Methodologies for Evaluation and Improvement of Software Processes in the Context of Quality and Maturity Models: a Systematic Mapping

Daniel Fernando G. Feloni and Rosana T. Vaccare Braga

University of Sao Paulo
Institute of Mathematics and Computer Sciences
Sao Carlos, Sao Paulo, P.O. Box 668 – 13.566-590
dfeloni@icmc.usp.br rtvb@icmc.usp.br

Abstract. A widely recognized fact is that the quality of software products is largely determined by the quality of the process used to develop them. Regardless of the number of tools/frameworks developed to aid companies to perform software process assessment (SPA) and improvement (SPI) activities, the industry often suffers with quality issues in their products. In order to evaluate the SPA and SPI research areas, a systematic mapping was performed to identify the available tools/frameworks and methodologies used in research, as well as the existing gaps in the area. As a result it was possible to identify the current trends in the SPA and SPI research areas that can be used as a guideline to future work and can contribute to the software engineering and quality community.

1 Introduction

Quality has been a well known issue inside the software industry for a long time. Thus, the application of software engineering concepts during software development represents an attempt to improve the quality of software produced. However, the definition, implementation and monitoring of a software process require resources (often limited) and therefore more costs for organizations. Not always the organization has a team mature enough to perform monitoring activities in a self-manageable way, which leads to the need to implement a software development process in the organization. To implement a good process, organizations can make use of a software quality model as a guide to make their software processes more mature, in accordance with specific and generic practices, which are structured in process areas defined by the quality model.

Among the causes of failure to implement best practice processes we can mention the low level of knowledge about quality models by software development companies. Another major problem is the cost involved in the implementation of quality standards or models, which can become an impediment factor, especially in the context of small and medium-sized companies with limited budgets [1].

In recent years, there has been a demand for better services and features in software products. Even with the development of various methods, techniques and tools, software products still suffer from excessive costs, delivery delays, and poor quality [18], [9] *apud* [2].

With the growth of demand for products and services, most software development companies continue to face problems in their projects due to lack of implementation of best practice processes and standards. A study performed in Malasya identified that 50% of all applications fail to meet their business objectives, while 40% of IT projects fail to achieve the financial returns expected, due to the lack of use of good software process practices [1]. Thus, we can infer that one of the major difficulties for many companies in relation to quality assurance is the lack of experience of managing their software processes.

History has shown that, when using a well defined process during software development, conventional models have brought a certain amount of useful structure for software engineering, besides providing a quite effective roadmap for software development teams. In fact, the quality of the software product is largely determined by the quality of the process used to develop and keep it [12]. What is missing for the industry is to consider that each organization has different factors that must be considered when designing a software process that will be used by their teams during the software development lyfe cicle, such as: (i) focus on functionality or documentation; (ii) make the development in small iterations or complete products; (iii) involvement of stakeholders throughout the process or only at the beginning; among other factors. Each variation of the combination of these factors may result in a different process to better serve that particular development scenario.

To attend these needs, quality and software maturity models began to emerge, such as the ISO/IEC 12207 [7], ISO/IEC 15504 [5], ISO/IEC 9126 [6], ITIL [8], COBIT [8], CMMI-DEV [23], and MPS.Br [24]. Quality models can be described as meta-processes, and they emerged to assess the quality of software processes when applied in an organization [8]. However, a major problem for the adoption of these quality models is the high cost of implementation, and according to some researches, that factor is independent of company size [14] *apud* [2] and may involve costs with internal human resources training or by external consultants in software quality. Apparently, the key constraints for small and medium companies, including the management of software processes and activities are as follows: principles of good practices in software process are not used to carry out tasks including developing plans and implementation; the execution of tasks is *ad hoc*, relying on internal human resources experience with no background on software quality; and there are few resources with appropriate skills to perform their activities [2].

In this context, Software Process Improvement (SPI) and Software Process Assessment (SPA) have emerged to increase quality and productivity in software projects. An increasing number of international initiatives related to SPI has been created and updated, such as the Capability Maturity Model Integration - Development (CMMI-DEV), ISO/IEC 15504, ISO/IEC 12207, among others [2].

The SPI and SPA activities are a key factor for the implementation of quality in process models by the software industry. The main objective is to achieve a better level of quality in their processes and consequently in their products, adding market value to software organizations. However, making an SPI implementation without an approach to support the corresponding SPA program can cause gaps in the documentation and, therefore, the knowledge management of the organization's products. This may result in the consumption of human and financial resources far beyond those available in the organizations budget [2]. Thus, it is possible to observe the existence of a number of initiatives to create approaches to the realization of SPA and SPI [4], [15] for the harmonization of quality models [16], [8] and the creation of repositories to support the adoption of software quality models [20], [25], featuring the growing need for innovation in the SPI and SPA research areas.

This article presents a systematic mapping (SM) conducted in order to further investigate the state of the art about SPA and SPI approaches. The remainder of this paper is organized as follows. Section 2 lists the aspects related to the planning of systematic mapping, i.e., the definition of the research objectives, criteria for inclusion and exclusion of studies for analysis, selection of search sources and the research protocol. In Section 3 it is presented a synthesis of the state of contemporary art regarding the area of SPA and SPI and also the research gaps identified. Finally, a conclusion with the impressions taken from the analyzed studies is presented in Section 4.

2 Methodologies for Evaluation/Improvement of Software Process in the Context of Quality/Maturity Models

As discussed in the previous section, software quality assurance can be achieved with the use of a well defined software development process institutionalized in the organization and, with further efforts, a certification in software quality model can be obtained. However, the implementation of SPI activities consists of a process that requires effort, experience, methods and tools to support the organization during a certification process.

The SPI existing tools usually address some of the following activities: (i) collection and management of general information about the company, evaluation of goals and projects, immediate information processing and evaluation of management items; (ii) selection of processes to be reviewed; generation of static questionnaires adressing specific aspects of quality models; (iii) self-assessment of tasks assigned by software engineering professionals (requiring the assistance of experts) [2]. Therefore, we identified the need to evaluate the approaches and methodologies available in the literature, defining which quality models are most adressed and how that information is stored for future reuse.

The objective of a systematic mapping is to reveal the state of the art in a particular subject, observing the number of selected studies, their types, the available results, the frequency of publications over time, among other factors [21]. The systematic mapping process can be described as a set of well defined and planned tasks according to a previously established protocol [11], composed of the following phases: planning, conduction and analyzis of the results, as shown in the following sections.

2.1 Planning the Systematic Mapping

The systematic mapping conducted aims to identify approaches that provide support to SPI activities. The term "approach" is used to exemplify any tool, framework, model, spreadsheet, among other ways we have found in the literature to perform SPA and SPI programs. The intention is to check which quality models are most often cited in the literature and which techniques/methodologies are most used by these models. Based on these objectives, it was possible to elaborate the following **research questions**:

1. What are the types of approaches that support the evaluation and improvement of software processes (SPA and SPI)?
2. Do such approaches allow the excecution of the evaluation considering one or more quality models?
3. Do the approaches enable the reuse of processes and quality models?
4. What criteria are used by these approaches to suggest process improvement activities?
5. How do these approaches make the mapping between the elements of the executed software process and the attributes required by quality models?

According to these questions, we can extract the specificities of this mapping, i.e, the identification of the **PICO (Population, Intervention, Comparison and Outcomes)**, namely:

- Population: approaches to implement SPA and SPI using software quality models;
- Intervention (which will be observed): techniques, approaches and criteria used for evaluation of processes and their adherence to quality models;
- Comparison: identify different approaches developed for evaluation of software processes and which criteria/indicators are used for the indication of SPI activities to facilitate the process of obtaining quality models certification;
- Results: approaches to implement techniques and criteria that can be used to evaluate and improve processes and to identify research gaps in the area of SPA and SPI.

Given the context of the research, some inclusion criteria were created to verify the relevance of the studies selected in the scope of the systematic mapping.

1. **IC-1**: Papers that describe the development and implementation of approaches used to define activities to improve processes;
2. **IC-2**: Papers that present methods/criteria to map processes between areas of different quality models;

3. **IC-3**: Papers presenting the main problems in the implementation of quality models;
4. **IC-4**: Papers defining abstraction models for software quality models and processes;
5. **IC-5**: Papers that present approaches to store and retrieve software engineering artifacts in the context of reuse in quality models.

To complete and refine the search and analysis of the retrieved articles, the following exclusion criteria were defined:

1. **EC-1**: Papers not related to SPA and SPI;
2. **EC-2**: Papers which do not support approaches to assessing and improving software processes;
3. **EC-3**: Papers that are in languages other than English and Portuguese;
4. **EC-4**: Papers that are not available to download in the Web.

The selected search engine, *Scopus*, is among the leading in computer area and has great impact in the research area. It covers conferences, magazines, books, and its contents are updated in a regular basis, and most articles are available for download. This choice was due to the number of studies related to the subject and the fact that *Scopus* indexes the top research sources such as *IEEE Xplore*, *ACM Digital Lybrary* and *Science Direct*.

2.2 Construction of the Search String

Synonyms were defined to compose the search string, acccording to a few papers used as basis for the search (these are named Control Group and were indicated by experts). The keywords that represent the core of the research are: *software process improvement* and *software process evaluation* (or *software process assessment*).

These terms were divided into five parts to form the search expression. As shown in Table 1, the **A** term refers to software engineering context; the **B** term refers to software process improvement (SPI) and assessment (SPA); the term **C** refers to the method used to define tools/approaches for execute (SPI) and (SPA); the term **D** is the context of maturity models and standards for the evaluation of software processes and the term **E** represent the words that indicate the purpose of creation of the techniques presented by the articles.

2.3 Conduction of the Systematic Mapping

The conduction of this systematic mapping occurred through four analysis phases before the full reading of the papers provided by the search engine. In the first phase, it was performed the identification of the papers in the search engine, in which 315 papers were identified, from these, 88 were removed because they were duplicated. In the second phase, 227 papers were analyzed by reading their abstracts and titles resulting in 80 papers selected for the next stage. The third

Table 1. Terms used in the search string construction.

Term	Synonym
A	“software”, “software engineering”
B	“process improvement”, “software process improvement”, “spi”, “self-assessment”, “assessment”, “process assessment”, “software process assessment”
C	“framework”, “tool”, “tool based”, “tool-based”, “tool oriented”, “tools”, “guideline”, “approach”, “method”, “criterion”, “techniques”, “methodologies”, “tool support”, “meta model”, “meta-model”, “meta-model”, “abstraction”, “abstract model”, “abstract models”
D	“maturity”, “maturity model”, “maturity models”, “capability model”, “capability models”, “maturity capability model”, “maturity capability models”, “software process standard”, “scampi”, “cmmi”, “spice”, “iso/iec 15504”, “iso/iec 20000”, “iso/iec 12207”, “cmmi-dev”
E	“aid”, “help”, “support”, “assist”
String	**(A AND B AND C AND D AND E)**

phase consisted in reading the introduction and conclusion of the 80 papers selected in the previous phase, which resulted in the selection of 17 papers to be fully read. Finally, an additional phase was conducted to point out the papers that represents the main aspects identified in all the samples analysed, in which resulted in 6 articles selected among those selected in phase 3. Table 2 summarizes the sequence of steps for the studies selection.

Table 2. Stages in the selection of the work process.

Phase	Performed Analysis	Analysed	Remained
1	Identification of papers in the search engine	**315**	**227**
2	Exclusion of papers based on the title, *abstract* and duplicated studies	**227**	**80**
3	Exclusion of the papers based on the introduction and conclusion	**80**	**17**
4	Selection of papers that represents the main aspects in the sample analysed	**17**	**6**

After the last phase of the papers analysis it was possible to retrieve some statistics regarding the acceptance and exclusion criterias defined for this systematic review. Table 3 shows the distribution of the acceptance and exclusion criteria in the papers analyzed among the studies returned from the search engine. The field “# of Papers” represents the number of papers analysed that fit the given criteria. Some papers were recorded in more than one criteria. Table 3 provides a view of the distribution among the defined criteria for this systematic mapping and highlights which aspects were most present in the literature.

Table 3. Inclusion and exclusion criteria distribution.

Criteria	Type	# of Papers
IC-1	Inclusion	69
IC-2	Inclusion	41
IC-3	Inclusion	14
IC-4	Inclusion	11
IC-5	Inclusion	6
EC-1	Exclusion	184
EC-2	Exclusion	181
EC-3	Exclusion	1
EC-4	Exclusion	9

Most of the papers refers to the development and use of tools and frameworks to perform SPA and SPI. Some studies suggests the use of approaches such as the creation of ontologies and a semantic analysis of project artifacts. The objective is to evaluate the current state of an executed software processes, which will be used to determine whether (and how much) they comply with the practices and objectives defined in a particular quality model. Amongst the selected papers, we have chosen 6 (selected in phase 4 of our analysis) that represent a synthesis of the approaches proposed by the academy in the effort to support SPA and SPI initiatives (see Table 4). It was also possible to present a categorization that divides the efforts found towards SPI and SPA activities in the literature, as listed below:

1. Tool/Framework: these papers present different approaches to the implementation of the assessment process, the main goal of the proposed methods consists in obtaining a gap report with measures to be taken to meet a certain maturity level in a specific process area [4], [15], [13];
2. Harmonization of reference models: in some studies [16], [8], [19] it is identified the need for some organizations to adopt different quality models to meet specific business needs. In this sense, some studies propose approaches to harmonize these models, in other words, meet similar practices and objectives through different quality models and/or standards. The goal is to identify process areas, roles, activities, artifacts, and other software process components that are similar in different quality models, trying to optimize the process of definition of SPI activities in these organizations. Among the approaches used, we can mention the use of MDE as the main tool to define conceptual models that can represent the quality models without being attached to any of them;

3. Repositories: some studies show initiatives to create repositories for the persistence of software engineering artifacts (or artifact metadata) that could be reused in the context of SPI and SPA [20], [25].

Table 4. Selected articles in phase 4.

Category	Reference	Title
Tool	[4]	SPIALS-II: A light-weight software process improvement self-assessment tool
Tool	[15]	Taba workstation: Supporting software process deployment based on CMMI and MR-MPS.BR
Harmonization	[16]	MATURE: A model driven based tool to automatically generate a language that supports CMMI process areas specification
Harmonization	[8]	Efficient adoption and assessment of multiple process improvement reference models
Repository	[20]	Software repository for software process improvement
Repository	[25]	Building a Maturity & Capability Model repository

3 Comparative Analysis

Most of the studies propose methodologies and approaches (with support of computational tools or abstraction models) to carry out the evaluation of software processes, as well as the suggestion of improvement actions to these processes activities.

The proposals range from tools based on questionnaires to MDE techniques that seek to represent quality models and identify simmilarities to facilitate the adoption of multiple quality models, according to business needs. However, the studies seem to be disconnected, using distinct approaches to provide the same results with the same limitations as discussed below. It is noteworthy that none of the analyzed studies could produce computational tools composed of these three concepts: representation (matching models), evaluation and persistence of these models in a single environment for further reuse.

Table 5 presents the main studies identified in phase 3 of the systematic mapping. It can be noticed the lack of studies that cover the whole process of modeling, evaluation and persistence of software quality models. The only work that can cover such characteristics [15] has a strong dependency on the maturity model defined during its development. It also does not make use of MDE techniques to enable the harmonization of quality models like it is suggested in the literature analysed. These aspects could make it more flexible to use with different quality models allowing users not to restrict their processes on one quality model in particular.

It could be noticed that the main models used in the elaboration of the studies are CMMI, ISO/IEC 15504 and ISO/IEC 12207. MPS.Br [24] was used in the Bazilian scope only, due to its nacional development. Another point observed in this systematic mapping was the use of MDE approaches (on all items analyzed in full) to try to harmonize different software quality models, which indicates the potential of this methodology to build a way to model different quality models using a single metamodel structure.

Through these observations, it can be seen that the studies related to SPI and SPA have some gaps, as listed below:

1. Quality model dependency
   – None of the studies presented a methodology that was independent of a particular quality model. In most cases, only a portion of the model was covered by the established approach (tool/spreadsheets/frameworks), indicating the creation of extensions to cover more maturity levels of quality models in future work.
2. Reuse
   – Software reuse, in the context of SPA and SPI, aims to create artifacts that can be used by different organizations and that have similar applicability contexts, such as document templates, documentation standards and project management documents. A concept that is still little explored in the context of SPA and SPI is the reuse of processes and quality models. A major challenge is creating a form of representation of existing quality models (MDE metamodels) so that they can be reused and instantiated by different tools, whether through web services or knowledge base repositories shared by the software engineering community.
3. Involvement of the Software Engineering community
   – Most of the studies analyzed aimed to create approaches to support SPA and SPI based on small industry groups, and they were generally targeted to a specific need. For example, the main target audience of the studies were small and medium-sized enterprises, however, there are studies indicating that the difficulties and challenges presented are too similar regardless of the company size. Another important detail is that the tools and methodologies found in the literature makes use of the lowest levels of maturity models, leaving the continuous evolution of processes entirely to the responsibility of organizations. The involvement of the software engineering community must be more significant, collecting constant feedback on the effectiveness of approaches provided by the academy. The effectiveness of the approaches created cannot be fully proven, because in many cases the tools provided should be used with the support of specialists. Therefore, the costs that were supposed to be decreased will remain the same.
4. Use of recent concepts of Software Engineering
   – The concept of matching quality models seems very promising, in order to create a unified way to represent them and identify similarities to facilitate its adoption. If successfully implemented, this concept could be

used to leverage the reuse of processes, since these would be modeled in a homogeneous structure and could be made available to the community. Not only MDE can be better used in SPA and SPI representation of quality models, but also the provision of services that are used by the industry to evaluate their processes.

5. Reference models in software process quality
   – ISO/IEC 15504 and ISO/IEC 12207 have been widely used as a technical background for the development of new quality models such as CMMI-DEV and MPS.Br. However, for each new quality model created, new terms, practices and objectives arise, which can be described differently but with the same purpose in all of them. This fact hinders the use of approaches like ontologies and semantic analysis in the context of SPA and SPI, because the terms used to describe similar concepts are often identified differently in quality models. Therefore, the existence of a standard reference model for software quality processes would be desirable, in order to unify terms, concepts, practices, and objectives that should be employed by existing quality models. Such a tool would facilitate the harmonization efforts of these models, as well as the creation of approaches to assist in the adoption of multiple models of process quality.
6. Integration methodologies
   – A constant deficiency in SPA and SPI approaches is the lack of integration methodologies to facilitate the implementation of these initiatives. In this sense, integrated environments should be created, which could allow the execution of SPA and SPI, as well as sharing the knowledge about adoption of quality models. This could be achieved with more involvement of the software/quality engineering community and availability of repositories with assets to be reused in certification processes. All these elements combined could lead to an integrated tool that would enable the community to share experiences in different contexts of software engineering. The existence of a knowledge base that concentrates processes, quality models and artifacts to support processes would be a considerable advance in the area of SPI and SPA, because it could be used as a reference guide to the concepts related to engineering and software quality.

In the studies selected in the context of harmonized quality models, it was possible to identify two approaches: (i) the use of metamodels to define a single structure which represents only a reference guide [16] while the other, (ii) [8] is an extension of the previous approach, because in addition to defining a metamodel, it also provides a framework to identify similarities between different maturity models, facilitating the adoption of multiple models.

Table 5. Summarization of the identified studies and its main features.

Title	Repository	Process	Project	Maturity
SPIALS-II: A light-weight software process improvement self-assessment tool [4]	No	No	Yes	Yes
Semantic based approach supporting CMMI gap analysis process [22]	No	No	No	Yes
Managing the software process with a software process improvement tool in a small enterprise [2]	No	No	No	Yes
A software tool to support the integrated management of software projects in mature SMEs [13]	No	Yes	Yes	Yes
MATURE: A model driven based tool to automatically generate a language that supports CMMI process areas specification [16]	No	No	No	Yes
A design of tool for software processes assessment and improvement [26]	No	No	Yes	Yes
Software repository for software process improvement/Frameworks of integration repository for software process improvement using SOA [20]	Yes	No	No	No
An application tool to support the implementation of integrated software process improvement for Malaysia's SME [1]	No	Yes	No	Yes
Taba workstation: Supporting software process deployment based on CMMI and MR-MPS.BR [15]	Yes	Yes	Yes	Yes
A framework for assisting the design of effective software process improvement implementation strategies [17]	No	No	No	Yes
An ontology based infrastructure to support CMMI-based software process assessment [3]	No	No	No	Yes
Efficient adoption and assessment of multiple process improvement reference models [8]	No	No	No	Yes
A self-assessment framework for finding improvement objectives with ISO 29119 test standard [10]	No	Yes	No	Yes
HProcessTOOL: A support tool in the harmonization of multiple reference models [19]	No	Yes	No	Yes
Building a Maturity & Capability Model repository [25]	Yes	No	No	Yes

4 Conclusion

This study allowed us to evaluate the range of researches in the context of SPA and SPI, to answer research questions identified in the planning of this paper, which can be summarized as follows:

1. What are the types of approaches that support the evaluation and improvement of software processes (SPA and SPI)?
   - It was possible to identify several approaches for performing SPI and SPA.
2. Do such approaches allow the excecution of the evaluation considering one or more quality models?
   - The approaches that used computational support do not provide the implementation of SPA/SPI considering more than one quality model. Only harmonization of quality models initiatives using abstraction models were proposed in order to instantiate more than one quality model, however, the approach used has some limitations such as: modeling is a manual and error-prone task; the approaches have not evolved toward other studies, as it was not identified studies that continued the evolution of a methodology solving its limitations and possibilities for future work identified by the authors; the usability was not evaluated, in most cases, validation is a way to create a development scenario to apply the proposed method.
3. Do the approaches enable the reuse of processes and quality models?
   - All the analyzed approaches concentrate the information collected in their own environments, with no possibility of extensions using web services.
4. What criteria are used by these approaches to suggest process improvement activities?
   - Most of the tools analysed that make suggestions of SPI activities make use of questionnaires in which, from the users responses, measurement indicators are applied to know how much a practice or objective required by the quality model is satisfied. Then, a gap report is provided from a knowledge base implemented by the tool.
5. How do these approaches make the mapping between the elements of the executed software process and the attributes required by quality models?
   - The evaluation of a software process should allow the mapping between activities executed in the real process and activities that are expected considering a quality reference model. This allows identifying the compliance level and suggest areas for improvement. However, none of the studies presented an approach that allows the user to instantiate the executed process and then evaluate how adherent it is to a quality model. This was the biggest deficiency found in the studies analysed, because the questionnaires are created based on objective criteria (what is expected in the process execution). These criteria are available in evaluation guides of quality models. One of the industry's questions regarding the quality

models is that they do not describe *how* something must be done, but only *what* should be done. However, this can be considered as the merit of these models, which can abstract the main concepts related to software quality and organize them in a structurally and evolutionary way. Thus, the evaluation performed strictly by a tool, making use of only objective criteria, can produce a partial result in determining that some practice has met its compliance criteria. Actually, the main purpose of a quality model practice could be somehow satisfied, but it would be difficult to assess it using only the objective criteria provided by the current methodologies, as it cannot assess something using subjective criteria.

The results presented here are part of a master's degree project that has the objective of identifying deficiencies and limitations that may be addressed via academic research in order to evolve the current paradigm of software quality processes, suggesting a new SPA approach and SPI (still under development), so that together with the software industry community it is possible to achieve better quality software products.

The use of methodologies that make use of centralized environments for managing software quality attributes could allow, in the medium/long-term, to collect information derived from the software engineering community and their projects. The analysis of this information could indicate the real difficulties software engineers have to overcome and provide ways to stretch the relationship between academia and industry.

References

1. Ali, R.Z.R.M., Ibrahim, S.: An application tool to support the implementation of integrated software process improvement for malaysia's sme. In: 2011 5th Malaysian Conference in Software Engineering, MySEC 2011. pp. 177–182 (2011)
2. García, I., Pacheco, C., Mendoza, E., Calvo-Manzano, J.A., Cuevas, G., Feliu, T.S.: Managing the software process with a software process improvement tool in a small enterprise. Journal of software: Evolution and Process 24(5), 481–491 (2012)
3. Gazel, S., Sezer, E.A., Tarhan, A.: An ontology based infrastructure to support CMMI-based software process assessment. Gazi University Journal of Science 25(1), 155–164 (2012)
4. Homchuenchom, D., Piyabunditkul, C., Lichter, H., Anwar, T., Methawachananont, A., Krootkaew, C., Krisanathamakul, T.: Spials-ii: A light-weight software process improvement self assessment tool. International Journal of Digital Content Technology and its Applications 6(21), 16–26 (2012)
5. ISO: [*On-line*], *ISO/IEC 15504* (Junho 2013), Available in http://www.iso.org/iso/home.html (Viewed in 11/06/2013)
6. ISO/IEC: ISO/IEC 9126. Software engineering – Product quality. ISO/IEC (2001)
7. ISO/IEC: ISO/IEC 12207. Software engineering – Product quality. ISO/IEC (2007)
8. Jeners, S., Lichter, H., Gomez Rosenkranz, C.: Efficient adoption and assessment of multiple process improvement reference models. E-Informatica Software Engineering Journal 7(1), 15–24 (2013)
9. Johnson, J.: My Life is Failure: 100 Things You Should Know to Be A Better Project Leader (2006)

10. Kasurinen, J., Runeson, P., Riungu, L., Smolander, K.: A self-assessment framework for finding improvement objectives with ISO/IEC 29119 test standard, Communications in Computer and Information Science, vol. 172 (2011)
11. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - a systematic literature review. Information and Software Technology 51(1), 7–15 (2009)
12. Lepasaar, M., Makinen, T.: Integrating software process assessment models using a process meta model. pp. 224–229. IEEE Computer Society (2002)
13. Mas, A., Mesquida, A.L.: A software tool to support the integrated management of software projects in mature SMEs, Communications in Computer and Information Science, vol. 172 (2011)
14. Mondragn, O.: Addressing infrastructure issues in very small settings. Proceedings of the First International Research Workshop for Process Improvement in Small Settings pp. 5–10 (2006)
15. Montoni, M., Santos, G., Rocha, A.R., Figueiredo, S., Cabrai, R., Barcellos, R., Barreto, A., Scares, A., Cerdeiral, C., Lupo, P.: Taba workstation: Supporting software process deployment based on CMMI and MR-MPS.BR, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4034 LNCS (2006)
16. Musat, D., Castaño, V., Calvo-Manzano, J.A., Garbajosa, J.: MATURE: A model driven based tool to automatically generate a language that supports CMMI process areas specification. Springer (2010)
17. Niazi, M., Wilson, D., Zowghi, D.: A framework for assisting the design of effective software process improvement implementation strategies. Journal of Systems and Software 78(2), 204–222 (2005)
18. Oktaba, H., Piattini, M.: Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies (2008)
19. Pardo, C., Pino, F., García, F., Romero, F.R., Piattini, M., Baldassarre, M.T.: HProcessTOOL: A support tool in the harmonization of multiple reference models, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6786 LNCS (2011)
20. Park, E., Kim, H., Lee, R.Y.: Software repository for software process improvement. Studies in Computational Intelligence, vol. 131, pp. 51–64 (2008)
21. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering
22. Rungratri, S., Usanavasin, S.: Semantic based approach supporting CMMI gap analysis process. Journal of Convergence Information Technology 7(20), 127–137 (2012)
23. SEI: [*On-line*], *Portal CMMI* (Junho 2013), Available in http://www.CMMIinstitute.com/ (Viewed in 11/06/2013)
24. Softex: [*On-line*], *Guia Geral MPS de Software* (Agosto 2012), Available in http://www.softex.br/mpsbr/guias/ (Viewed in 12/10/2013)
25. Von Wangenheim, C.G., McCaffery, F., Hauck, J.C.R., Lacerda, T.C., Buglione, L., Vieira Da Cruz, R.F.: Building a maturity & capability model repository. In: ACM International Conference Proceeding Series. pp. 2–5 (2011)
26. Yeom, H.., Hwang, S..: A design of tool for software processes assessment and improvement. In: Proceedings of the 2008 Advanced Software Engineering and its Applications, ASEA 2008. pp. 245–248 (2008)