



Approaches to promote product quality within software process improvement initiatives: A mapping study



Gabriel Alberto García-Mireles^{a,b,*}, Ma Ángeles Moraga^{b,1}, Félix García^{b,1}, Mario Piattini^{b,1}

^a Departamento de Matemáticas, Universidad de Sonora, Blvd. Encinas y Rosales s/n col. Centro, 83000 Hermosillo, Sonora, México

^b Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, España

ARTICLE INFO

Article history:

Received 11 April 2014

Revised 30 January 2015

Accepted 31 January 2015

Available online 7 February 2015

Keywords:

Software product quality

Software process improvement

Systematic mapping study

ABSTRACT

Enhancing product quality might be a main goal of a software process improvement initiative (SPI). Quality is, however, a complex concept, and experts recommend identifying relevant product quality characteristics to satisfy users/customers' needs. There is thus a need to understand how SPI initiatives contribute to the improvement of software product quality characteristics. This paper aims to provide an overview of an up-to-date state-of-the-art regarding initiatives that focus on promoting product quality improvement by applying SPI approaches. This goal was achieved by conducting a systematic mapping study, as a result of which we identified 74 primary papers including both theoretical (75.7%) and empirical (24.3%) papers. The main product quality characteristics addressed are security, usability and reliability. Security-related process models, on the other hand, are those most cited (53%). The empirical papers suggest that traditional process reference models, such as CMM, CMMI or ISO 9001, moderately increase product quality characteristics, these principally being maintainability and reliability. However, there is a need for more empirical research to evaluate the impact of SPI initiatives on software product quality by considering contextual factors. SPI initiatives should be more driven by performance goals related to product quality characteristics.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Software organizations usually increase product quality by implementing a software process improvement (SPI) initiative (Staples and Niazi, 2008). The common approach used to assess the effectiveness of the SPI program has traditionally been that of measuring the number of defects found in products. The assumption behind this is that an end product with many defects lacks quality (Card, 1998). However, this approach provides a narrow view of software quality (Ashrafi, 2003).

Industrial practice does in fact recognize the importance of addressing software quality characteristics such as usability, performance, or reliability for product success (Berntsson Svensson et al., 2012). But, methods with which to support the achievement of product quality goals or requirements (in terms of quality characteristics) are not usually described in process models and little is known about the interaction of these methods with other process elements (Chiam et al., 2013). Reference models such as the CMMI and ISO/IEC 12207

barely address the specific practices used to improve product quality characteristics (García-Mireles et al., 2012), and there is still a need to develop process approaches with which to specify, achieve and validate product quality requirements (Allen et al., 2006). Indeed, trends as regards developing software systems require quality management tasks to deal with nonfunctional requirements such as security and performance (Breu et al., 2014).

Since addressing the quality of software can be associated with both product and process, we use the following terms throughout this paper: software quality, software product quality and software process quality. *Software quality* refers to a general view of product quality without implying that any specific approach is used to measure or evaluate it. *Software product quality* means that there is an underlying quality model that defines a set of product quality characteristics. *Software process quality*, on the other hand, refers to the capability of a software process to produce software with increased quality.

Given the importance of research on software quality, and the acknowledgment that a software process can be a systemic approach with which to enhance product quality (Aaen et al., 2001), we have carried out a systematic mapping study (SMS) with the aim of identifying those papers whose purpose is to enhance software product quality within SPI initiatives based on process reference models. The SMS may additionally provide useful information about the strategies applied, the product quality characteristics addressed and the

* Corresponding author. Tel.: +34 926 295300 ext. 96682.

E-mail addresses: mireles@mat.uson.mx, mireles@gauss.mat.uson.mx, Gabriel.Alberto.Garcia@alu.uclm.es (G. A. García-Mireles), MaríaAngeles.Moraga@uclm.es, Felix.Garcia@uclm.es, Mario.Piattini@uclm.es (M^a Ángeles Moraga).

¹ Tel.: +34 926 295300 ext. 96682.

impact of software process reference models on such characteristics. The SMS may also identify research gaps that could be addressed by researchers.

The remainder of this paper is structured as follows. Section 2 presents an overview of related work, while Section 3 describes the method used to conduct the mapping study. Section 4 depicts the results of the mapping and Section 5 presents the discussion. Section 6 presents the study limitations. Finally, Section 7 depicts the conclusions and future work.

2. Related work

In this section we provide an overview of the main background to the SMS conducted, along with related systematic literature reviews and/or SMS in the field of SPI and software product quality.

2.1. Background

In the software industry, the main quality approaches applied in order to improve software quality correspond to the manufacturing view and the product view (Berander et al., 2005). In a *manufacturing view* the quality focuses on conformance to specification and adherence to a software process (Berander et al., 2005). Concepts such as defect counts, failure rates and rework costs are common when researchers or practitioners seek objective evidence with which to measure quality. Quality improvement is achieved when the project performance obtains a reduction in software defects (Agrawal and Chari, 2007).

Studying the defects and their relationship with software quality is a common practice in the SPI field. For instance, Harter et al. (2000) reported that in customer oriented software development, an increase in software process maturity also increases product quality, measured as the errors uncovered during system and acceptance testing. Counting defects is useful as regards identifying improvement areas within the software process, making decisions about testing activities, and monitoring the effects of process changes (Kitchenham and Pfleeger, 1996).

The measurement of defects can therefore be used to assess either process quality or product quality (Unterkalmsteiner et al., 2012). In the former case, the number of defects found by a particular testing technique is an example of a process measure, while in the latter case, the number of defects found by KLOC in source code is a product quality measure. This signifies that in order for a defect measure to be appropriately interpreted, it should describe the context in which the measurement was taken (Unterkalmsteiner et al., 2012). Nonetheless, defect counts alone, including those used in product quality, may result in either biased or restricted conclusions regarding the quality of software systems (Allen et al., 2006; Ashrafi, 2003; Harter et al., 2012).

Improvement strategies can be classified as problem-driven or norm-driven (Hansen et al., 2004). The former is focused on activities whose intention is to identify and resolve issues in the software development process, while the latter is focused on the alignment of current software development practices with an underlying model of best practices. In a similar vein, Paulk (2008) considers the analytic paradigm and the benchmarking paradigm. While the benchmarking paradigm considers best practices models specified in a reference model or standard, there is a slight difference as regards the definition of the analytic paradigm. In comparison with the problem-driven strategy, Paulk (2008) additionally addresses the principles-based approach. Approaches such as Six-Sigma or Deming's fourteen points are thus categorized into the analytic paradigm. These two paradigms or strategies can be complementary to each other. Benchmarking paradigms do not typically specify performance levels while the analytic paradigm can deal with specific quality goals in accordance with the business or project goals. Both paradigms are ultimately led by a measurement based strategy (Paulk, 2008).

In a *product view*, quality is seen as fitness for purpose. This view relies on product characteristics that meet user needs. Product quality is therefore measured by considering the task context or scenario (Kitchenham and Pfleeger, 1996). Software quality models describe a number of product quality characteristics which can be used to identify the relevant characteristics for each scenario addressed in a software project. Although several quality models have been proposed, Coté et al. (2007) pointed out that ISO/IEC 9126 (ISO, 2001) is a good model with which to address a software product quality strategy. The ISO/IEC 25010 (ISO, 2010), as a descendent of this model, can also be used with the same purpose.

In order to establish a product quality framework for this SMS, we have considered the definition of product quality found in ISO/IEC 25010 (ISO, 2010) as a base reference. Software quality is defined as the “degree to which a software product satisfies stated and implied needs when used under specified conditions” (ISO, 2010). In order to specify, measure, and evaluate those needs, a product quality model can be used. It is defined as a model which “categorizes product quality properties into eight characteristics (functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability and portability). Each characteristic is composed of a set of related sub-characteristics” (ISO, 2010). Software quality models have, however, been researched independently from the SPI field (Ashrafi, 2003).

In summary, the ultimate goal of the SPI is to achieve an optimal software process which responds to business objectives. Despite the diversity of goals that might be attained in an SPI initiative, a literature review found that software quality is the most common (Staples and Niazi, 2008). The SPI field has been strongly influenced by the manufacturing perspective in which software quality can be defined as “the density of post-release defects in a software program” (Agrawal and Chari, 2007). This view is useful as regards improving the quality of software process and product quality, but it has weaknesses when a customer's requirements are not well defined (Harter et al., 2012). Software quality, however, can be viewed from a product perspective, in which the stakeholder needs can be related to specific software product quality characteristics. Our work is thus focused on the software product quality view within SPI initiatives.

2.2. Previous literature reviews

Before carrying out the SMS we reviewed IEEE Xplore, Scopus and Web of Science digital libraries in order to identify systematic literature reviews that deal with SPI and software product quality. The list of synonyms for this methodology was adapted from de Almeida Biolchini et al. (2007). The search string used was:

“software process improvement” OR spi AND (“systematic literature review” OR “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” OR “mapping study” OR “scoping study”)

We identified two relevant papers that address software product quality within an SPI initiative. The first (Lavalée and Robillard, 2011) reviewed six journals from the Journal Citation Reports to study whether SPI initiatives imply product quality improvement from the developer perspective. These authors only considered four quality sub-characteristics from ISO/IEC 9126 in their review: suitability, accuracy, changeability and stability. The conclusion reported was that while a reduction in defects enhances product quality, it does not necessarily enhance architectural quality when factors such as changeability and stability are considered (Lavalée and Robillard, 2011).

The second review is focused on measurement and evaluation approaches applied in SPI initiatives (Unterkalmsteiner et al., 2012). The majority of the studies reported (61% of 148 papers) rely on reference models such as CMM, Six-Sigma, CMMI, PSP, ISO/IEC 15504,

TSP, QIP, Bootstrap, TQM, IDEAL and PDCA. A product quality success indicator was found in 32% of papers. Product quality can be analyzed from a general view or a specific view. The general view considers those studies (19% of 148 papers) that lack information with which to appropriately identify which quality characteristic was measured or evaluated. This category can include papers that measure software quality as a defect density in a software product. The specific view of product quality considers quality characteristic described in ISO/IEC 9126 (10% of 148 papers) and reusability (3% of 148 papers). Reliability, maintainability, and reusability were reported as being the most common success indicator for product quality characteristics. In addition, roughly a quarter (24% of 148) of the papers uses defects as a success indicator without relating them to process quality or product quality. The findings also show a clear tendency toward measuring quality at process level (57% of 148 papers). Few empirical studies report the measurement of product quality characteristics as a quality success indicator (10% of papers).

The scope of research is the main difference between this SMS and the reviews discussed. Lavallée and Robillard (2011) focus the impact of SPI on the software architecture and only consider CMMI in their analysis. The selected journals, although relevant for the SPI field, only represent a subset of the field. In addition, the paper is not explicit as regards the set of 25 empirical papers. We are, however, interested in the impact of SPI on all the characteristics described in a product quality model, such as those described in ISO/IEC 25010. Furthermore, we expect to identify the process reference models used, and not only CMMI, in order to deal with product quality and report the list of empirical papers.

The second literature review (Unterkmsteiner et al., 2012) focused on the identification of measures and success factors for SPI initiatives. This review also classifies the papers when the measures are related to process, product or organization. Overall, the review provides a comprehensive view of the measurement approaches in SPI initiatives. However, measurement is only one aspect of an SPI initiative. Our goal is to identify papers that address any of the activities considered in the SPI field whose purpose is to enhance software product quality.

3. SMS of software product quality within SPI initiatives

We carried out this SMS by considering the recommendations described in Kitchenham and Charters (2007). The changes made to guidelines with which to carry out systematic reviews (Kitchenham and Brereton, 2013) were also considered. A protocol with which to support the work of this review was built on Oct 5, 2013. The goal of this research is to provide an overview of the current approaches used to enhance software product quality in the context of SPI initiatives based on process frameworks. The SMS also considers those papers which discuss the impact of SPI on product quality.

3.1. Research questions

The main research question is:

- Which approaches have been proposed or implemented in SPI initiatives in order to enhance software product quality? What results have been obtained?

An SMS was therefore conducted in order to answer this question. The papers selected were reviewed in two phases. The first phase classified the papers by considering the quality characteristic/quality model addressed, process reference model used, and the type of research contribution according to the types provided in Wieringa et al. (2006). The type of research contribution was also used to breakdown the set of primary papers into theoretical and empirical categories. The latter category is composed of both evaluation and validation articles types, according to the Wieringa taxonomy. The selected papers

were also classified by considering their purpose and the potential impact on the Plan-Do-Check-Act (PDCA) improvement cycle (Deming, 1992). The second phase analyzed empirical papers identified in the previous phase in order to discover their rigor and relevance using the model proposed in Ivarsson and Gorschek (2011). Furthermore, the definition of quality characteristics was reviewed and the stakeholder's quality perspective was also identified. Finally, the impact of the SPI initiative on software product quality was considered.

In order to achieve the main objective of the first phase, we needed to answer the following questions:

- RQ1. What types of research have been reported to address software product quality in SPI initiatives?
- RQ2. What software product quality characteristics are managed in SPI initiatives?
- RQ3. What process reference models are considered in SPI initiatives that deal with product quality?
- RQ4. What activities in the PDCA cycle have been covered by the primary papers selected?

The classification of empirical papers led us to identify the process reference models researched in order to improve a software product quality characteristic. An SPI initiative can be broken down into various stages. Each stage requires specific activities to be executed using supporting models and methods. The research approach may also show the maturity of the SPI field with regard to product quality characteristics.

The objective of the second phase was to answer the following questions:

- RQ5. What is the rigor and relevance of studies reported?
- RQ6. How are quality characteristics related to software product defined in SPI initiatives?
- RQ7. What stakeholders are considered in the SPI initiative?
- RQ8. What is the impact of SPI on product quality characteristics?

3.2. Search process

We applied an automatic search process to select relevant papers. This approach has been used in other literature reviews on software process improvement (Pino et al., 2008). The search process takes into account the following resources: ACM Digital Library, IEEE Xplore, Scopus and Web of Science. The records retrieved from the last three resources were processed using EndNote. The records retrieved from ACM were processed manually. Although the search process began in October 2013, and the first version of this paper was based on articles retrieved in February 2014, the final list of retrieved papers includes those obtained in October 2014. The search process is presented in Fig. 1.

The search string consists of two groups of terms: software process improvement and product quality characteristics (Table 1). Since the terms used in systematic literature reviews (Section 2.2) for software process improvement are very broad and depend on the research objectives, we based this part of the search string on Pino et al. (2008). In fact, other literature reviews have also used this search string to present an overview of the SPI field (Hansen et al., 2004; Müller et al., 2010).

The second group of terms relies on the product quality characteristics described in the ISO/IEC 25010 and ISO/IEC 9126 software product quality models. We also reviewed two literature reviews to identify relevant terms for synonyms for the quality characteristic concept (Barney et al., 2012; Svensson et al., 2010). The generic terms for software quality can, on the other hand, support the identification of papers addressing others product quality characteristics, or sub-characteristics, which were not explicitly included in the search string.

The procedure used to retrieve records from digital libraries was performed in two steps. The first step, which we called automatic

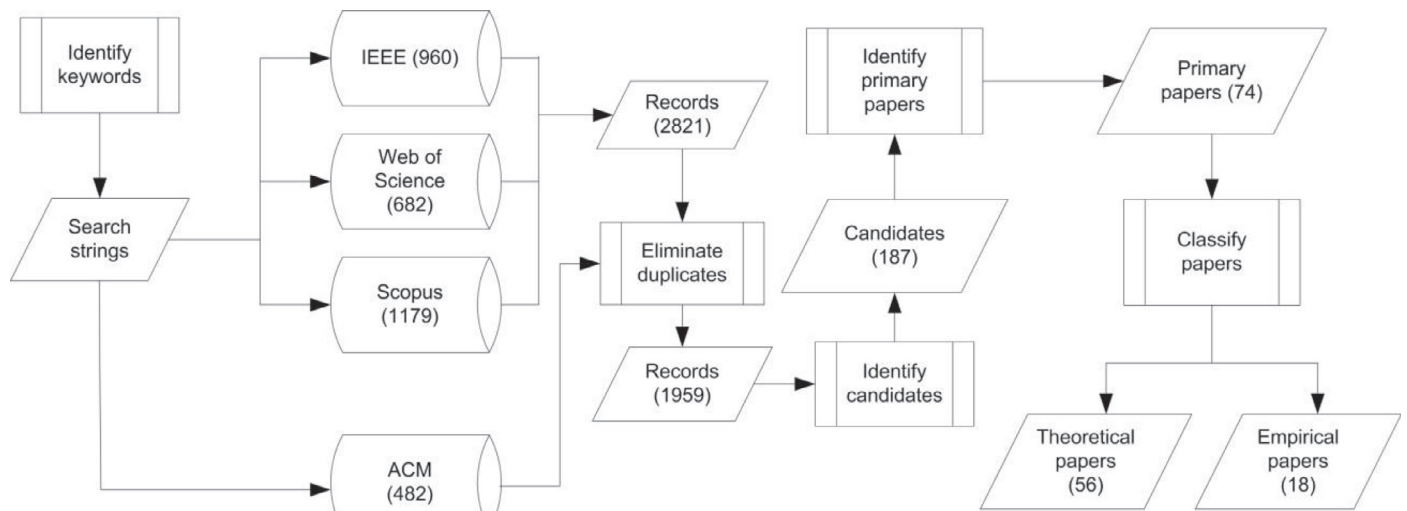


Fig. 1. Process for the SMS.

Table 1
Terms for the search string.

Terms	Synonyms
Software process improvement	"software process improvement" OR cmm OR cmmi OR 15504 OR spice OR 12207 OR 9001
Product quality characteristics	9126 OR 25010 OR usability OR maintainability OR security OR efficiency OR portability OR reliability OR functionality OR compatibility OR "non-functional requirements" OR "nonfunctional requirements" OR nfr OR "quality characteristics" OR "quality factors" OR "quality requirements" OR "quality attributes" OR "software quality" OR "product quality"

processing, extracted 2821 records from IEEE, Web of Science and Scopus databases. These were then processed in a reference tool to eliminate duplicates. The second step was the manual processing of the 482 records extracted from ACM in order to identify duplicated records.

3.3. Selection criteria

The records retrieved were reviewed against the selection criteria described in the review protocol. The inclusion criteria applied were:

1. The article must be written in English and have been published including December, 2013.
2. The article must address the topic of software process improvement and software quality by considering the product perspective. This includes papers which directly or indirectly support any of the activities considered in the SPI. Direct activities are those described in SPI methods which assess or change a software process in an organization. Indirect activities are those which provide support for an SPI initiative such as the development of reference models, the adaptation of reference models, assessment models and methods, and the evaluation of process frameworks with regard to software product quality.
3. The article's abstract must include a quality characteristic, or sub-characteristic, such as those described in ISO/IEC 9126 or ISO/IEC 25010. It may also reference a product quality model or product quality term such as quality goals, quality requirements or quality goals (Table 1). The decision to include generic product quality terms was made because we reviewed the paper's introduction in order to identify any product quality characteristic or an approach used to deal with it.
4. The articles must include a process reference model such as CMM, CMMI, ISO/IEC 12207, among others.
5. The article must be published in a journal, conference proceeding or workshop.

The exclusion criteria applied were:

1. The article addresses the manufacture view of quality. Product quality is measured as the number of defects found during the software development lifecycle. This view lacks information about the specific product quality characteristics enhanced as explained in Section 2.1.
2. The article addresses a proposal for educational settings.
3. The article addresses issues/proposals for either process management or software process quality.
4. The article only describes improvement results in terms of productivity or reduced cycle time.
5. The article addresses characteristics described in the quality in use model (ISO, 2010). We are interested in understanding how process reference models can impact on quality during software development rather than their effects when software is in the operation stage.
6. Literature reviews about software process improvement.
7. Forwards, editorials, posters, books and theses.
8. Articles with an identical or similar contribution.

3.4. Selection procedure

The selection criteria were applied by the first author when reviewing the 2441 articles. Most of papers were discarded after reading only the title because the topic appertained to other fields such as electronics, networks or cryptography. After reading the abstract, we identified 187 candidates. These were read again to identify whether they complied with the inclusion criteria, and in some cases we also read the paper's introduction to make a decision. The selection process led to 74 primary papers being obtained. A random sample of 200 records was additionally checked by the second author in order to verify the selection process. The results of this procedure were consistent with the outcomes reported by the first author.

Table 2
The manual search procedure results.

Title	Date range	Records	Candidate	Primary	Empirical
PROFES	2005–2013	402	4	0	0
EUROSPI	2005–2007, 2010–2013	186	9	4	0
SPICE	2011–2013	100	3	1	0
Journal of Systems and Software	1997–2013	2442	3	2	2
Journal of Software: Evolution and Process	2012–2013	121	0	0	0
Software Process: Improvement and Practice	2003–2009	261	0	0	0
IEEE Software	1997–2013	1824	0	0	0
Information and Software Technology	1997–2013	1571	3	1	1
Software Quality Journal	1997–1999, 2001–2013	380	4	3	3
Totals		7287	26	11	6

In order to validate the set of 74 primary papers identified above, a manual search was conducted. The use of a collection of known studies is recommended when validating the search process (Kitchenham et al., 2011b; Zhang et al., 2011). The set of these papers were extracted from articles published in conference proceedings and journals addressing software process improvement topics (Table 2). Some journals are general to the software engineering field.

The journals and conference proceedings used in the manual search should also be accessed by the databases used in the automatic search. The Scopus database was therefore selected in order to ensure that the outcomes would be comparable. The name of the conference or the title of the journal was used with an appropriate command from Scopus. The range of years was based on the result of the automatic search process, from 1997 to 2013.

As a result, Scopus retrieved 7287 records from three conference proceedings and six journals. The range of years that can be accessed by Scopus also is reported for each venue. Twenty-six papers were selected as candidates using inclusion criteria. After applying inclusion and exclusion criteria, 11 primary papers were selected. Of the set of primary papers, six belong to the empirical category.

Empirical papers found in the manual search also appeared in the outcome of the automatic search process. Of the primary papers identified, only one was not included in the theoretical group of selected papers.

The manual search process found 11 papers, and 10 papers were selected in the automatic search process. After applying the sensitivity formula described by Kitchenham et al. (2011b), a sensitivity score of 90.9% was obtained.

3.5. Data extraction

In order to answer the research questions we extracted the following information from each article:

- Quality characteristic.** The quality characteristic described in the paper. In some cases a quality model or generic quality terms are reported (such as those depicted in Table 1).
- Approach for SPI.** We identified several types of approaches for SPI. In order to classify the main contribution of each paper, we used the PDCA stages (Deming, 1992) to formulate a categorization scheme as regards whether the papers discuss an SPI initiative in an organization or they discuss either a model or method supporting each stage:
 - Plan.** The papers discuss improvement methods, an SPI infrastructure, critical factors for implementing an SPI initiative, aligning business needs with process improvement goals, coordinating multiple SPI initiatives and mappings between process models.
 - Do.** The papers present solutions for improvement opportunities. These can be process models applied at operative level, measurement process, pilot projects and the respective outcomes, and the deployment of improvement activities throughout an organization.

c. **Check.** The papers present both models and methods with which to assess the software process, assessment results, and a profile of assessment models and methods. This category includes maturity models and any other process reference models that are used to compare the practices that a software organization carries out.

d. **Act.** The papers discuss approaches with which to improve the SPI process carried out in the organization, to analyze organizational lessons learned, surveys from literature or organizations with regard to SPI efforts, adoption of reference process models in industrial settings, and experiences in continuous process improvement.

- Process framework.** We identified the process frameworks, or SPI model, explicitly included in the article.
- Research approach.** Research approaches are opinion papers, philosophical papers, experience reports, proposals of solutions, validation research and evaluation research (Wieringa et al., 2006). The first four research approaches were considered to be theoretical papers, while the latter two were treated as empirical papers.
- Venue.** This describes whether the paper was published in a journal, conference or workshop.
- Year.** Year of publication.

The second part of the SMS analysis takes into account the empirical papers, which are classified as validation or evaluation. A validation type article “investigates the properties of a solution proposal that has not yet been implemented in practice” (Wieringa et al., 2006) and the paper discusses the research design and results. An evaluation type paper investigates a problem in industrial practice or an implementation of a method or technique in industrial settings. It also includes the discussion of the research design and results (Wieringa et al., 2006). We therefore extracted the following information from the empirical papers: goal of study, definition of quality characteristics, kind of stakeholders involved, approach used to improve product quality, and aspects used to assess rigor and relevance (Ivarsson and Gorschek, 2011).

Two authors independently performed the classification of 74 primary papers as regards the research approach using the Wieringa taxonomy (2006). The classification had an agreement of 78.3%. The kappa value obtained was 0.634. Since this is a fairly good agreement, the remaining 16 papers upon which an agreement had not been reached were reviewed using a detailed guideline. The guidelines were developed on the basis of reviews of papers that did not fit into the same category.

A paper can include elements of diverse categories, e.g. it can present both a proposal and the validation in industrial settings. It can therefore be classified in several categories (Wieringa et al., 2006). If a paper includes empirical evidence, the decision as to which category to place it in tends toward papers of the validation or evaluation types. A further disagreement was based on how mappings between process models are categorized. These papers were categorized as theoretical proposals. If the mapping paper includes a validation in an industrial setting it could therefore be considered in the validation-type papers.

On the other hand, the validation category considers those papers that address a case carry out in industry, although few contextual data are provided. The list of arguments used to discuss the most appropriate category was extracted from the abstract, paper content, and classification of papers provided by Scopus. The list of arguments was reviewed by the second and third authors in order to make a decision by consensus.

The first and the third author categorized the empirical papers using the rubrics for rigor and industrial relevance (Ivarsson and Gorschek, 2011). The first comparison of both types of independently extracted data attained an agreement of around 66% for rigor and 55% for relevance. The kappa value was 0.5518 for rigor while that for relevance was 0.2763. Additional data from empirical papers were therefore extracted in order to improve this classification. The rigor rubric was improved by adding the contextual facets (Petersen and Wohlin, 2009). In order to assess the context as a strong description it should describe at least four features from the contextual facets and two from the product facet. A medium description includes two to three features and a weak description describes up to one feature of any contextual facet.

The study design was organized as aim/hypothesis, sampling, instrument design and procedure. Threats to validity were composed of external, internal and other types (e.g. construct). A strong description of a study design includes at least three elements from the study design. A medium description includes up to two elements. On the other hand, a description of validity must consider at least two main validity threat categories such as, for instance, internal and external, if it is to be considered as strong. Otherwise, a medium description only considers one of the main validity threat categories. The relevance rubric can be applied as is in this SMS: It is only necessary to identify that relevant elements are presented in the empirical paper.

The first author extracted data using the new template and the rigor elements were filled in as regards the content of the empirical paper under review. The 18 templates extracted were compared with the third author's comments in order to reach an agreement regarding the rigor and relevance of the empirical papers.

4. Results of the SMS

We identified 74 primary papers which were classified by publication year. Fig. 2 shows publications trends. The graph also shows the trends of the empirical papers reported, which represent 24.3% of all the papers selected. The year in which most articles were published was 2009, and the main topics are mappings among process reference models and approaches used to deal with security.

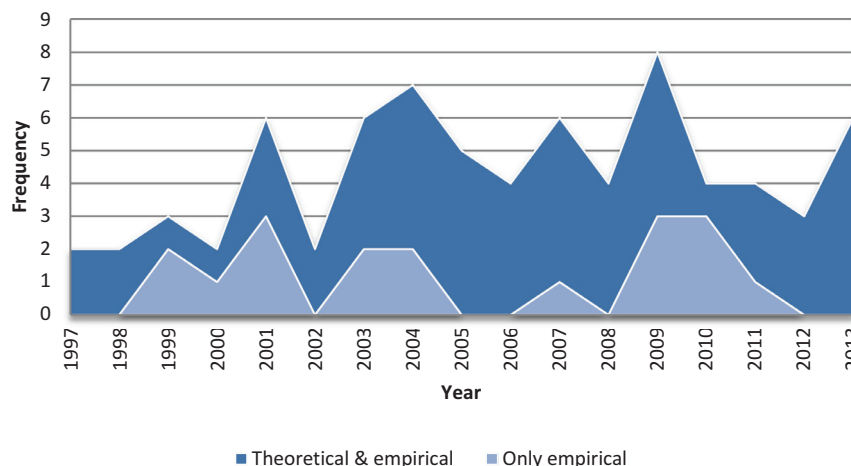


Fig. 2. Papers published by year.

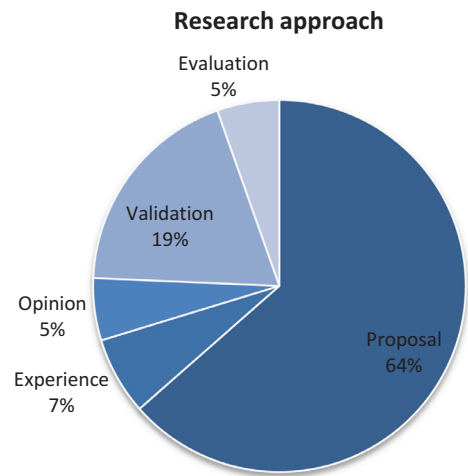


Fig. 3. Research approach found in articles.

Table 3 depicts the empirical papers. Some empirical papers recommend either a product quality model or a process reference model that could be used in an SPI initiative. However, the papers lack validations of these recommendations (Asthana and Olivieri, 2009; Boegh et al., 1999). These papers were included because they provide an approach that can be used to improve or monitor product quality goals.

The empirical papers were published in several conferences and journals. The articles published in conferences and workshops represent 73% of all papers. The conferences which contain most papers are PROFES (6), ICCSA (3), EuroSPI (3), and SPICE (2), among others. Upon considering journal papers, which represent 28%, they are published in Software Quality Journal (4), Journal of System and Software (2) and Information and Management (2), Information Management & Computer Security (2), Journal of Computer Science (2) among others. These conferences and journals, however, represent roughly a third of the primary papers.

4.1. Research approach (RQ1)

We classified the selected papers with regard to the main research approach reported in the article using the taxonomy described in Wieringa et al. (2006). We found that 64% of the papers belong to the proposal category. Empirical studies represent 24% of the papers (evaluation 5% and validation 19%). The distribution of the papers by research approach is depicted in Fig. 3.

Table 3
List of empirical papers. QC/QM: quality characteristic/quality model.

PaperId	Paper type	PDCA stage	PDCA topic	QC/QM	Process reference/SPI models
asthana2009 (Asthana and Olivieri, 2009)	Validation	Do	Measures	Reliability	CMMI
boegh1999 (Boegh et al., 1999)	Validation	Do	Measures	ISO9126	CMM, ISO9001 and SPICE
nunes2010 (Nunes et al., 2010)	Validation	Do	Developing a process	Security	SSE-CMM, CC, ISO27002, OCTAVE
ortega2003 (Ortega et al., 2003)	Validation	Check	Assessment	ISO9126, Dromey, McCall	SPICE
wagner2011 (Wagner et al., 2011)	Validation	Do	Developing a process	Security	SSE-CMM
zia2010 (Zia, 2010)	Evaluation	Act	Adoption of SPI practices	Security	CobIT, ISO27002, COSO
zucatto2007 (Zucatto, 2007)	Validation	Do	Developing a process	Security	SSE-CMM
ardimento2004 (Ardimento et al., 2004)	Validation	Do	Measures	Product goals	Method used in ISO9001 certified organization
ashrafi2003 (Ashrafi, 2003)	Evaluation	Act	SPI experience outcome	Handbook of SQA	CMM, ISO9000
balla2001 (Balla et al., 2001)	Validation	Act	SPI experience outcome	Quality product	ISO9000
chen2009 (Chen and Huang, 2009)	Evaluation	Act	SPI experience outcome	Maintainability	CMMI, ISO9001
kuilboer2000 (Kuilboer and Ashrafi, 2000)	Evaluation	Act	SPI experience outcome	Handbook of SQA	CMM, ISO9000
trienekens2001 (Trienekens et al., 2001)	Validation	Do	Developing a process	ISO9126	Generic process model
vanLatun2000 (van Latun and van Uijtrecht, 2000)	Validation	Do	Pilot project outcomes	Reliability, fitness for use, and predictability	BOOSTRAP
vanSolingen1999 (van Solingen et al., 1999)	Validation	Do	Pilot project outcomes	Reliability, fitness for use, predictability, maintainability	BOOSTRAP
winter2010 (Winter and Rönkkö, 2010)	Validation	Do	Measures	Usability	ISO13407, ISO9241-11
oconnor2009 (O'Connor, 2009)	Validation	Act	Adoption of SPI practices	Usability	Generic process model
zhou2004 (Zhou and Stålhane, 2004)	Validation	Act	Adoption of SPI practices	Reliability	Generic process model

4.2. Quality characteristics addressed (RQ2)

In order to understand how quality characteristics are addressed, we classified the primary papers by taking into account the main quality characteristic considered. We split the results into two groups: individual quality characteristics which correspond to ISO/IEC 25010 or ISO/IEC 9126 (53 papers, 71.62%), and papers that present other approaches (21 papers, 28.38%). It is important to note that the papers which address more than one quality characteristic, without referencing a product quality model, are the least frequent (5 papers, 6.75%).

The most frequent quality characteristics from ISO/IEC 25010 or ISO/IEC 9126 addressed by the primary papers were security (69%), reliability (12%), usability (14%) and maintainability (5%). When the empirical papers were analyzed we found a similar distribution: security (33%), reliability (33%), usability (17%), and maintainability (17%). The main difference between theoretical and empirical papers concerns security quality characteristic (Fig. 4). This can be explained by the number of mapping studies published that deal with various security reference models without validating the mapping outcomes.

Other primary papers address quality models (21 papers, 28%). The distribution of these models is depicted in Fig. 5. The ISO/IEC 9126 is the main quality model used (42%). Other quality models are also addressed, which are principally the following: ISO/IEC 25010, Handbook of SQA (HSQA) and McCall model.

The category named *Others* (Fig. 5) includes those papers which consider several product quality characteristics that do not fit well into the aforementioned categories. For instance, Fonseca et al. (2005) deal with mappings between process standards that consider RAMS (reliability, availability, maintainability, and security) as a set of quality characteristics to be addressed by railway systems. A similar situation occurs in the paper by von Wangenheim et al. (2012) which points out that usability, portability, security and safety are relevant quality characteristics for telemedicine systems. The other two papers include fitness for use and predictability in addition to reliability. Finally, three primary papers were not included in this classification as they do not consider specific quality models.

As can be observed in Fig. 5, we found nine papers that address ISO/IEC 9126. Al-Qutash (2009) performs a mapping between ISO/IEC

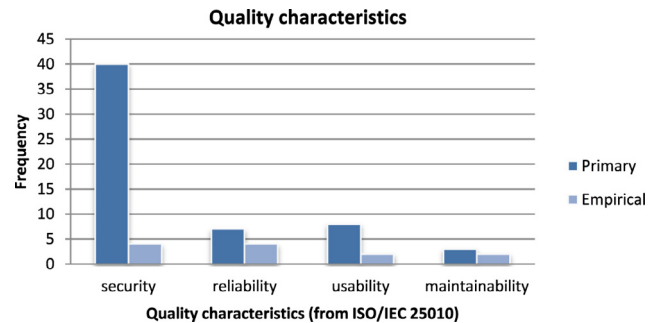


Fig. 4. Quality characteristics addressed by papers selected.

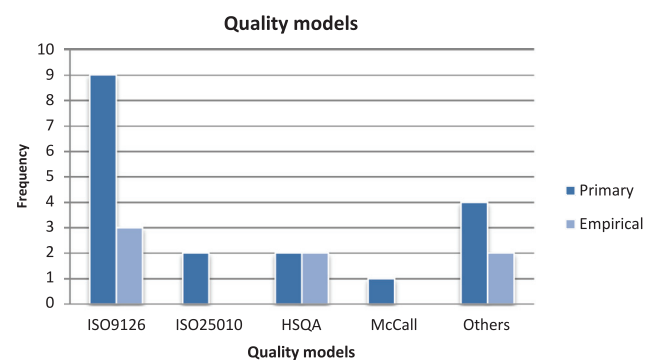


Fig. 5. Quality models used in primary papers.

9126 and ISO/IEC 12207 to identify cross-references between processes and measures from an internal quality model, an external quality model and quality in use. Boegh et al. (1999) suggest that a specific quality model for a system component can be based on the quality characteristics and measures from ISO/IEC 9126. Chandra et al. (2011) define a measurement process based on GQ(1)M, ISO/IEC 9126 and CMMI. Ortega et al. (2003) propose a systematic quality model and they develop a process framework with which to evaluate product quality (ISO/IEC 9126 is the main product quality

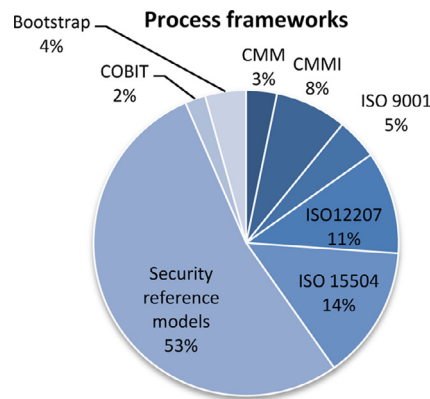


Fig. 6. Process frameworks used in primary papers.

model), considering the processes that contribute to product quality (from ISO/IEC 15504). Tzeng (2012) propose the use of fuzzy Quality Function Deployment (QFD) to identify quality requirements (five from ISO/IEC 9126) and seven processes from ISO/IEC 12207 which have an impact on quality improvement in information system maintenance. Hamann et al. (1998) describe a method that can be used to develop a product/process dependency model. GQM is used in combination with ISO/IEC 9126 for the hypothesis generation, analysis and validation of product/process dependency (Hamann et al., 1998). Shubhamangala et al. (2013) use ISO/IEC 9126 as to relate quality attributes with defect severity.

Trienekens et al. (2001) describe the experience of using the model product focused SPI (P-SPI). The main conclusion is that P-SPI put products to be developed in a central position in improvement programs. As a result, the specific quality goals of a company and project could be fulfilled. Satpathy et al. (2000) propose a model with which to improve product quality by integrating activities from reference models and product quality characteristics. They also include a measurement approach based on GQM.

ISO/IEC 25010 is used by García-Mireles et al. (2012) to map quality characteristics within practices described in both ISO/IEC 12207 and CMMI. The Handbook of SQA (HSQA) quality model is used to request the effect of ISO 9001 or CMM on product quality (Ashrafi, 2003; Kuilboer and Ashrafi, 2000).

In summary, when researchers cite several quality characteristics, they are considering two main approaches: mappings between reference models and measurement processes. On the one hand, mappings are carried out in order to propose a software process, assessment model or maturity model. On the other hand, quality models are used to identify relevant quality characteristics and their related measures in order to evaluate product quality. Finally, some articles also use generic quality terms, such as product goals or quality goals to propose a measurement process with which to monitor process or product quality.

4.3. Process frameworks (RQ3)

Since one of the main approaches used to deal with software process improvement is to use a process reference framework (Hansen et al., 2004), we identified the main process frameworks used in SPI initiatives that are focused on enhancing product quality (Fig. 6). More than half of the primary papers deal with security issues (53%), and we have therefore identified a category namely *security reference models* which includes: SSE-CMM, Common Criteria (CC), ISO/IEC 27002, and ISO/IEC 21827. ISO 9001, ISO/IEC 12207 and ISO/IEC 15504 standards are used in 25% of the papers. CMM and CMMI are used in 11% of the papers. Other reference models (or assessment methods) are Bootstrap and COBIT.

Process frameworks in empirical papers

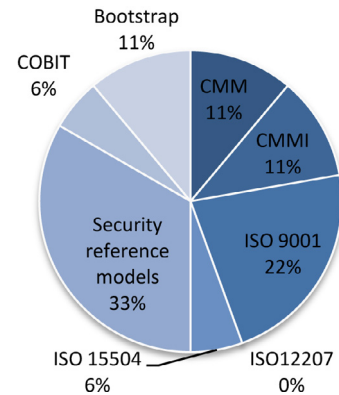


Fig. 7. Process frameworks used in empirical studies.

PDCA stages

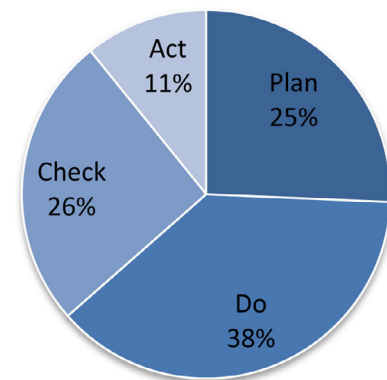


Fig. 8. Classification of papers using PDCA stages.

However, when we reviewed the empirical papers we found that security-related reference models are the most frequent (33%). ISO 9001 is used in 22% of the papers in addition to the CMM/CMMI models. ISO/IEC 15504 is cited in only 6% of the empirical papers. Upon comparing the outputs of the empirical papers with that of the theoretical papers, we can suggest that ISO/IEC 15504 and ISO/IEC 12207 are considered when designing proposals, but these proposals have not yet been validated. The security reference models also have proposals that have been not validated. The lack of empirical studies on security and ISO frameworks signifies that CMM and CMMI play a relevant role in this graph (Fig. 7).

There is a trend in the SPI field to focus on models such as CMMI, ISO/IEC 15504 and ISO 9001 (Pino et al., 2008; Unterkalmsteiner et al., 2012). Given the relevance of these models, we identified the product quality characteristics addressed by them. We found that CMM has been used with security practices and the HSQA quality model. CMMI has been used with security, ISO25010, reliability, availability, and maintainability. On the other hand, proposals based on ISO/IEC 12207 consider security, usability, ISO/IEC 9126, ISO/IEC 25010 and generic quality terms. ISO/IEC 15504 was used to propose the enhancement of security, usability, reliability, and ISO/IEC 9126.

4.4. Classification of SPI approaches used to deal with product quality (RQ4)

Using the PDCA stages as a classification scheme (Section 3.4), we found that the set of primary papers covers all the stages (Fig. 8):

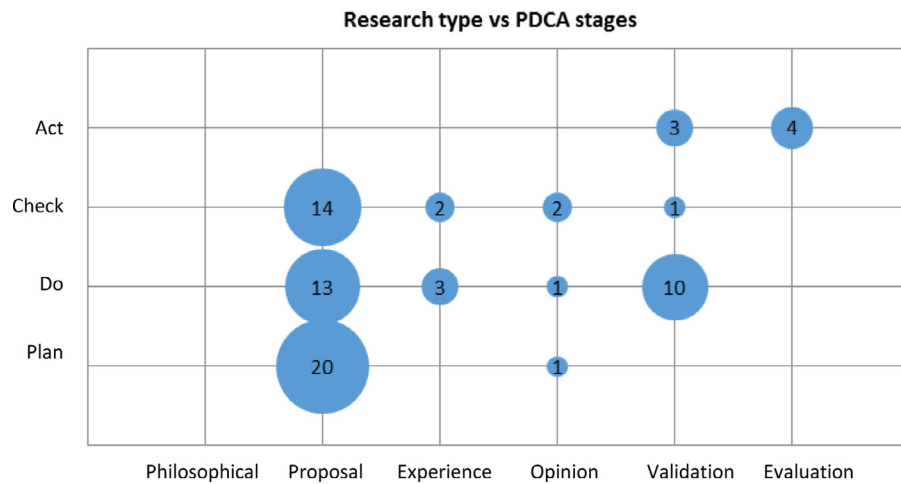


Fig. 9. Distribution of research papers by PDCA stages.

the *Plan stage* (25%) principally contains proposals for a particular form of framework mapping (Halvorsen and Conradi, 2001). There are different degrees of comparison, from a discussion on a standard's purpose to a detailed mapping between practices. Mapping has also been used to create a new process model which combines practices from various reference models in an integrated proposal. Finally, some proposals extend a process reference model, CMMI or ISO/IEC 15504, principally to address security issues.

The main topics in the *Do stage* are how to develop a suitable process model in order to carry out the activities in a software project (at an operative level) and the development of processes and methods with which to measure product quality. The measurement proposals deal with processes that can be used to support the monitoring and quality control during the software development lifecycle. The formulation of appropriate measures can be carried out with GQM. There are other topics such as introducing specific practices from other disciplines (e.g., Reliability Engineering), outcomes from pilot SPI projects, tools to improve product quality, and the extension of agile methods in order to support security and usability.

Almost half the papers in the *Check stage* address the development of assessment models by considering standards and guidelines for security practices (Cheng et al., 2007). The assessment models may be maturity models. However, only two papers consider the quality characteristics described in ISO/IEC 9126 (Ortega et al., 2003; Tzeng and Chen, 2012). The assessment methods, on the other hand, include questionnaires and some mechanisms with which to make decisions, such as measures, the Analytical Hierarchy Process (AHP) or Quality Function Deployment (QFD) (Cheng et al., 2007; Tzeng and Chen, 2012).

The *Act stage* (10%) contains papers which address either the adoption of practices in order to improve specific product quality characteristics or process performance with regard to quality characteristics. The first group discovers specific practices with which to enhance either usability or reliability in web systems that software organizations implement in their software projects. There is also a paper that surveys organizations in order to understand the extent to which security reference models are deployed in industry (Zia, 2010). The last group of papers discusses findings concerning the perception as regards software quality in organizations that implement process reference models. Since all the papers in this category also correspond to the evaluation research type, they will be reviewed in Section 4.8.

The bubble graph (Fig. 9) shows the distribution between research type and the PDCA-based categories. The majority of the papers corresponds to the proposal category and deal with topics in the *Plan*, *Do* and *Check* categories. The graph shows a lack of evaluation studies for practically all PDCA stages.

4.5. Rigor and relevance of primary papers (RQ5)

In order to assess the quality of the empirical papers we used the method proposed by Ivarsson and Gorschek (2011) to evaluate the rigor and industrial relevance of technology evaluations. Rigor refers to both how an evaluation is performed and how it is reported. The framework is focused on the extent to which aspects related to rigor are presented. The aspects used to evaluate rigor are: the extent to which context, study design and validity issues are described. These aspects are scored using an ordinal scale: weak (0), medium (0.5), and strong (1). Relevance is, meanwhile, focused on the potential use of research results in industrial settings and it considers that the environment in which the results are obtained influence the relevance of using a research result in an industrial context. The aspects taken into account are: subjects, scale and context. The scoring scale considers two values: contributes to relevance (1) or does not contribute (0).

The 18 empirical papers were assessed using the framework proposed by Ivarsson and Gorschek (2011) in order to describe the rigor and relevance of the articles. Data were extracted from each empirical paper to determine the extent to which each rigor and relevance factor was described. Table 4 shows the results. The mean for rigor is about 1.30 which can be interpreted as the fact that papers report little information about the method employed to perform the study. On the other hand, the relevance data show that research has been conducted in an environment similar to an industrial context (mean of around 3.16). However, the lack of rigor reported hinders the possibility of either achieving reliable conclusions or generalizing the results. Fig. 10 presents a bubble chart showing the relationship between rigor and relevance.

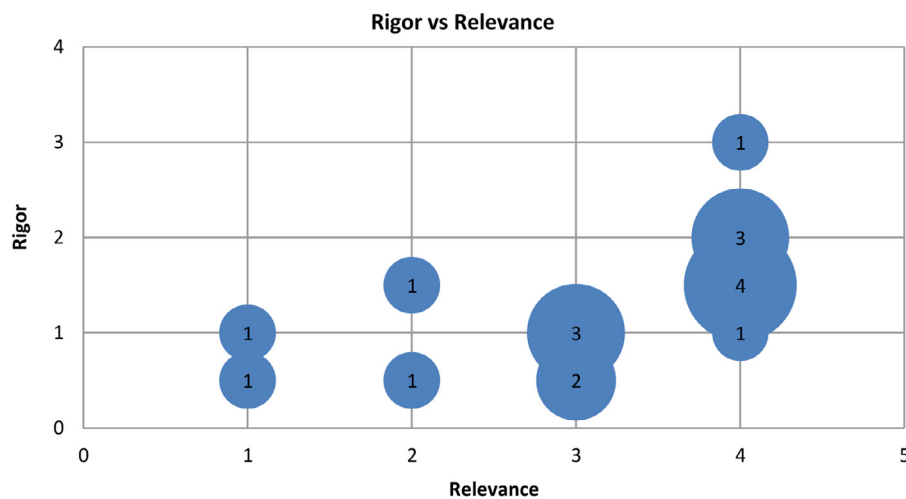
4.6. Definition of quality characteristics (RQ6)

Since defining and measuring quality is a difficult endeavor, we reviewed how the empirical papers deal with the quality concept. We found that a third of the papers lack a definition of the software quality characteristic being studied. Papers discussing security-focused processes only make reference to standards and guidelines such as the SSE-CMM and Common Criteria (Wagner et al., 2011). Other research takes a practical definition of a quality characteristic, such as reliability, and relates the definition to any form of measure in a software project, such as defect trending graphs or the number of defects in operational software (Balla et al., 2001; van Latum and van Uijtregt, 2000).

Some proposals use the definition of quality characteristics as they are described in standards. The models used to define quality characteristics are the ISO/IEC 9126, ISO 9241, glossary of terms in IEEE, and

Table 4
Rigor and relevance of empirical papers.

PaperId	Context	Study design	Validity	Rigor	Subjects	Context2	Scale	Research method	Relevance
ardimento2004	0.5	0.5	0.5	1.5	0	1	1	0	2
ashrafi2003	0.5	0.5	0.5	1.5	1	1	1	1	4
asthana2009	0.5	0	0	0.5	0	1	0	0	1
balla2001	1	0.5	0	1.5	1	1	1	1	4
boegh1999	0.5	0.5	0	1	0	0	0	1	1
chen2009	1	1	1	3	1	1	1	1	4
kuilboer2000	0.5	0.5	0.5	1.5	1	1	1	1	4
nunes2010	0.5	0.5	0	1	1	1	0	1	3
oconnor2009	1	0.5	0.5	2	1	1	1	1	4
ortega2003	0.5	0.5	1	2	1	1	1	1	4
trienekens2001	0.5	0	0	0.5	1	1	1	0	3
vanLatum2000	0.5	0	0	0.5	1	1	1	0	3
vanSolingen1999	0.5	0	0	0.5	0	1	1	0	2
wagner2011	0.5	0.5	0	1	0	1	1	1	3
winter2010	1	0.5	0	1.5	1	1	1	1	4
zhou2004	1	1	0	2	1	1	1	1	4
zia2010	0.5	0.5	0	1	1	1	1	1	4
zucatto2007	0.5	0.5	0	1	0	1	1	1	3

**Fig. 10.** Rigor and relevance of empirical papers.

Handbook of SQA (Table 5). Some proposals recommend using the quality characteristics as a basis to identify relevant quality project goals (Boegh et al., 1999), while others recommend establishing target values in order to identify quality characteristics and to support the specification of quality requirements (Balla et al., 2001; Boegh et al., 1999).

4.7. Participating stakeholders (RQ7)

The fitness for use quality perspective considers that different stakeholders have specific quality needs. It is thus relevant to understand how the stakeholders have been addressed in empirical papers (Table 5). The empirical studies consider case studies, the action research approach and surveys in order to either validate their respective proposals or evaluate the impact of SPI models on product quality. A third of the papers, however, do not include a description of the stakeholder sample. Another third take into account managers and software developers. Finally, the remaining studies report either managers or software developers. The majority of the studies do not discuss the impact on software quality by considering the stakeholder profile. The stakeholder profile is mainly used to establish sample demographics (Ashrafi, 2003; Chen and Huang, 2009).

Only two empirical papers assign different activities to stakeholders by considering their respective role. Ortega et al. (2003) use the instrument for assessing software quality by considering the roles

of stakeholders. Project managers' answer questions about software process activities, users answer questions about reliability, usability and efficiency, while developers can answer questions about functionality, reliability, usability, maintainability, efficiency and portability. In a similar vein, the usability framework (Winter and Rönkkö, 2010) considers users as usability testers who carry out the scenarios described in each testing procedure and answer questionnaires in order to assess the usability of a software product; usability experts analyze quantitative and qualitative data and provide the project team with feedback regarding measures and experience-based suggestions. These articles, however, only describe the experience of developing assessment proposals without discussing the effect of methods on different stakeholders.

4.8. Impact of SPI on product quality (RQ8)

To answer this question, we reviewed the surveys focused on evaluating the impact of SPI on product quality. We found three papers that analyze two datasets. Kuilboer and Ashrafi (2000) and Ashrafi (2003) analyze surveyed data to understand the impact of CMM and ISO 9001 on product quality. Chen and Huang (2009) analyze the effect of development problems on software maintainability and discuss the effect of process maturity (based on ISO9001 or CMMI) in order to mitigate these problems.

Table 5
Stakeholders and quality characteristics addressed in SPI initiatives.

PaperID	Stakeholder sample	Quality characteristic	Quality definition	Quality approach
boegh1999	N/A	Develop a specific quality model	Specific to project needs	Measures and target values to improve requirements specification and quality monitoring
kuilboer2000	Software developers	Handbook of SQA	Handbook of SQA	Implementing SPI models moderately increases perception of product quality because the model considers specific product quality
ashrafi2003	Software developers	Handbook of SQA	Handbook of SQA	Implementing SPI models moderately increases perception of product quality because the model considers specific product quality
ortega2003	Project managers, developers and users	ISO9126, Dromey model, McCall model	ISO9126	Measures and practices to improve product quality
chen2009	Project managers, software engineers	Maintainability	ISO9126	SPI initiatives can moderately improve software maintainability because they reduce the level of documentation and project management issues. SPI projects have a significantly higher level of software maintainability than non-SPI projects
ardimento2004	N/A	N/A. Generic quality term	N/A	Measures considering stakeholder's viewpoint in GQM
balla2001	Project manager, project leader, project members	N/A. Generic quality term	N/A. Reduced number of customers' reclamations and number of nonconformities	Defining process based on SPI models and product quality requires definition of relevant attributes and measures
ashtana2009	N/A	Readiness criteria	Measure-based. Functionality, operational quality, stability and reliability	Measures from product and process to determine readiness criteria
vanLatum2000	Project manager, engineers, test team	Reliability	Lower number of defects in operational software	Project quality goals and process assessment guide the selection of practices to improve quality. GQM to establish a monitoring process
vanSolingen1999	N/A	Reliability	Lower number of defects in operational software	Project quality goals and process assessment guide the selection of practices to improve quality. GQM to establish a monitoring process
trienekens2001	N/A	Reliability	Measure-based. Number of defect graph trend	Quality requirements support the identification of measures and practices to customize software processes
zhou2004	Project managers and developers	Reliability, robustness	Reliability not defined. Robustness in IEEE Glossary	Reliability engineering provides methods to improve software reliability
nunes2010	Software developers	Security	N/A	Mappings between security-focused models can be used to develop a software process
wagner2011	Project managers	Security	N/A	Security practices from SSE-CMM can be associated with security patterns
zucatto2007	N/A	Security	N/A	Business needs drive the identification of security requirements
zia2010	Software developers	Security	N/A	Adoption of practices based on security governance models
winter2010	Software developers, usability experts, users	Usability	ISO9241	Usability measures and usability testing practices
oconnor2009	Web development managers	Usability	ISO9241	UCD practices can improve software usability

The majority of the respondents in these surveys pointed out that the project size tended to be small to medium. [Chen and Huang \(2009\)](#) reported that 76% of respondents stated that the project team size was in the range of 1–10 members and that project duration was between 1 and 2 years. [Kuilboer and Ashrafi \(2000\)](#) reported that the average range of a team was between 6 and 31 members. These authors also found that companies which tend to implement process reference models are those that develop software under contract or in-house. As an explanation, the authors argued that organizations which develop these types of projects are more interested in quality than in speeding up the process ([Kuilboer and Ashrafi, 2000](#)). However, the authors do not provide additional details in order to characterize the software. The software type is relevant since it is known that certain quality characteristics are more important than others with regard to the software type category ([Wagner et al., 2012](#)).

[Kuilboer and Ashrafi \(2000\)](#) only surveyed software developers, whereas [Chen and Huang \(2009\)](#) surveyed software developers and project managers. However, the articles do not analyze the effect of different type of participants on software projects. It is known that employees have different priorities with regard to software quality characteristics which may be influenced by the practices and compe-

tency which correspond to their respective work positions ([Berntsson Svensson et al., 2012](#)).

Process maturity is another factor that contributes toward improving software quality. [Chen and Huang \(2009\)](#) asked about certifications or maturity level. They found that organizations that develop projects with reference models have better maintainability than projects developed without reference models. [Ashrafi \(2003\)](#), on the other hand, found a wide variability in the responses from developers who use CMM and, as an explanation, argued that the process capability is a factor which may cause that variability. However, she only reported that the time of an SPI initiative is roughly three years for 75% respondents.

These studies show a moderate increase in software quality according to the participants' perception. [Kuilboer and Ashrafi \(2000\)](#) reported a moderate increase in design quality while [Chen and Huang \(2009\)](#) reported a moderate increase in software maintainability when projects are developed in an SPI environment. Factors which contribute to this perception are the emphasis on documenting intermediate products on software development, and improving the software process to meet project objectives. Although the papers do not provide project data to show a stronger conclusion about the

enhancement of product quality characteristics, perceptual attributes are important to establish a basis for a general theory (Iivari, 1996).

The ideas with regard to improving software quality are summarized as follows. We found that [Kuilboer and Ashrafi \(2000\)](#) and [Ashrafi \(2003\)](#) argued that the improvement to software quality is owing to the quality characteristics that are included in the requirements of reference models such as ISO 9001 and CMM. A reference model may be selected for the quality characteristics addressed in their requirements. Software organizations could therefore select reference models if there is an artifact that shows the effect of a process reference model on product quality characteristics. The empirical data provided ([Ashrafi, 2003](#)), however, do not sustain this proposal, since adaptation quality factors are not found in CMM and respondents reported an increased impact on quality factors from this category. On the other hand, [Chen and Huang \(2009\)](#) found that reference models can enhance software maintainability at a moderate level when problems with documentation and process management are addressed.

Other approaches with which to improve product quality within an SPI initiative are the outcomes of the PROFES project ([van Latum and van Uijtrecht, 2000](#); [van Solingen et al., 1999](#)). In this initiative, the product quality required is the driver used to decide which activities should be included in a software project. The PROFES method relies on an assessment of current practices (Bootstrap), the establishment of product quality goals using GQM, and the definition of new software process. Product-Process Dependency models are the critical artifact in this method, since they describe the potential impact that a software process activity could have on particular product quality characteristics. The method was applied to the development of embedded systems and the authors reported an improvement in the product quality and process maturity of three organizations. However, they also pointed out that it is very difficult to define and validate product-process dependency models.

Although a measurement process has the goal of monitoring a process or product, defining target values for product quality characteristics and tailoring quality models to specific product are two aspects that can contribute toward improving product quality ([Boegh et al., 1999](#); [Ortega et al., 2003](#)). Defining target values for software quality characteristics can be based on an experience repository. Software developers could therefore specify more reliable quality requirements. The tailoring of quality models meanwhile implies the need to

define a specific quality model for a product, including indicators and measures. The indicators can be a means to define and evaluate the product quality. In both cases, software developers are required to spend more time analyzing product quality requirements.

5. Discussion

The main objective of an SPI is to optimize the software process in order to achieve business needs. Enhancing software quality is one of the main arguments for starting a SPI initiative. However, we found few empirical papers that either evaluate the impact of an SPI initiative on product quality or enhance product quality within an SPI initiative. We believe that this is a very important area for further research, but the outcomes in the studies reported show that the topic is difficult to address.

The main findings of this SMS, along with some remarks, are depicted in [Tables 6](#) and [7](#). In addition to remarks for RQ1 about the type of research reported, the vast majority of the proposals show, as noted by [Barney et al. \(2012\)](#), that the area is still in the development stage. Indeed, the evidence collected does not provide conclusive results as regards the effect of SPI initiatives on product quality characteristics.

Addressing product quality characteristics within SPI initiatives (RQ2) requires software organizations to implement specific activities (or practices) which should be combined with their respective software processes. The practices might be retrieved from both industrial standards and literature on the scientific discipline that addresses the quality characteristic. In particular, a strategy based on multi-models can support the identification of the practices required when implementing solutions that address some specific quality characteristics ([Siviy et al., 2008](#)). Furthermore, organizations need support to use product quality models during the software development lifecycle. Product quality models can help in the specification of quality requirements, measurement, monitoring, and the evaluation of product quality.

When considering the process reference models used in SPI initiatives which deal with product quality characteristics (RQ3), we found that security-based reference models are the most common. This finding contrasts with existing results of other related literature reviews about SPI. For instance, [Pino et al. \(2008\)](#) found that CMM and CMMI are the most common models in SPI initiatives in small companies. [Hansen et al. \(2004\)](#) showed that the majority of papers discuss SPI

Table 6
Main findings of the first phase of review.

Findings	Remarks
RQ1. Research approaches 64% proposals, 24% empirical papers	<ul style="list-style-type: none"> • There is a lack of empirical studies dealing with SPI initiatives and product quality characteristics. • More work is required in this field to identify the relevant contextual factors to be considered when implementing an SPI initiative to improve a product quality characteristic.
RQ2. Product quality characteristics managed Security is considered in 69% of papers addressing one quality characteristic. The quality model most often used in measurement processes is ISO/IEC 9126	<ul style="list-style-type: none"> • Security, usability and reliability require specific practices to be deployed in software processes. These can be gathered from standards or from related disciplines. • Quality models are mainly used to define measurement programs or to develop questionnaires to evaluate product quality. • Although product quality characteristic definition helps to define product quality, the diversity of definitions for quality terms and quality models shows that it is difficult to define product quality in software projects objectively. In addition, this situation hinders the comparison between outcomes of SPI projects.
RQ3. Process reference models Security-based reference models are the most common (53%), followed by ISO standards (30%) and CMM/CMMI (11%)	<ul style="list-style-type: none"> • A lot of proposals for security are based on mappings between security-related standards. Some of them propose processes at the operative level. On the other hand, several proposals are based on ISO standards and others on CMM/CMMI models. Few empirical studies. • Traditional models for SPI (CMMI or ISO/IEC 15504) barely addressed when research works report improvements to product quality characteristics.
RQ4. PDCA – classification of proposals Plan (25%), Do (38%), Check (27%), Act (10%)	<ul style="list-style-type: none"> • Majority of proposals in Plan stage correspond to mappings between reference models. However, they are not validated. • Do stage contains proposals for measurement process and for tailoring software process to enhance (mainly) security aspects. • Check stage describes assessment methods while Act stage includes surveys to evaluate the impact of SPI on product quality. The latter stage requires more empirical studies to evaluate the impact of SPI initiatives on specific product quality characteristics.

Table 7
Main findings of the second part of the review.

Findings	Remarks
<p>RQ5. Rigor and relevance Rigor mean (1.30), relevance mean (3.16)</p> <p>RQ6. Definition of quality characteristics A third of papers lack definitions of product quality terms. Several quality models used</p> <p>RQ7. Stakeholder preferences Project managers and developers are the main stakeholders surveyed</p> <p>RQ8. Impact of SPI on product quality Few empirical studies analyze the impact of SPI on product quality. They report a moderate increase in maintainability and design quality factors (correctness, maintainability, and verifiability)</p>	<ul style="list-style-type: none"> • Empirical studies are focused on defining process proposals. The rigor reported is poor in the majority of empirical studies, although they were carried out in a context similar to industrial practice. • Lack of definition of quality terms makes it difficult to establish appropriate indicators and measures to provide objective evidence that SPI initiatives actually enhance a particular quality characteristic. • There is no consensus as to the role of product quality models in SPI initiatives. • Although it is known that stakeholders have different priorities as regards product quality, the empirical studies only use stakeholders' data to establish the sample profile. • There is a need to understand how SPI initiatives impact on product quality characteristics when stakeholders' preferences are taken into account. • There is a lack of empirical papers that study the impact of SPI initiatives on product quality. The existent evidence is based on perceptions of developers and project managers gathered using questionnaires and interviews. • Evidence based on measures can provide stronger evidence about the impact of SPI on product quality.

initiatives based on CMM. [Unterkalmsteiner et al. \(2012\)](#) found that CMM and CMMI are among the most common frameworks reported. Indeed, the last work, which is the most recently published, does not report security focused reference models. From a research perspective, this could lead to an in-depth study of the relationships that can be established among traditional models for SPI and specialized models such as those in the security area.

By using the PDCA cycle to classify the main contribution of the primary papers in the SPI field (RQ4), we found that the primary papers have been addressed in each stage. However, the Act stage, which requires an evaluation of the results of implementing an improvement action, requires more research to evaluate the level of improvement achieved in software product quality and the relevant contextual factors that should be considered.

The second part of the research focused on empirical papers, and we found that only 18 papers discuss issues concerning the improvement of software product quality within SPI initiatives. The majority of them show a lack of rigor in the documents reported (RQ5). This finding hinders the formulation of trustworthy conclusions, although the studies were performed in a context relevant to industrial practice.

The next two questions focused on the definition of product quality characteristics within SPI initiatives (RQ6), and the role of stakeholders when SPI initiatives impact on software quality is reported (RQ7). We found that the difficulty in defining quality characteristics is also apparent in SPI studies ([Kitchenham and Pfleeger, 1996](#)). Some papers lack a definition of quality characteristics or quality terms. Others papers make reference to standards such as ISO/IEC 9126 and ISO 9241 without providing details about how they were adapted. Since different stakeholders have distinct priorities when considering quality characteristics ([Berntsson Svensson et al., 2012](#)), we expected the empirical studies to take this aspect into account. However, the stakeholder profile is only used to describe the study sample.

Although there is little evidence to support the impact of SPI initiatives on software product quality characteristics (RQ8), we identified three main approaches that could be used in practice: capability-based approach, measurement-based approach and combined approach. The capability-based approach relies on the implementation of practices to improve software quality; the main focus of the measurement-based approach is that of monitoring quality levels using a measurement process. The combined approach uses elements from the two aforementioned approaches.

The *capability-based approach* could be generic or specific. We consider that an approach is *generic* when a software organization implements software practices described in traditional reference models, such as CMMI and ISO/IEC 15504. The documentation practices,

project management and quality assurance activities can contribute to a perception of a moderate increase in software product quality. The effects are particularly apparent in software maintainability and, to some extent, in software reliability as reported in [Ashrafi \(2003\)](#), [Chen and Huang \(2009\)](#), and [Kuiliboer and Ashrafi \(2000\)](#).

The *specific focus* attempts to identify relevant practices in disciplines related to software quality characteristics that are being improved. For instance, User Centered Design and Reliability Engineering can provide methods and techniques that could be incorporated into software development processes ([O'Connor, 2009](#); [Winter and Rönkkö, 2010](#); [Zhou and Stålhane, 2004](#)). Security practices described in standards and reference models can be tailored before they are implemented in software projects ([Nunes et al., 2010](#); [Zuccato, 2007](#)). However, the empirical data showed a low adoption level of specific practices, and some proposals require that an expert in the discipline, such as a usability expert, be part of the development team.

The *measurement-based approach* promotes the usage of measures in order to assess the quality of a software product. This can be implemented using a quality model to identify the quality characteristics and the appropriate measures to monitor the software project ([Ardimento et al., 2004](#)). The quality control mechanisms are more effective when a target value is assigned ([Boegh et al., 1999](#); [Ortega et al., 2003](#)). A variant of this approach is to introduce a measurement program based on a customized quality model. The model can be based on a standard, such as ISO/IEC 9126, but the specific quality model should identify relevant quality indicators and measures that make sense to project stakeholders. The customized quality model can additionally support the identification of quality requirements and their respective specification ([Boegh et al., 1999](#)).

The *combined approach* includes elements of both previously discussed approaches. They are based on the introduction of software practices by considering project quality goals. The quality goals drive the specification of quality requirements, the selection of relevant practices to meet the quality goals and the establishment of a measurement process to monitor the effectiveness of a software process ([Balla et al., 2001](#); [Trienekens et al., 2001](#); [van Latum and van Uijtrecht, 2000](#); [van Solingen et al., 1999](#)).

5.1. Research implications

Several proposals are based on mappings between standards or process models. The mappings are important when optimizing organizational resources in SPI initiatives based on multi-models ([Siviy et al., 2008](#)). However, the lack of empirical papers hinders the evaluation of proposals as regards understanding the benefits and weaknesses of each method or model. Furthermore, more research

is needed on the multi-model SPI initiatives that take into account the most common process reference models (CMMI, ISO/IEC 12207, ISO/IEC 15504, and others) and standards focused on specific product quality characteristics.

The dearth of primary papers in this SMS, and particularly the lack of empirical papers dealing with software product quality, has also been reported in others literature reviews. [Barney et al. \(2012\)](#) found that in the abstract section, most papers address software quality aspects generally. In the case of most publications it was not possible to ascertain which specific qualities were studied, and only 8% of the papers selected reported the ISO/IEC 9126. In similar vein, [Unterkalmsteiner et al. \(2012\)](#) found that 15 of 148 papers (10%) dealt with quality characteristics as described in ISO/IEC 9126. These reviews therefore provide strong evidence about the relevance of research on software product quality and the need to research the topic, in our case, within the software process field.

The results of this review show that despite the broad range of quality (sub-) characteristics that can be studied, few of them are considered in SPI initiatives based on process reference models. Security was the main quality characteristic addressed in this SMS using an SPI approach based on mappings. However, there is a need to understand the effect of these mappings on the improvements made to the specific software quality characteristic. In addition, there are other quality characteristics that should be studied in order for software organizations to be able to evaluate the effect of software processes on product quality.

We found a lack of focus on the definitions of software product quality characteristics. This topic is relevant as regards deploying a measurement program within software organizations, because the first step is to rely on an agreement about the meaning of a product quality characteristic and how it can be operationalized using indicators and measures. On the other hand, the measurement program must be adapted to a specific organizational context, in which the process maturity is a main factor. Some proposals, such as those of [Ardimento et al. \(2004\)](#) and [van Latum \(2000\)](#), propose using GQM to measure product quality characteristics.

The ubiquity of software signifies that software organization must have processes that are able to deal with the levels of quality required by the industrial sector and market demands. Quality demands can be considered as a driver for research in this topic ([Allen et al., 2006](#)). Dealing with empirical data in order to assess the impact of a software process on software product quality requires researchers to also report contextual data, such as the type of software, the organizational context and the maturity level ([Petersen and Wohlin, 2009](#)). It is also important to discuss the moderating effect of the type of stakeholders' construct when the impact on product quality is analyzed.

5.2. Implications for practitioners

These results showed that there are a variety of proposals that can be used to consider software product quality within an SPI initiative. If a software company wishes to improve its practices in order to enhance how a quality characteristic is addressed in software development, e.g. security, then various standards are available for implementation. These can follow the steps of an improvement method in order to make changes in the software process. If the company has deployed another process framework, a harmonization approach with which to deal with the implementation of multiple models could support its efforts. This strategy belongs to the capability-based approach ([Section 5](#)).

If a software company has specific business goals with regard to product quality and its customer expects specific quality values and goals, then a measurement-based approach can be used. This approach can provide data with which to monitor the performance of the software products; however, it is necessary to introduce a measurement program into the organization. The measure-

ment program should take into account the quality characteristics, sub-characteristics, indicators, and measures that are relevant to the software product under study. However, the target quality values cannot be achieved by using the measurement program alone.

Benchmarking and analytic approaches can complement each other ([Paulk, 2008](#)). Any SPI managers who wish to deal with product quality characteristics can therefore base their improvement strategy on a combination of both types of SPI approaches. The software process perspective provides a holistic view of the factors that influence software development ([Aaen et al., 2001](#)) which can be used as a framework to establish product quality goals in accordance with business goals. A combined approach, such as that described at the start of [Section 5](#), could therefore be an appropriate strategy with which to deal with software product quality. However, this strategy needs to be detailed in order to facilitate its use in industrial settings.

Although SPI literature has showed that process reference models contribute toward improving product quality ([Agrawal and Chari, 2007](#)), there is not sufficient sound empirical evidence to be able to assert that SPI can result in enhanced software quality when the product view is taken into account. A few reports ([Ashrafi, 2003](#); [Chen and Huang, 2009](#); [Kuiliboer and Ashrafi, 2000](#)) pointed out a moderate effect on some product quality characteristics. However, the evidence is obtained as a result of qualitative empirical methods, based on opinion surveys. In this situation it is therefore difficult to suggest an SPI approach based on solid evidence that can be used to improve product quality.

6. Study limitations

An SMS has different requirements to a systematic literature review. The main differences between an SMS and an SLR are discussed by [Kitchenham et al. \(2011a\)](#) and [Petersen et al. \(2008\)](#). After analyzing the work of the aforementioned authors we found that purpose, research question, selection of primary papers, data extraction procedures and analysis methods are the main factors that may differentiate one method from the other. In addition, SMSs have a limited dissemination because their main target audience is composed of peer researchers. A list of differences between SLRs and SMSs is presented in [Table 8](#). Although these two types of secondary studies are different, the procedures used to select primary papers in SMS should be the same as those used in SLRs ([Wohlin et al., 2013](#)).

This SMS therefore used the selection procedures of the SLR method. We are interested in the trend in the SPI field since we wish to address software product quality. The identification of relevant papers may therefore be auditable but not necessarily complete ([Kitchenham et al., 2011a](#)). Furthermore, [Petersen et al. \(2008\)](#) suggest that a mapping study must include empirical and non-empirical studies.

Two of the main limitations in a review are the bias in selection and data extraction procedures ([Dybå and Dingsøy, 2008](#); [Kitchenham and Charters, 2007](#)). In order to mitigate the impact of selection bias, a protocol was built in the stages prior of this literature review. The protocol clearly identifies the research terms and the inclusion and exclusion criteria. During the execution of the selection of papers, the protocol was refined in order to include only papers addressing product quality characteristics during the development stage, as described in ISO/IEC 25010. This decision excluded papers addressing quality sub-characteristics described in a quality in use model, such as safety. However, during the manual and automatic search, several papers addressing safety using process models were found.

The search string used in this review could also have impacted on the selection of primary papers. Software process improvement is closely related to other concepts such as software process, software

Table 8
Main differences between SLR and SMS.

Aspect	SLR	SMS
Purpose	Identify best practices by comparing technologies	Classification of software engineering studies about a research topic
Research question	Specific. It compares the outcomes of empirical studies	Generic. It could depict research trends.
Search strategy requirements	Rigorous approach to identify all relevant studies	Often less stringent if only research trends are of interest
Quality evaluation	A strong conclusion can only be derived from high quality primary papers	Is not essential because it could include non-empirical papers
Results	Synthesis of outcomes of primary papers to answer specific research question(s)	Classification of papers, considering diverse dimensions and taxonomies

measures, standardization, maturity models, among others (Hansen et al., 2004; Müller et al., 2010; Paulk, 2008). These concepts can influence the perspective with which papers address the topic of SPI and whether they can be included in the literature survey. An SPI is, on the other hand, a full concept that can be investigated independently of other related terms. This argument also relies on a previous literature review that selected literature based on this term (Hansen et al., 2004; Pino et al., 2008). The specific process models considered in the search string were based on other reviews which found that they were those most frequently used in the SPI field (Kelemen et al., 2012; Pardo et al., 2011; Pino et al., 2008; Unterkalmsteiner et al., 2012). The manual search showed, particularly in the papers considered as candidates, that this SMS is a high level review of the field of SPI and software product quality. Narrower SMSs can be conducted to explore each stage of an SPI initiative and each one of the product quality characteristics. In addition, other SMSs could address characteristics of the quality in use model and explore the contribution of defects in order to characterize software product quality.

Keywords used in mapping studies dealing with quality requirements were also considered (Barney et al., 2012; Hansen et al., 2004; Pino et al., 2008; Svensson et al., 2010; Unterkalmsteiner et al., 2012). The number of quality characteristics and quality terms is huge. Standards such as ISO/IEC 9126 or ISO/IEC 25010 propose the most commonly used quality characteristics. Although these standards also identify some sub-characteristics for each characteristic, we only considered the characteristics in the search string since they are those most frequently used. We also included the quality models (ISO/IEC 9126 and ISO/IEC 25010) in the search string. Since we are dealing with a product quality view in a field in which the manufacturing view is the most common, we also selected papers that use generic quality terms related to software products. This decision was made in order to look at the introduction and manually search for a specific product quality characteristic or any approach used to deal with software quality characteristics.

Additional actions were considered to mitigate the impact of selection bias. The second author reviewed 200 records from the collection of retrieved papers. Her results are consistent with the selection of the first author. In addition, the manual search was conducted and found the same empirical papers. The estimate of completeness of the search procedure is around 90%. However, since the automatic and manual search was conducted by the first author there is risk of selection bias. This bias was mitigated by manually labeling each record with the main reason considered to reject it.

Guidelines for systematic reviews also consider other alternatives for the selection of primary papers. In particular, the snowballing procedure is suggested (Kitchenham and Brereton, 2013). The manual search process was used to calculate the sensitivity of the search process. The probable completeness of the search process was around 90%. Since this review is not an SLR it is not therefore necessary to ensure a complete coverage of the papers addressing the topic under study (Kitchenham et al., 2011b).

In order to control the publication bias, the SMS considered four databases usually consulted in the Software Engineering field and recommended for use in this activity (Kitchenham and Brereton, 2013). These databases include papers published in both conference pro-

ceedings and journals. The SMS additionally includes both empirical and non-empirical studies.

In order to address the accuracy of the data extraction procedures, a template was built which was filled in with verbatim data from the primary papers. The data extracted was the basis used to carry out the classification of papers using the Wieringa taxonomy (Wieringa et al., 2006) and the classification of the research approach and the rigor and industrial relevance of empirical papers (Ivarsson and Gorschek, 2011). In order to classify the SPI approaches we reviewed the main activities used in SPI methods and developed a classification schema based on the PDCA cycle (Deming, 1992). Our classification schema considers both the implementation of an SPI project in organizations and the development of methods and models that support each PDCA stage.

There were some differences in the classification of the primary papers. In order to reach an agreement as regards the categories in which each primary study belonged, the results of the first author were compared with the classification carried out by the second and third authors. The main differences were found in the research approach categorization based on the Wieringa taxonomy and the application of the rigor rubric (Ivarsson and Gorschek, 2011). In both cases additional guidelines were built based on the original categories and rubrics. This adaptation considers the current approach to report research results found in the primary papers. In order to address the rigor, an additional template was built considering contextual facets (Petersen and Wohlin, 2009), the main components of a research design, and the main validity threats. Guidelines with which to score the data extracted also were included. After carrying out the refinement of the extraction data procedures, it was easier to reach an agreement as to the classification of the papers. However, the lack of information in empirical papers signified that the authors had to make interpretations about the papers' content, and it is therefore possible that some papers were misclassified.

7. Conclusions and future work

The present mapping study has reviewed the relevant literature in order to understand the approaches used to deal with software product quality in an SPI context. We found 74 primary papers, of which 18 provide empirical data. The papers selected report the majority of the proposals that deal with security-related process frameworks such as SSE-CMM, CC, ISO/IEC 27002. The main quality characteristic addressed is the security aspect in software development. We also found that CMM, CMMI, ISO/IEC 12207 and ISO/IEC 15504 have been used in SPI proposals. Other quality characteristics addressed are usability, reliability and maintainability.

The main approaches used to improve software product quality characteristics are capability-based and measurement-based. In the first case, the research community is developing new process models or assessment models based on the combination of different regulations, standards or reference models. These models basically address one specific quality characteristic, such as security. However, few proposals deal with the implementation aspects of these initiatives.

With regard to the measurement-based approach, we found that some papers provide high-level guidelines with which to combine

diverse quality technologies, including measurement methods, to support product quality characteristics within a software process. The causal relationships between process and product quality, however, are very difficult to establish and depend on software project settings.

The product quality view is barely addressed in SPI literature and there is a lack of empirical studies that validate the current proposals. The research implication Section allows us to suggest research areas such as understanding the effect of traditional software process models on software product quality, studying the impact of mappings between reference models on the implementation of software process, or developing methods with which to introduce specific practices oriented toward enhancing a product quality characteristic in organizational software processes. In addition, it is important to develop methods that can identify when a product quality characteristic should be considered in particular industrial settings or domain applications areas.

There are other issues regarding software product quality that should be addressed by SPI initiatives, but we did not find any papers that consider them. For instance, software organizations are beginning to realize that product quality characteristics should be managed (Berntsson Svensson et al., 2012). In order to provide software organizations with support, we are currently developing a process framework focused on the management of interactions between product quality characteristics (García-Mireles et al., 2013). Our proposal deals with the interactions, particularly the negative ones, which could arise between quality characteristics. The quality characteristics are taken from ISO/IEC 25010 and correspond to usability, maintainability and security. We have also considered the respective sub-characteristics. The identification of conflicting interactions between product quality characteristics and the mechanisms used to resolve them requires process support. The processes additionally provide a set of specific practices to be introduced in software process in order to improve a particular product quality characteristic.

Acknowledgments

This work has been funded by the projects: GEODAS-BC (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01), VILMA (PEII 2014-048-P, Junta de Comunidades de Castilla-La Mancha and FEDER) and INGENIOSO (PEII-2014-050-P, Junta de Comunidades de Castilla-La Mancha and FEDER).

References

- Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O., 2001. A conceptual MAP of software process improvement. *Scand. J. Inform. Syst.* 13 (1), 123–146.
- Agrawal, M., Chari, K., 2007. Software effort, quality, and cycle time: a study of CMM level 5 projects. *IEEE Trans. Softw. Eng.* 33 (3), 145–156.
- Al-Qutaish, R.E., 2009. Measuring the software product quality during the software development life-cycle: an international organization for standardization standards perspective. *J. Comput. Sci.* 5 (5), 392–397.
- Allen, J., Kitchenham, B., Konrad, M., 2006. Theme Q. The relationships between processes and product qualities. In: Forrester, E. (Ed.), *A Process Research Framework*. Software Engineering Institute, Carnegie Mellon, pp. 19–28.
- Ardimento, P., Baldassarre, M.T., Caivano, D., Visaggio, G., 2004. Multiview framework for goal oriented measurement plan design. In: Bomarius, F., Iida, H. (Eds.), 5th International Conference, PROFES 2004, LNCS 3009. Springer, Berlin, Heidelberg, Kansai Science City, Japan, pp. 159–173.
- Ashrafi, N., 2003. The impact of software process improvement on quality: in theory and practice. *Inform. Manage.* 40 (7), 677–690.
- Asthana, A., Olivieri, J., 2009. Quantifying software reliability and readiness, IEEE International Workshop Technical Committee on Communications Quality and Reliability, 2009. CQR 2009. Naples, FL, pp. 1–6.
- Balla, K., Bemelmans, T., Kusters, R., Trienekens, J., 2001. Quality through managed improvement and measurement (QMIM): towards a phased development and implementation of a quality management system for a software company. *Softw. Qual. J.* 9 (3), 177–193.
- Barney, S., Petersen, K., Svahnberg, M., Aurum, A., Barney, H., 2012. Software quality trade-offs: a systematic map. *Inform. Softw. Technol.* 54 (7), 651–662.
- Berander, P., Damm, L.O., Eriksson, J., Gorschek, T., Henningson, K., Jonsson, P., Kagstrom, S., Milicic, D., Martensson, F., Ronkko, K., Tomaszewski, P., June 2005. *Software Quality Attributes and Trade-offs*. Blekinge Institute of Technology.
- Berntsson Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahroki, A., Feldt, R., 2012. Quality requirements in industrial practice—an extended interview study at eleven companies. *IEEE Trans. Softw. Eng.* 38 (4), 923–935.
- Boegh, J., Depanfilis, S., Kitchenham, B., Pasquini, A., 1999. Method for software quality planning, control, and evaluation. *IEEE Softw.* 16 (2), 69–77.
- Breu, R., Kuntzmann-Combelles, A., Felderer, M., 2014. New perspectives on software quality [Guest editors' introduction]. *IEEE Softw.* 31 (1), 32–38.
- Card, D.N., 1998. Learning from our mistakes with defect causal analysis. *IEEE Softw.* 15 (1), 56–63.
- Coté, M.-A., Suryn, W., Georgiadou, E., 2007. In search for a widely applicable and accepted software quality model for software quality engineering. *Softw. Qual. Control* 15 (4), 401–416.
- Chandra, E., Christopher, D.F.X., Vijaykumar, S.D., 2011. Study of CMMI based process framework for quality models. 2011 3rd International Conference on Trend in Information Sciences and Computing (TISC), pp. 173–176.
- Chen, J.C., Huang, S.J., 2009. An empirical analysis of the impact of software development problem factors on software maintainability. *J. Syst. Softw.* 82 (6), 981–992.
- Cheng, X., Li, M., Zhao, H., 2007. Information security evaluation of e-government systems. In: Guo, Q.P. (Ed.), *DCABES 2007 Proceedings*, Vols. I and II. China, pp. 532–535.
- Chiam, Y.K., Staples, M., Ye, X., Zhu, L., 2013. Applying a selection method to choose Quality Attribute Techniques. *Inform. Softw. Technol.* doi:10.1016/j.infsof.2013.02.001.
- de Almeida Biolchini, J.C., Mian, P.G., Natali, A.C.C., Conte, T.U., Travassos, G.H., 2007. Scientific research ontology to support systematic review in software engineering. *Adv. Eng. Inform.* 21 (2), 133–151.
- Deming, W.E., 1992. *Out of the Crisis: Quality, Productivity and Competitive Position*. Cambridge University Press, Cambridge.
- Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: a systematic review. *Inform. Softw. Technol.* 50 (9–10), 833–859.
- Fonseca, J.A., de Almeida, J.R., 2005. CMMI RAMS extension on based on CENELEC railway standard. In: Winther, R., Gran, B.A., Dahll, G. (Eds.), *Computer Safety, Reliability, and Security, Proceedings, LNCS 3688*. Springer, Berlin, Heidelberg, pp. 1–12.
- García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M., 2012. Towards the harmonization of process and product oriented software quality approaches. In: Winkler, D., O'Connor, R., Messnarz, R. (Eds.), *EuroSPI2012, CCIS 301*. Springer, Berlin, Heidelberg, pp. 133–144.
- García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M., 2013. A framework to support quality trade-offs from a process-based perspective. In: McCaffery, F., O'Connor, R.V., Messnarz, R. (Eds.), *EuroSPI2013, CCIS 364*. Springer-Verlag, Berlin, Heidelberg, pp. 96–107.
- Halvorsen, C.P., Conradi, R., 2001. A taxonomy to compare SPI frameworks. *Proceedings of the 8th European Workshop on Software Process Technology*. Springer-Verlag, pp. 217–235.
- Hamann, D., Jarvinen, J., Birk, A., Pfahl, D., 1998. A product-process dependency definition method. 24th Euromicro Conference – Proceeding, Vols. 1 and 2. IEEE, pp. 898–904.
- Hansen, B., Rose, J., Tjørnehøj, G., 2004. Prescription, description, reflection: the shape of the software process improvement field. *Int. J. Inform. Manage.* 24 (6), 457–472.
- Harter, D.E., Kemerer, C.F., Slaughter, S.A., 2012. Does software process improvement reduce the severity of defects? A longitudinal field study. *IEEE Trans. Softw. Eng.* 38 (4), 810–827.
- Harter, D.E., Krishnan, M.S., Slaughter, S.A., 2000. Effects of process maturity on quality, cycle time, and effort in software product development. *Manage. Sci.* 46 (4), 451–466.
- Iivari, J., 1996. Why are CASE tools not used?. *Commun. ACM* 39 (10), 94–103.
- ISO, 2001. ISO/IEC 9126 Software engineering – product quality, in Part 1 – Quality model.
- ISO, 2010. ISO/IEC FCD 25010: Systems and software engineering – system and software product quality requirements and evaluation (SQuARE) – System and software quality models.
- Ivarsson, M., Gorschek, T., 2011. A method for evaluating rigor and industrial relevance of technology evaluations. *Empir. Softw. Eng.* 16 (3), 365–395.
- Kelemen, Z.D., Kusters, R., Trienekens, J., 2012. Identifying criteria for multimodel software process improvement solutions—based on a review of current problems and initiatives. *J. Softw.: Evol. Process* 24 (8), 895–909.
- Kitchenham, B., Brereton, P., 2013. A systematic review of systematic review process research in software engineering. *Inform. Softw. Technol.* 55 (12), 2049–2075.
- Kitchenham, B., Budgen, D., Brereton, P., 2011. Using mapping studies as the basis for further research – A participant–observer case study. *Inform. Softw. Technol.* 53 (6), 638–651.
- Kitchenham, B., Charters, S., 2007. *Guidelines for Performing Systematic Literature Review in Software Engineering*, ver. 2.3. Keele University, EBSE Technical Report, EBSE-2007-01.
- Kitchenham, B., Li, Z., Burn, A., 2011. Validating search processes in systematic literature reviews. *Proceeding of the 1st International Workshop on Evidential Assessment of Software Technologies, EAST 2011, in Conjunction with ENASE 2011*, pp. 3–9.

- Kitchenham, B., Pfleeger, S., 1996. Software quality: the elusive target. *IEEE Softw.* 13 (1), 12–21.
- Kuilboer, J.P., Ashrafi, N., 2000. Software process and product improvement: an empirical assessment. *Inform. Softw. Technol.* 42 (1), 27–34.
- Lavallée, M., Robillard, P.N., 2011. Do software process improvements lead to ISO 9126 architectural quality factor improvement. 8th International Workshop on Software Quality, pp. 11–17.
- Müller, S.D., Mathiassen, L., Balshøj, H.H., 2010. Software process improvement as organizational change: a metaphorical analysis of the literature. *J. Syst. Softw.* 83 (11), 2128–2146.
- Nunes, F.J.B., Belchior, A.D., Albuquerque, A.B., 2010. Security engineering approach to support software security. In: 2010 6th World Congress on Services (SERVICES-1), pp. 48–55.
- O'Connor, R.V., 2009. Exploring the role of usability in the software process: a study of Irish software SMEs. In: EuroSPI2009, pp. 161–172.
- Ortega, M., Pérez, M., Rojas, T., 2003. Construction of a systemic quality model for evaluating a software product. *Softw. Qual. J.* 11 (3), 219–242.
- Pardo, C., Pino, F.J., García, F., Velthuis, M.P., Baldassarre, M.T., 2011. Trends in harmonization of multiple reference models. *Commun. Comput. Inform. Sci.*, 61–73.
- Paulk, M.C., 2008. A taxonomy for improvement frameworks. In: Fourth World Congress for Software Quality, September 2008. Bethesda, Maryland.
- Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. British Computer Society, Italy, pp. 68–77.
- Petersen, K., Wohlin, C., 2009. Context in industrial software engineering research. In: ESEM 2009. 3rd International Symposium on Empirical Software Engineering and Measurement, 2009, pp. 401–404.
- Pino, F.J., García, F., Piattini, M., 2008. Software process improvement in small and medium software enterprises: a systematic review. *Softw. Qual. J.* 16 (2), 237–261.
- Satpathy, M., Harrison, R., Snook, C., Butler, M.J., 2000. A Generic Model for Assessing Process Quality, 10th International Workshop on New Approaches in Software Measurement, pp. 94–110.
- Shubhamangala, B.R., Suma, V., Amarendra Reddy, P., Goodluck Singh, C., 2013. Quality attribute focused multilayer requirement elicitation: judicious approach to drive business value. In: Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, pp. 2069–2075.
- Siviy, J., Kirwan, P., Morley, J., Marino, L., 2008. Maximizing your Process Improvement ROI through Harmonization. SEI – Carnegie Mellon University, Access: Nov. 2012 from http://www.sei.cmu.edu/library/assets/whitepapers/multimodelExecutive_wp_harmonizationROI_032008_v1.pdf.
- Staples, M., Niazi, M., 2008. Systematic review of organizational motivations for adopting CMM-based SPI. *Inform. Softw. Technol.* 50 (7–8), 605–620.
- Svensson, R.B., Höst, M., Regnell, B., 2010. Managing quality requirements: a systematic review. In: Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), IEEE, pp. 261–268.
- Trienekens, J., Kusters, R., Van Solingen, R., 2001. Product focused software process improvement: concepts and experiences from industry. *Softw. Qual. J.* 9 (4), 269–281.
- Tzeng, J.R., Chen, C.H., 2012. Applying of ISO9126 and ISO12207 integrated Fuzzy QFD to improve system maintenance quality. *ICIC Express Lett.* 6 (2), 321–327.
- Unterkalmsteiner, M., Gorschek, T., Islam, A.K.M.M., Cheng, C.K., Permedi, R.B., Feldt, R., 2012. Evaluation and measurement of software process improvement – A systematic literature review. *IEEE Trans. Softw. Eng.* 38 (2), 398–424.
- van Latum, F., van Uijtrecht, A., 2000. Product driven process improvement PROFES experiences at Drager. In: Bomarius, F., Oivo, M. (Eds.), Product Focused Software Process Improvement. LNCS 1840. Springer, Berlin, Heidelberg, pp. 232–243.
- van Solingen, R., Derks, P., Hirvensalo, J., 1999. Product focused SPI in the embedded systems industry – experiences of Drager, Ericsson and Tokheim. In: Oivo, M., Kuvaja, P. (Eds.). International Conference on Product Focused Software Process Improvement, pp. 86–99.
- von Wangenheim, C.G., von Wangenheim, A., Hauck, J.C.R., McCaffery, F., Buglione, L., 2012. Tailoring software process capability/maturity models for telemedicine systems. 18th Americas Conference on Information Systems 2012, pp. 2472–2480.
- Wagner, R., Fontoura, L.M., Fontoura, A.B., 2011. Using security patterns to tailor software process. *SEKE 2011*, pp. 672–677.
- Wagner, S., Lochmann, K., Winter, S., Goeb, A., Klaes, M., Nunnenmacher, S., 2012. Software Quality Models in Practice: Survey Results. Technische Universität München. TUM-I129, Access: Oct. 2013 from <http://mediatum.ub.tum.de/doc/1110601/1110601.pdf>.
- Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Require. Eng.* 11 (1), 102–107.
- Winter, J., Rönkkö, K., 2010. SPI success factors within product usability evaluation. *J. Syst. Softw.* 83 (11), 2059–2072.
- Wohlin, C., Runeson, P., Da Mota Silveira Neto, P.A., Engström, E., Do Carmo Machado, I., De Almeida, E.S., 2013. On the reliability of mapping studies in software engineering. *J. Syst. Softw.* 86 (10), 2594–2610.
- Zhang, H., Babar, M.A., Tell, P., 2011. Identifying relevant studies in software engineering. *Inform. Softw. Technol.* 53 (6), 625–637.
- Zhou, J., Stålhane, T., 2004. Web-based system development: status in the Norwegian IT organizations. In: 5th International Conference, PROFES 2004, pp. 363–377.
- Zia, T.A., 2010. An analytical study of IT security governance and its adoption on Australian organisations. In: 8th Australian Information Security Management Conference, pp. 182–194.
- Zuccato, A., 2007. Holistic security management framework applied in electronic commerce. *Comput. Security* 26 (3), 256–265.

Gabriel Alberto García-Mireles holds the MSc degree in Computer Science from the CICESE (2001) and he is professor at the University of Sonora. Actually he is a PhD student in Computer Science at the University of Castilla-La Mancha. His research interests are software process improvement and software quality.

M^a Ángeles Moraga is associate professor of Computer Science at the University of Castilla-La Mancha. PhD in Computer Science by the same University, her research interests are software quality and software measurement, especially in web portals and data.

Félix García received his MSc (2001) and PhD (2004) degrees in Computer Science from the University of Castilla-La Mancha (UCLM). He is currently an associate professor in the Department of Information Technologies and Systems at the UCLM. He is member of the Alarcos Research Group and his research interests include business process management, software processes, software measurement, and agile methods. More information about publications in these topics can be found in <http://alarcos.esi.uclm.es/per/fgarcia/>.

Mario Piattini is a full professor at the UCLM. He holds the PhD degree in Computer Science from the Technical University of Madrid and leads the Alarcos Research Group. He is CISA, CISM, CGEIT, and CRISC by ISACA. His research interests include software quality, metrics and maintenance. He is the Director of the Joint UCLM-Indra Software Research and Development Center and the Institute of Information Systems and Technologies.