University of Bolzano / Bozen (FUB) in 2007 and is currently completing the M. Sc. degree in software engineering at BTH.

Tony Gorschek is a professor of software engineering at Blekinge Institute of Technology (BTH) with over ten years industrial experience. He also manages his own industry consultancy company, works as a CTO, 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 innovation. Contact him at tony.gorschek@bth.se or visit www.gorschek.com.

A. K. M. Moinul Islam is a researcher at the Technical University of Kaiserslautern, Germany. He is with the Software Engineering: Process and Measurement Research Group. His research interests include global software engineering, software process improvement and evaluation, and empirical software engineering. He received his double master's degree, M. Sc. in Software Engineering, in 2009 jointly from University of Kaiserslautern, Germany and Blekinge Institute of Technology, Sweden within the framework of European Union's Erasmus Mundus Programme. Prior to his master's degree, he worked for 3 years in the IT and Telecommunication industry.

Chow Kian Cheng is a software engineer at General Electric International Inc. based in Freiburg, Germany. He is responsible for the development of clinical software in the healthcare industry. He holds a joint master degree, M. Sc. in Software Engineering, from the Blekinge Institute of Technology, Sweden and the Free University of Bolzano / Bozen, Italy. Prior to his master degree, he worked for 4 years with Motorola Inc. and Standard Chartered Bank.

Rahadian Bayu Permadi received his Bachelor degree in Informatics from Bandung Institute of Technology, Indonesia. He obtained in 2009 the double-degree master in software engineering from the Free University of Bolzano / Bozen, Italy and the Blekinge Institute of Technology, Sweden. Currently, he is working as a software engineer at Amadeus S. A. S, France. His interests are software measurements & process improvement, software architecture and software project management. He was a Java technology researcher in Indonesia before he got awarded with the Erasmus Mundus scholarship for European Master in Software Engineering programme.

Robert Feldt (M'98) is an associate professor of software engineering at Chalmers University of Technology (CTH) as well as at Blekinge Institute of Technology. He has also worked as an IT and Software consultant for more than 15 years. His research interests include software testing and verification and validation, automated software engineering, requirements engineering, user experience, and human-centered software engineering. Most of the research is conducted in close collaboration with industry partners such as Ericsson, RUAG Space and SAAB Systems. Feldt has a PhD (Tekn. Dr.) in software engineering from CTH.