On the Role of Software Quality Management in Software Process Improvement

Wiedemann Jacobsen, Jan; Kuhrmann, Marco; Münch, Jürgen; Diebold, Philipp; Felderer, Michael

Published in:
Product-Focused Software Process Improvement

DOI:
10.1007/978-3-319-49094-6_21

Publication date:
2016

Document version
Peer reviewed version

Document license
Unspecified

Citation for pulished version (APA):
Wiedemann Jacobsen, J., Kuhrmann, M., Münch, J., Diebold, P., & Felderer, M. (2016). On the Role of Software Quality Management in Software Process Improvement. In P. Abrahamsson, A. Jedlitschka, A. N. Duc, M. Felderer, S. Amasaki, & T. Mikkonen (Eds.), Product-Focused Software Process Improvement: Proceedings of the 17th International PROFES conference (pp. 327-343). Springer. (Lecture Notes in Computer Science). DOI: 10.1007/978-3-319-49094-6_21

On the Role of Software Quality Management in Software Process Improvement

Jan Wiedemann Jacobsen[1], Marco Kuhrmann[1], Jürgen Münch[2], Philipp Diebold[3], Michael Felderer[4]

[1] University of Southern Denmark, The Mærsk Mc-Kinney Møller Institute, Odense, Denmark
janwj12@student.sdu.dk, kuhrmann@mmmi.sdu.dk
[2] Herman Hollerith Center, Reutlingen University, Böblingen, Germany
juergen.muench@reutlingen-university.de
[3] Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany
philipp.diebold@iese.fraunhofer.de
[4] University of Innsbruck, Institute of Computer Science, Innsbruck, Austria
michael.felderer@uibk.ac.at

Abstract. Software Process Improvement (SPI) programs have been implemented, inter alia, to improve quality and speed of software development. SPI addresses many aspects ranging from individual developer skills to entire organizations. It comprises, for instance, the optimization of specific activities in the software lifecycle as well as the creation of organizational awareness and project culture. In the course of conducting a systematic mapping study on the state-of-the-art in SPI from a general perspective, we observed Software Quality Management (SQM) being of certain relevance in SPI programs. In this paper, we provide a detailed investigation of those papers from the overall systematic mapping study that were classified as addressing SPI in the context of SQM (including testing). From the main study's result set, 92 papers were selected for an in-depth systematic review to study the contributions and to develop an initial picture of how these topics are addressed in SPI. Our findings show a fairly pragmatic contribution set in which different solutions are proposed, discussed, and evaluated. Among others, our findings indicate a certain reluctance towards standard quality or (test) maturity models and a strong focus on custom review, testing, and documentation techniques, whereas a set of five selected improvement measures is almost equally addressed.

Keywords: Software Process Improvement, Software Quality Management, Software Test, Systematic Mapping Study, Systematic Literature Review

1 Introduction

To organize software development companies look for *Software Process Improvement* (SPI; [19]) allowing them to analyze and to continuously improve their development approaches. In the course of conducting a systematic mapping study [24], SPI was mentioned a diverse field: many SPI facets are studied, several hundreds of custom SPI approaches were proposed, e.g., to address weaknesses of standard approaches like CMMI [34], SPI success factors are collected and analyzed, and new trends such as SPI

employing agility as improvement principle are addressed. SPI thereby aims at improving companies' competitiveness and is considered important regardless of a company's size [16].

Besides accelerated development procedures, the quality of the software products developed is another important criterion (cf. Bennett and Weinberg [4], who found bug fixing cost increasing by magnitudes in later lifecycle phases). Therefore, improving the quality of software and determining the economic value [14], notably for small and very small companies [28] is of certain relevance. For those companies, emphasizing quality is crucial, as software testing is a strenuous and expensive process [5] consuming up to 50% of the total development costs [17]. Therefore, improving the quality management and, in particular, the software test activities provide a perfect starting point for improving the software process and hence product quality.

Problem Statement and Objective SPI programs have been implemented to improve product quality and speed of software development and have shown impact [2]. Also, software quality assurance techniques play an important role to guarantee and improve quality. Yet, the role of software quality assurance and SQM in SPI programs has not explicitly been investigated so far. The objective of this research is therefore to analyze the literature to characterize the role of SQM in SPI.

Contribution This paper provides an overview of the study population on SPI with a special focus on SQM and shows how these studies are evaluated. It presents the software quality assurance techniques and improvement measures addressed in SPI. Our findings show indication that SPI in the context of SQM is equally focussed on software testing as well as on complementing (support) activities including reviews and documentation techniques. Furthermore, our findings show a trend towards utilizing individual testing approaches rather than implementing/following standards.

Context: A Systematic Mapping Study on SPI This study is grounded in a comprehensive systematic mapping study on the state of SPI of which the findings where published in [24] (to which we refer to as the *main study*). Outcomes of this study show SPI being an actively researched topic, yet lacking theories and models. Instead, the field of SPI is shaped by a constant rate of approx. 10-12 new SPI models per year. These trends observed were used to form topic clusters of which one cluster addresses *Software Quality Management and Software Test*. The study at hand investigates this particular cluster in more detail utilizing a systematic review (cf. Sect. 3).

Outline The remainder of the paper is organized as follows: Section 2 discusses related work. In Sect. 3, we describe our research approach, before we present the results of our study in Sect. 4. We provide a discussion on the results in Sect. 5 and conclude the paper in Sect. 6.

2 Related Work

In (general) SPI, different topics are researched in secondary studies. For instance, Monteiro and Oliveira [31], Bayona-Oré [3], and Dybå [7] study SPI success factors, while

Helgesson et al. [15] and van Wangenheim et al. [36] review maturity models, and Hull et al. [18] review different assessment models. These exemplarily mentioned studies show that the SPI community has started the search for generalizable knowledge. Yet, the mentioned studies address more general SPI issues.

The study at hand is the first literature study explicitly dedicated to the role of *Software Quality Management* (SQM) and *Test Process Improvement* (TPI) in SPI. It is, however, related to other reviews and secondary studies in SPI, TPI, and the improvement of other analytical and constructive software quality aspects. For instance, regarding TPI, Afzal et al. [1] provide a systematic review, which identified 18 approaches and their characteristics, and an industrial case study on two prominent approaches, i.e., TPI Next and TMMi. Authors found that many of the test process improvement approaches do not provide sufficient information nor do the approaches include assessment instruments. A systematic review by Garcia et al. [10] identified 23 test process models, many of them adapted from TMMi and TPI. Reviews and comparisons of TPI models are also covered by a number of industrial white papers (so-called "grey literature", e.g., [21, 27]), which points to the practical relevance of this field. At the more general level of analytical verification and validation processes, Farooq and Dumke [9] discuss research directions for the improvement of verification and validation processes. Authors identify research challenges concerning quantitative management, improvement of existing approaches, approaches for emerging development environments as well as empirical investigation of success factors and tool selection. Regarding constructive software quality aspects, several systematic reviews (e.g., for software documentation [39]) are available, but reviews discussing these quality aspects in relation to SPI are missing so far.

All these representatively selected studies address specific topics, yet, they do not contribute to a more general perspective on SPI in the context of SQM. The paper at hand thus fills a gap in literature by collecting and analyzing publications that emphasize SPI in the SQM context and, therefore, also lays the foundation to direct future research in this field in SPI research.

3 Research Design

This study is an in-depth analysis of a data subset identified in a systematic mapping study [24]. In this section, we present the research design including research questions, data collection and analysis procedures, as well as considerations on the study's validity. Our research approach for the present study follows the procedures applied in [25]; an in-depth analysis of SPI in Global Software Engineering.

3.1 Research Questions

In the course of analyzing the selected papers on SQM, this study aims to answer the following research questions:

RQ 1 *What is the study population on SPI with a special focus on SQM?* This research question aims at capturing the field of SPI from the perspective of quality management and test. It also helps positioning the sub-study to the main study.

RQ 2 *Which software quality assurance techniques and improvement measures are addressed in SPI?* Based on 58 new metadata attributes, this research question aims at determining the different quality assurance techniques and improvement measures addressed by SPI.

RQ 3 *How are studies on SQM in SPI evaluated?* This research question is concerned with the determination of the impact of the investigated studies, in particular, to determine the rigor and relevance [20] of the result set.

3.2 Data Collection Procedures

Being a study on a data subset (see also [25]), in this study, we had no need for an explicit and self-contained data collection. Input data was obtained from the main study's result set [24], which we refer to as the study's *raw data*. The selection of the data of interest in the raw data was carried out by selecting all publications from the raw data having the attributes "Quality Management" and/or "Test" set (Fig. 3), which initially results in 96 publications. The resulting subset (to which we refer to as the *study data*) was then copied to an own spreadsheet. To improve the reliability of the data analysis, two external researchers joined the team. Finally, two researchers carried out the data selection and cleaning procedures and the initial data analysis, one researcher was concerned with the definition of the extended metadata set and the data classification and analysis, and the two remaining researchers took over quality assurance tasks.

Having the study data available, in the course of downloading all selected papers, an initial quality assurance was performed. This quality assurance led to the exclusion of four papers (reasons: misclassification, violation of language constraints). Those papers' metadata was updated, such that they will be returned to the main study (Sect. 6). Eventually, **92** papers remained in the cleaned study dataset, which where then analyzed as described in Sect. 3.3.

3.3 Analysis Procedure

As "preparatory" study with the purpose of getting the big picture, the main study was conducted as a systematic mapping study following the guidelines as proposed by Petersen et al. [32]. The present study however aims to deliver more insights and details and, thus, is carried out also using the systematic review instrument as described by Kitchenham and Charters [23]. In particular, during the paper download and quality assurance, the initial metadata set (40 attributes, Fig. 3) was revisited and, if necessary, updated. Furthermore, with calling in an external researcher (an expert in quality management and testing), the set of metadata was substantially extended by 58 extra attributes in nine new metadata categories (see Fig. 4).

During the analysis, each paper was inspected by two researchers, who checked (and if necessary revised) the initial values of the metadata, provided an initial assignment of values to the new attributes, and developed a paper summary of 2-3 sentences. Finally, to evaluate the papers regarding their rigor and relevance, we applied the model proposed by Ivarsson and Gorschek [20] to complete the picture. These steps were iteratively double-checked by a third researcher, and finally independently checked by the

two researchers concerned with (general) quality assurance. The analysis as such utilizes descriptive statistics (e.g., charts and tables), whereas we mainly rely on bubble-charts and heat maps.

3.4 Validity Procedures

To improve the validity of the results, we applied the following measures: First, we called in two external researchers and formed two teams. Team 1 (3 persons) conducted the data analysis, while team 2 (2 persons) was taking over the quality assurance. Second, in the data analysis phase, team 1 re-applied the procedures of the main study [24], i.e., all papers were re-inspected to check the correct assignment and to complete the assignment of the 40 metadata attributes. Third, in the inspection, the assignment of the attributes (40: main study, 58: new, scoped), and the evaluation according to the rigor-relevance model [20] were carried out using the *systematic review* instrument [23] using the full text of the study-relevant papers.

4 Study Results

In this section, we present the results of the study. We start with an overview of the study population, before we present the results of the analyses structured according to the research questions in Sect. 4.1 – Sect. 4.3. Section 5 presents an integrated discussion of the results obtained from the study.

In total, 92 papers remained in the study data set for inspection. Figure 1 provides an overview of the publication frequency in the study timeframe. In general, in the result set, we see about 3-4 papers on the topic of interest published per year, but Fig. 1 also shows a first big jump in 1998 (from there on, the average publication frequency is 5+ papers per year). In subsequent sections, we provide further details and analyze them in relation to the trend observed.

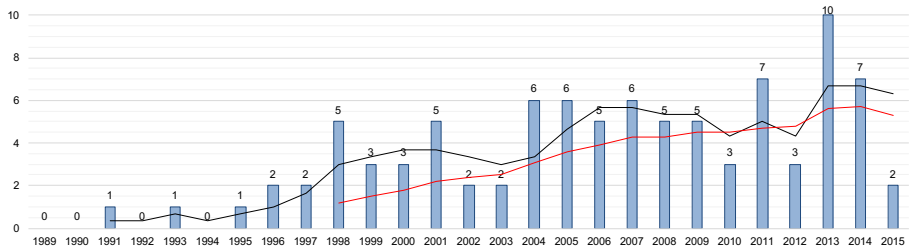


Fig. 1: Number of publications on SPI with a focus on software quality management and/or testing ($n = 92$). The graph includes two trend lines to visualize the long-term development of the field (calculation basis: mean, 3-year (black) and 10-year (red) period), which show periodical waves, but also a continuously growing general interest.

4.1 RQ1: General Study Population

In this section, we first give an overview of the general study dataset using the instruments from the main study [24] to allow for comparability. Figure 2 provides an integrated overview of the study dataset according to the classification using the standard schemas (*research type facet* (RTF) according to Wieringa et al. [37]) and *contribution type facet* (CTF) according to Petersen et al. [32]).

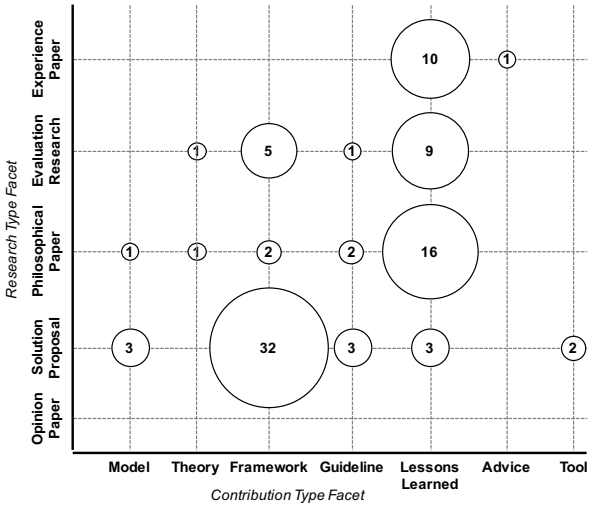


Fig. 2: Classification of the study dataset according to the RTF and CTF schemas.

Figure 2 shows the studied publications forming two CTF-clusters. In particular, SPI with a special emphasis on software quality management and software test is mainly reported as *framework* or as *lessons learned*, whereas the framework-classified papers usually propose solutions and the lessons learned emerge from experience and evaluation research. Furthermore, a considerable share of the *lessons learned* papers are classified as *philosophical papers*, i.e., secondary studies or discussion/comparison papers. In line with the findings from the main study [24], models and theories are in the minority or missing. Another (unexpected) finding is the small number (only 2 out of 92 papers) of tool-related publications. However, although tools are underrepresented in the "formal" literature, in [11], authors argue that more tool-related material can be found in the "grey literature". Insofar, the chart from Fig. 2 can be considered consistent with the findings from [11].

Figure 3 shows the classification of the study dataset using the metadata system introduced in [24]. Regarding the *process* dimension, the study dataset shows a strong focus on general improvement and custom models. Furthermore, standard SPI and maturity models (CMMI and ISO/IEC 15504) are addressed, but we can also see a certain focus on general measurement (and assessment) activities. Regarding the *context* dimension, in the lifecycle phases, only project management is significantly represented

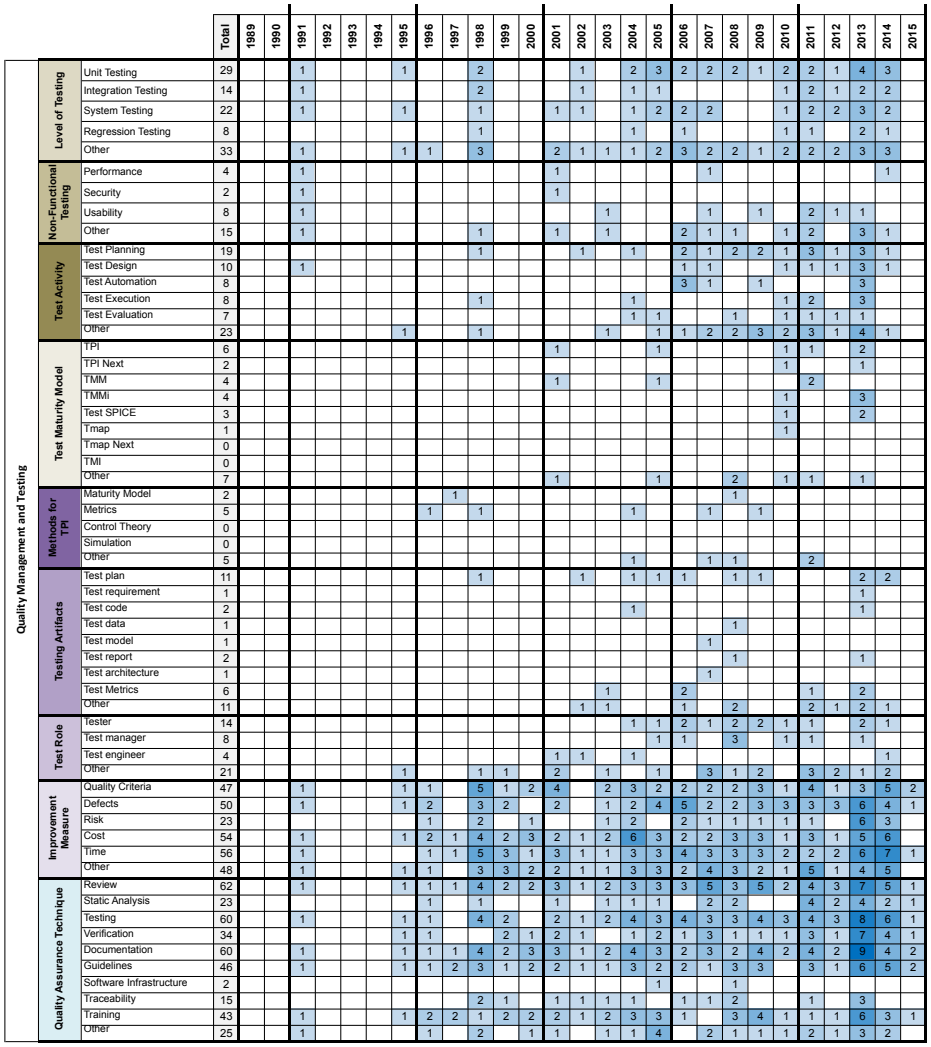
			Total	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Dimension: Process	Publication Objective	Agile / Lean	19															1	2		1		2	2	2	2	1	5	1	
		Process Simulation	3											1		1					1									
		Process Line/Patterns	1																	1										
		Product Line/Management	1																	1										
		Success Factors	17					1			1		1				2		1	1		1	2	2		1		2	1	1
		Custom Model	56			1				1	1		2	3	1	3		2	3	5	3	4	3	3	2	4	3	7	4	1
		General Improvement	31					1			1	2	4	1	1	3		1	2	2	2	1	1	3		3		2		1
	Assessment and Ass. Models	CMMI	59					1		1	1	2	4	3	2	3	2	2	5	4	2	1	3	2	1	4	1	9	4	2
		ISO/IEC 15504	30					1		1	1	1	1	2	1	1	1		5	2		2	1		2	1		6		1
		General Measurement	37			1		1		1	1	1		1	1	1	1		2	1	3	3	3	2	1	4	2	3	3	1
	(Quasi-) Standards (and Techniques)	Six Sigma	4															1									1		1	1
		Bootstrap	13										1	2		2	1		2	1						1		1	1	1
		CompetiSoft	2																				1						1	
		Continuous Improvement	6												1				1					1	1	2				
		PSP/TSP	7											1					1	2	1			1						1
		ISO/IEC 29110	1																										1	
		ISO/IEC 12207	6																			1	1					1	1	2
Dimension: Study Type and Method		Survey/Interview	16								1				1	1	1			1	2	1	1			2		2	3	
		Single Case-Study	33							1			2	1	1	1	1	2	3	3	1	3	1	2		2	2	3	3	1
		Multi-Case/Long. Study	24					1			1	2	3			3				1	2		1	3	2	2	1	1	1	
		Replication Study	0																											
		SLR/SMS	7																				1			2		1	2	1
		Grounded Theory	4																		1			1		1		1		
Dimension: Context	Life Cycle Phases	Project Management	33								1		4	2	1	1	1	2	1	4	3	2	1	2		2		3	3	
		Quality Management	89			1		1		1	2	2	5	3	3	5	2	2	6	6	5	6	5	5	2	7	2	9	7	2
		Requirements Engineering	10										1	2			1	1	1		1	1						1	1	
		Architecture	4											1			1		1			1								
		Implementation	3											1							1								1	
		Test	37								1		2	1		2	1	1	2	2	2	1	2	2	3	3	2	6	4	
	Company Size and Scale	VSE/SME	22					1			1		1		1	1			2	1	1	1	2	2	1	2	1	1	3	
		Other Company Size	31					1		1		2	4		1	2	1	1	2	2	2	2	1	4	1			3	1	
		GSE	22					1		1		1				3	2	1	2	2	1	2		1	2			2	1	
	Application Domain	Embedded Systems	12										1			1			2	2		1				1	1	2	1	
		Telecommunications	12							1	1				1		1	1	1	1	1			1		1	1	1			
		Medical Devices	4													1					1							2		
		Automotive	4																2	1								1		
		Mission-critical Defense	5									1	1										1	1			1			
		Business IS	3														1	1				1								
		Web/Mobile/Cloud	6														1				1	1		1				1	1	
		Skills and Education	1									1																		

Fig. 3: Overview of the different standard metadata attributes addressed over time. The darker the color, the more papers in a year have this attribute assigned, whereas one paper can have multiple attributes assigned.

showing the close relation of project- and quality management. Other lifecycle phases are scarcely addressed, which suggests the publications from the dataset being narrowly scoped. Concerning the *application domain*, the classification does not highlight any favorite, i.e., SQM and testing are considered relevant in all application domains. Finally, regarding the *company size and scale* group, publications address companies of all sizes. Furthermore, globally distributed development is also addressed by the study dataset. Figure 3 shows the studied field mostly researched in a practical manner, i.e., case study research is found the most frequently used instrument. The figure shows that a number of multi-case or longitudinal studies are available (which is above the general tendency observed in the main study), yet, still, replication research is absent.

4.2 RQ2: Improvement Measures and Quality Assurance Techniques

To investigate which improvement measures and quality assurance techniques are addressed by the study dataset, we extended the metadata system from [24] and defined



Quality Management and Testing			Total	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
	Level of Testing	Unit Testing	29			1				1			2				1		2	3	2	2	2	1	2	2	1	4	3	
		Integration Testing	14			1							2				1		1	1					1	2	1	2	2	
		System Testing	22			1				1			1			1	1		1	2	2	2			1	2	2	3	2	
		Regression Testing	8										1						1		1				1	1		2	1	
		Other	33			1				1	1		3			2	1	1	1	2	3	2	2	1	2	2	2	3	3	
	Non-Functional Testing	Performance	4			1										1						1							1	
		Security	2			1										1														
		Usability	8			1												1				1		1		2	1	1		
		Other	15			1							1			1		1			2	1	1		1	2		3	1	
	Test Activity	Test Planning	19										1				1		1		2	1	2	2	1	3	1	3	1	
		Test Design	10			1															1	1			1	1	1	3	1	
		Test Automation	8																		3	1		1				3		
		Test Execution	8										1						1						1	2		3		
		Test Evaluation	7																1	1			1		1	1	1	1		
		Other	23							1			1					1		1	1	2	2	3	2	3	1	4	1	
	Test Maturity Model	TPI	6													1				1					1	1		2		
		TPI Next	2																						1			1		
		TMM	4													1				1						2				
		TMMi	4																						1			3		
		Test SPICE	3																						1			2		
		Tmap	1																						1					
		Tmap Next	0																											
		TMI	0																											
		Other	7													1				1			2		1	1		1		
	Methods for TPI	Maturity Model	2									1											1							
		Metrics	5								1		1						1			1		1						
		Control Theory	0																											
		Simulation	0																											
		Other	5																1			1	1			2				
	Testing Artifacts	Test plan	11										1				1		1	1	1		1	1				2	2	
		Test requirement	1																									1		
		Test code	2																1									1		
		Test data	1																				1							
		Test model	1																			1								
		Test report	2																				1					1		
		Test architecture	1																			1								
		Test Metrics	6															1			2					1		2		
		Other	11														1	1			1		2			2	1	2	1	
	Test Role	Tester	14																1	1	2	1	2	2	1	1		2	1	
		Test manager	8																	1	1		3		1	1		1		
		Test engineer	4													1	1		1										1	
		Other	21							1			1	1		2		1		1		3	1	2		3	2	1	2	
	Improvement Measure	Quality Criteria	47			1				1	1		5	1	2	4		2	3	2	2	2	2	3	1	4	1	3	5	2
		Defects	50			1				1	2		3	2		2		1	2	4	5	2	2	3	3	3	3	6	4	1
		Risk	23								1		2		1			1	2		2	1	1	1	1	1		6	3	
		Cost	54			1				1	2	1	4	2	3	2	1	2	6	3	2	2	3	3	1	3	1	5	6	
		Time	56			1					1	1	5	3	1	3	1	1	3	3	4	3	3	3	2	2	2	6	7	1
		Other	48			1				1	1		3	3	2	2	1	1	3	3	2	4	3	2	1	5	1	4	5	
	Quality Assurance Technique	Review	62			1				1	1	1	4	2	2	3	1	2	3	3	3	5	3	5	2	4	3	7	5	1
		Static Analysis	23								1		1			1		1	1	1		2	2			4	2	4	2	1
		Testing	60			1				1	1		4	2		2	1	2	4	3	4	3	3	4	3	4	3	8	6	1
		Verification	34							1	1			2	1	2	1		1	2	1	3	1	1	1	3	1	7	4	1
		Documentation	60			1				1	1	1	4	2	3	3	1	2	4	3	2	3	2	4	2	4	2	9	4	2
		Guidelines	46			1				1	1	2	3	1	2	2	1	1	3	2	2	1	3	3		3	1	6	5	2
		Software Infrastructure	2																	1			1							
		Traceability	15										2	1		1	1	1	1		1	1	2			1		3		
		Training	43			1				1	2	2	1	2	2	2	1	2	3	3	1		3	4	1	1	1	6	3	1
		Other	25			1					1		2		1	1		1	1	4		2	1	1	1	2	1	3	2	

Fig. 4: Overview of the 58 new metadata attributes addressed over time. The darker the color, the more papers in a year have this attribute assigned, whereas one paper can have multiple attributes assigned.

58 new attributes for classifying the papers under study. We added "Quality Management and Testing" as new dimension, and we refined this dimension into nine groups (Fig. 4). For space limitations, in the following, we provide the big picture in Fig. 4, but focus on the groups "Improvement Measures" and "Quality Assurance Techniques". The big picture in Fig. 4 shows the groups *test activity*, *non-functional testing*, and *level of testing* well covered. Furthermore, the dataset provides rich information regarding the groups *improvement measures* and *quality assurance techniques*. However, espe-

		Improvement Measure						Quality Assurance Technique									
		Quality Criteria	Defects	Risk	Cost	Time	Other	Review	Static Analysis	Testing	Verification	Documentation	Guidelines	Software Infrastructure	Traceability	Training	Other
Study Type	Survey / Interviews	7	6	7	9	10	8	9	3	9	5	7	9		1	5	3
	Single Case-Study	17	21	6	22	20	19	27	12	25	16	26	16	1	8	20	11
	Multi-Case/Long. Study	10	13	7	13	16	10	14	3	16	4	15	10		3	10	5
	Replication Study																
	SLR/SMS	5	3	1	4	4	4	3	1	3	2	3	6		1	1	1
	Grounded Theory	3	1	1	3	3	3	3	1	3	1	2	2			1	

Fig. 5: Overview study types applied to the groups *improvement measures* and *quality assurance techniques*.

cially regarding test maturity models (or "standardized" testing approaches in general), the dataset provides only little information, which indicates to a confirmation of the observed trend from [24] regarding the reluctance towards standardization—also for quality management and testing (and as initially found in [26]).

Regarding the groups "Improvement Measures" and "Quality Assurance Techniques", in the data, we see a fairly balanced distribution, i.e., a variety of topics is equally researched. The only remarkable outlier is the attribute *software infrastructure*. Favorites regarding the *improvement measures* are the improvement of defect handling (50 mentions), cost and time optimization (54 and 56 mentions). Regarding the *quality assurance techniques*, review (62), as well as testing and documentation (60 mentions each) are the most frequently mentioned ones. Subsequent sections provide further details for the aforementioned two "favorite" groups.

4.3 RQ3: Evaluation of Software Quality Management and Software Testing

In this section, we limit our analysis to the groups *improvement measures* and *quality assurance techniques*. As a first step, we review the study methods applied to the papers reporting knowledge in the groups of interest. In the second step, the publications contributing to the groups of interest are evaluated according to the rigor-relevance model [20] to allow for rating the (general) impact of the different topics.

Methods Applied Figure 5 provides a heat map summarizing the study types applied to investigate the different topics. The overview shows that SPI in the context of SQM is a fairly practically researched field. The majority of the papers assessed combine different research methods, whereas case study research is the most used approach—quite often in a mixed-method approach and also implementing a multi-case or longitudinal study approach (for term definitions, see Wohlin et al. [38]). A remarkable insight is the absence of replication research. Secondary studies and research based on Grounded Theory is present in the study data set, yet the action research approach prevails. Regarding the topic clusters, from the data, we see the cluster "Improvement Measures" fully covered, whereas in the cluster "Quality Assurance Technique" the topics *software infrastructure*, *traceability*, *training*, and *other* are only partially covered.

Evaluation of Rigor and Relevance In the second step, we evaluate the papers within the groups of interest for their rigor and relevance according to [20]. In the overall dataset, 58 out of 92 papers are rated highly relevant (4 points), and of those, 37 papers are rated of high to very high rigor (2-3 points). In the following, we break-down our analysis to the groups "Improvement Measures" (Fig. 6) and "Quality Assurance Techniques" (Fig. 7). In Sect. 5, we use the following presentation to direct the detailed discussion.

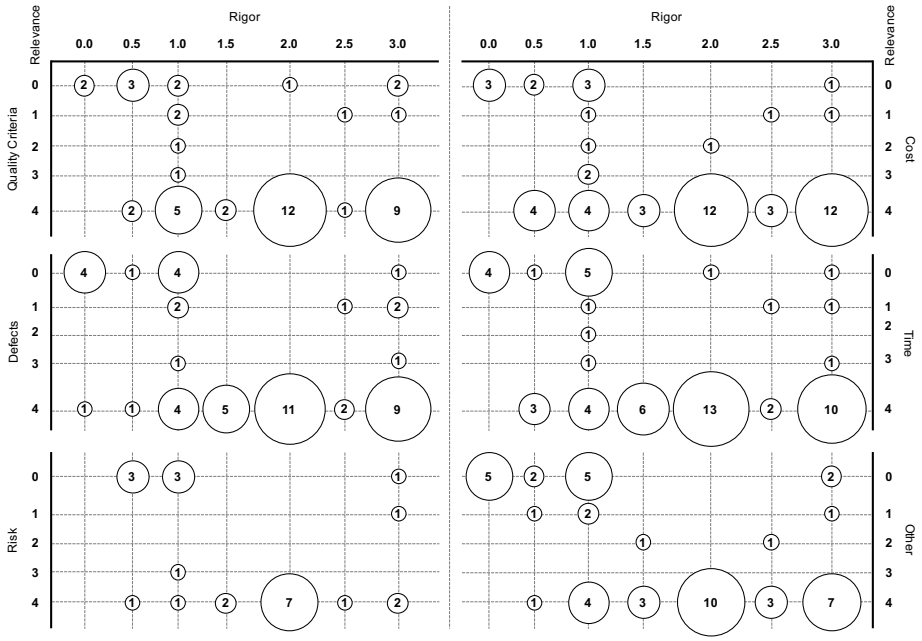


Fig. 6: Classification of the study dataset (attributes from "Improvement Measures") according to the rigor-relevance model.

Figure 6 visualizes the six topics within the group "Improvement Measures" and shows that the topics of favor in these groups are (general) *quality criteria*, *defects*, *cost*, and *time*. Research addressing the improvement of risk management is, so far, underrepresented and of less rigor and relevance. Remarkable, the majority of the papers in the aforementioned four categories is considered highly relevant (score 4).

Regarding the group "Quality Assurance Techniques", Fig. 7 shows the following topics of relevance: *review*, *testing*, *documentation*, *guideline*, and *training*. The groups *guideline* and *training* comply with an expectation when coming from the 'pure' SPI perspective—a focus on methods, their documentation (as guideline) and training. Among the more 'applicable' techniques, *review*, *testing*, and (test) *documentation* show a clear focus of the study data, whereas the techniques *static analysis* and *verification* are not that present in the data. More "sophisticated" topics, such as *traceability* and *software infrastructures* are (yet) not well represented in the study data.

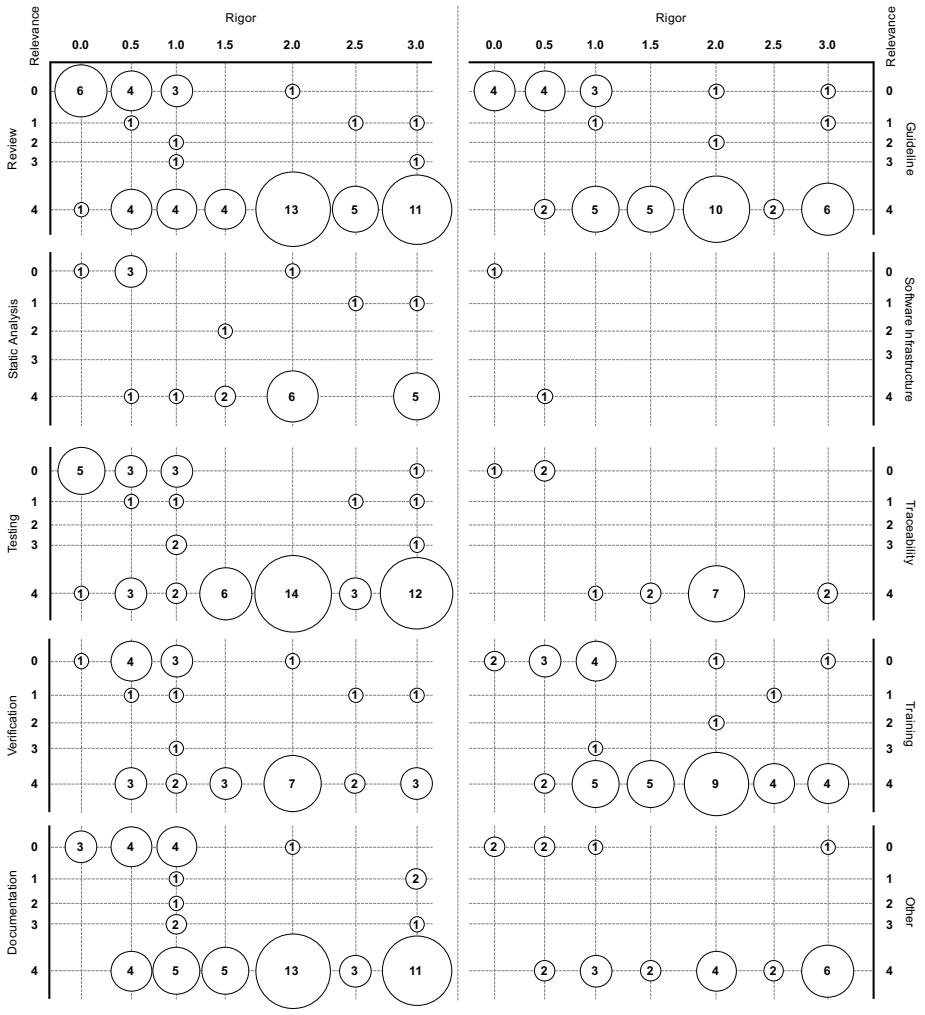


Fig. 7: Classification of the study dataset (attributes from "Quality Assurance Techniques") according to the rigor-relevance model.

5 Study Summary & Discussion

To provide a in-depth discussion, we ranked the highest rated papers regarding their coverage of improvement measures and quality assurance techniques (Fig. 6 and Fig. 7; both based on the classification according to the rigor-relevance model). Table 1 summarizes these papers for the two categories "Improvement Measure" and "Quality Assurance Technique", whereas we only provide a subset for the in-depth discussion. In particular, we select the papers [8, 14, 22, 29] as sample from the study data set, as we found those papers represented in both categories.

Table 1: Overview of the highest rated papers according to the rigor-relevance model in the categories Improvement Measure and Quality Assurance Technique.

Impr. Measure	**Paper**	**QA Technique**	**Paper**
Quality Criteria	[8, 13, 14, 22, 29, 30, 33]	Review	[8, 14, 22, 29, 35]
Defects	[12–14, 22, 29, 30]	Static Analysis	[6, 35]
Risk	[12, 30]	Testing	[6, 8, 14, 22, 29, 35]
Cost	[8, 12–14, 22, 29, 30, 33]	Verification	[22, 35]
Time	[8, 12–14, 22, 29, 30, 33]	Documentation	[6, 8, 14, 22, 29, 35]
		Guideline	[8, 14, 22]
		Software Infrastructure	—
		Traceability	[6, 22]
		Training	[6, 8, 35]
Other	[8, 12, 13, 30, 33]	Other	[6, 8, 22, 35]

Elliot et al. [8] document a methodology for implementing a software quality management system (SQMS). Table 1 shows the method proposed addressing quality management in general thus covering a number of attributes (in particular *documentation*, *guideline*, and *training*; *reviews* and (general) *testing* were mentioned as concrete techniques to, inter alia, better address different quality criteria, especially in the "system use" section). Key factors for the successful implementation of the SQMS were staff training and treating users like customers, which was also required for a cultural change within the organization.

Harter et al. [14] present a framework for assessing the economic value of SPI and quality over the software lifecycle. The effects to be measured are defined based on the number of defects (development quality: defects found prior customer testing; conformance quality: defects found in customer testing prior acceptance)—similar measures are defined for development effort and cycle time, and support costs. Therefore, in [14], authors mainly address the attributes *defects*, *cost*, and *time* to conclude the economic value of SPI (Table 1). Eventually, authors found that higher quality is associated with reduced cycle times and development effort, and that savings accrue due to reduced rework and, moreover, that support activity savings outweigh development savings. Harter et al. conclude that future research efforts should focus on how SPI strategies affect support activities.

Kasoju et al. [22] use evidence-based software engineering (EBSE) to help an organization improve its testing process (domain: automotive software). They use an in-depth investigation of automotive test processes using a mixed-method approach including case study research, systematic reviews and value stream analysis/mapping. For eight analyzed projects, authors collect information regarding the test approaches, project/system kind and size, and the development approach used (Table 1; mainly attributes *cost*, *time*, *testing*, *verification*). In interview sessions, among other things, authors found interviewees stating a lack of a clear test process, which can be applied to any project lifecycle. Only 3 out of 8 studied projects follow a defined process (which indicates to the mainly individual and non-standardized process selection as already found in [26]; moreover, authors found that a basic testing strategy is actually defined,

yet not implemented by most of the teams, which is also consistent with our previous findings from [26]). Eventually, in [22], authors conclude strengths found for automotive software testing, such as work in small agile teams, implementing agile (communication) practices, or different approaches like exploratory testing. However, authors also mention that these findings also depend on project/team size, i.e., teams of different size might go for different solution, e.g., comprehensive test case management tools are considered more valuable for larger teams. Nevertheless, authors found process issues problematic for teams of any size (consistent with [16]), e.g., lacking unified testing process, unawareness of the process, or different process-related constrains like available time windows. Finally, authors identified seven wastes, which were mapped to the testing process to drive process improvement.

Li et al. [29] describe how agile processes affect software quality, software defects and defect fixing efficiency (Table 1; mainly attributes *defects*, *testing*, *time*). A major finding is that a significant reduction of defect densities or changes of defect profiles could not be found after Scrum was used. Yet, due to the iterative development approach, the development was considered more efficiently (e.g., fewer surprises, better control over the quality, and better schedule adherence). However, on the downside, authors also mention that Scrum puts more stress and time pressure on the developers (which could make them more reluctant towards performing tasks relevant for later maintenance). In a nutshell, authors conclude that the actual development approach is less important than iterative development and early testing (in their study, authors showed that about half of the (critical) defects was identified and fixed early thus reducing the risk of finding bugs late).

Summarizing the big picture obtained (Fig. 4) and the exemplarily selected papers (Table 1), we conclude: first, testing as such is not that massively represented in the study data as expected. For this, we argue that there is specialized (grey) literature on test process improvement (TPI), which is not properly linked to SPI—a phenomenon that we already observed for GSE [25]. In particular, so far, we did not found detailed data, e.g., regarding the actual impact of switching to an alternative test approach. On the other hand, we found indication for individual and project-specific test approach selection (even in highly-regulated domains; [22]), which confirms a finding we made in [26]. Second, so far, we found improving the quality focussing on reducing the number of defects. In [22] and [29], the authors found a lack of unified (standardized) testing approaches [22], and that the actual development approach (agile or traditional) seemingly not affects the defect densities or defect profiles. Harter et al. [14] suggest putting more effort in improving support activities. It therefore remains as a question for future work whether an SPI program with a "broader" perspective is more beneficial then optimizing a "technical" test method.

Threats to Validity In the following, we evaluate our findings and critically review our study regarding the threats to validity. As a literature study, this study suffers from potential incompleteness of the search results and a general publication bias. Beyond this general threat to validity, we have to particularly discuss the internal and external validity. The *internal validity* could be biased by personal ratings of the researchers. To address this risk, we continued and refined our study [24], which follows a proven procedure that utilizes different tools and researcher triangulation to support dataset

cleaning, study selection, and classification. The internal validity is also affected by the limited data collection, in particular, no new data was collected, and data analyzed is derived from the main study that serves as an umbrella. Calling in extra researchers to analyze and/or confirm decisions therefore further increases internal validity. The *external validity* is threatened by missing knowledge about the generalizability of the results. Furthermore, this study "inherits" several limitations regarding the external validity by relying on the main study's raw data only. Consequently, this study also inherits the main study's scope thus having certain limitations regarding the generalizability. Nevertheless, to increase the external validity, further independently conducted studies are required to confirm our findings.

6 Conclusion

The paper at hand provides an in-depth investigation of how software quality management (SQM) is treated in software process improvement (SPI). Based on a systematic mapping study [24], we selected all papers from the main study's dataset that address the topics SQM and software testing. In total, in this study, we inspected 92 papers.

Our findings show indication that SPI in the context of SQM is equally focussed on software testing as well as on complementing (or support) activities including reviews and documentation techniques. Furthermore, our findings show a trend in SPI towards utilizing individual testing approaches rather than implementing/following standards. A detailed discussion of four exemplarily selected papers reveals that the actual software process is less relevant than a smart arrangement of test activities (early testing) and an interactive implementation of the development process [29]. Furthermore, Harter et al. [14] suggest putting more effort on supporting activities rather than optimizing (isolated) technical tasks.

Limitations Our study is limited by the context of the main study [24], yet showed some overlap and similar trends as obtained in other independently conducted studies, such as [11, 26]. In total, only 92 papers were selected for analysis and, therefore, this study cannot claim to have delivered a generalizable set of conclusions. A major limitation is the use of a given dataset only without an extra topic-specific literature search, which potentially limits the reliability of the data. An extension and a complementing search, however, is subject to future research.

Future Work This paper provides the first analysis iteration of the 92 papers selected thus barely scratching the surface. Future work therefore includes further detailed analyses of the study data. Furthermore, as being a study on a data subset, in future iterations, the data analyzed will be (re-)integrated with the main study's data to improve the overall data quality and reliability of the data.

References

1. W. Afzal, S. Alone, K. Glocksien, and R. Torkar. Software test process improvement approaches: A systematic literature review and an industrial case study. *Journal of Systems and Software*, 111:1–33, 2016.

2. N. Ashrafi. The impact of software process improvement on quality: in theory and practice. *Information & Management*, 40(7):677–690, 2003.
3. S. Bayona-Oré, J. Calvo-Manzano, G. Cuevas, and T. San-Feliu. Critical success factors taxonomy for software process deployment. *Software Quality Journal*, 22(1):21–48, 2014.
4. Bennett, T. and Wennberg, P. Eliminating Embedded Software Defects Prior to Integration Test. *Quality Assurance Institute Journal*, 2006.
5. A. Bertolino and E. Marchetti. A brief essay on software testing. *Software Engineering, 3rd edn. Development process*, 1:393–411, 2005.
6. D. Damian, D. Zowghi, L. Vaidyanathasamy, and Y. Pal. An industrial case study of immediate benefits of requirements engineering process improvement at the australian center for unisys software. *Empirical Software Engineering*, 9(1):45–75, 2004.
7. T. Dybå. An instrument for measuring the key factors of success in software process improvement. *Empirical Software Engineering*, 5(4):357–390, 2000.
8. M. Elliott, R. Dawson, and J. Edwards. An evolutionary cultural-change approach to successful software process improvement. *Software Quality Journal*, 17(2):189–202, 2009.
9. A. Farooq and R. R. Dumke. Research directions in verification & validation process improvement. *ACM SIGSOFT Software Engineering Notes*, 32(4):3, 2007.
10. C. Garcia, A. Dávila, and M. Pessoa. Test process models: Systematic literature review. In *Software Process Improvement and Capability Determination*, pages 84–93. Springer, 2014.
11. V. Garousi, M. Felderer, and M. V. Mäntylä. The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, EASE '16, pages 26:1–26:6, New York, NY, USA, 2016. ACM.
12. K. Gomes Camargo, F. Cutigi Ferrari, and S. Camargo Pinto Ferraz Fabbri. Identifying a subset of tmmi practices to establish a streamlined software testing process. In *Brazilian Symposium on Software Engineering*, SBES, pages 137–146. IEEE, 2013.
13. G. C. Green, A. R. Hevner, and R. W. Collins. The impacts of quality and productivity perceptions on the use of software process improvement innovations. *Information and Software Technology*, 47(8):543–553, 2005.
14. D. E. Harter, M. S. Krishnan, and S. A. Slaughter. The life cycle effects of software process improvement: A longitudinal analysis. In *Proceedings of the International Conference on Information Systems*, ICIS, pages 346–351, Atlanta, GA, USA, 1998. Association for Information Systems.
15. Y. Y. L. Helgesson, M. Höst, and K. Weyns. A review of methods for evaluation of maturity models for process improvement. *Journal of Software: Evolution and Process*, 24(4):436–454, 2012.
16. R. V. Horvat, I. Rozman, and J. Györkös. Managing the complexity of spi in small companies. *Software Process: Improvement and Practice*, 5(1):45–54, 2000.
17. L. Huang and B. Boehm. How much software quality investment is enough: A value-based approach. *Software, IEEE*, 23(5):88–95, 2006.
18. M. Hull, P. Taylor, J. Hanna, and R. Millar. Software development processes - an assessment. *Information and Software Technology*, 44(1):1–12, 2002.
19. Humphrey, W. S. *Managing the Software Process*. Addison Wesley, 1989.
20. M. Ivarsson and T. Gorschek. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, 16(3):365–395, June 2011.
21. S. Karthikeyan and S. Rao. Adopting the right software test maturity assessment model. Technical report, Cognizant, 2014.
22. A. Kasoju, K. Petersen, and M. V. Mäntylä. Analyzing an automotive testing process with evidence-based software engineering. *Information and Software Technology*, 55(7):1237 – 1259, 2013.

23. B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, Keele University, 2007.
24. M. Kuhrmann, P. Diebold, and J. Münch. Software process improvement: A systematic mapping study on the state of the art. *PeerJ Computer Science*, 2(1):1–38, 2016.
25. M. Kuhrmann, P. Diebold, J. Münch, and P. Tell. How does software process improvement address global software engineering? In *International Conference on Global Software Engineering*, ICGSE, pages 89–98. IEEE, 2016.
26. M. Kuhrmann and D. M. Fernández. Systematic software development: A state of the practice report from germany. In *International Conference on Global Software Engineering*, ICGSE, pages 51–60. IEEE, 2015.
27. P. Kumar. Test process improvement – evaluation of available models. Technical report, Maveric, 2012.
28. X. Larrucea, R. V. O'Connor, R. Colomo-Palacios, and C. Y. Laporte. Software process improvement in very small organizations. *IEEE Software*, 33(2):85–89, Mar 2016.
29. J. Li, N. B. Moe, and T. Dybå. Transition from a plan-driven process to scrum: A longitudinal case study on software quality. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '10, pages 13:1–13:10, New York, NY, USA, 2010. ACM.
30. F. McGarry, S. Burke, and B. Decker. Measuring the impacts individual process maturity attributes have on software products. In *Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International*, pages 52–60. IEEE, 1998.
31. L. F. S. Monteiro and K. M. de Oliveira. Defining a catalog of indicators to support process performance analysis. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(6):395–422, 2011.
32. K. Petersen, R. Feldt, S. Mujtaba, and M. Mattson. Systematic mapping studies in software engineering. In *International Conference on Evaluation & Assessment in Software Engineering*, EASE, pages 68–77. ACM, 2008.
33. F. J. Pino, F. García, and M. Piattini. Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*, 16(2):237–261, 2008.
34. M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy. An exploratory study of why organizations do not adopt cmmi. *Journal of Systems and Software*, 80(6):883–895, 2007.
35. M. Sylemez and A. Tarhan. Using process enactment data analysis to support orthogonal defect classification for software process improvement. In *International Conference on Software Process and Product Measurement*, IWSM-MENSURA, pages 120–125, Oct 2013.
36. C. G. von Wangenheim, J. C. R. Hauck, C. F. Salviano, and A. von Wangenheim. Systematic literature review of software process capability/maturity models. In *International Conference on Software Process Improvement and Capability Determination–SPICE*, 2010.
37. R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering*, 11(1):102–107, Dec. 2005.
38. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer, 2012.
39. J. Zhi, V. Garousi-Yusifoğlu, B. Sun, G. Garousi, S. Shahnewaz, and G. Ruhe. Cost, benefits and quality of software development documentation: A systematic mapping. *Journal of Systems and Software*, 99:175–198, 2015.