Information and Software Technology 58 (2015) 20–43

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

# Using CMMI together with agile software development: A systematic review

Fernando Selleri Silva [a,b,*], Felipe Santana Furtado Soares [a,c], Angela Lima Peres [a,d], Ivanildo Monteiro de Azevedo [a,b], Ana Paula L.F. Vasconcelos [a,b], Fernando Kenji Kamei [a,e], Silvio Romero de Lemos Meira [a,c]

[a] *Center of Informatics (CIn), Federal University of Pernambuco (UFPE), Recife, PE, Brazil*
[b] *Computer Science Course (FACET), Mato Grosso State University (UNEMAT), Barra do Bugres, MT, Brazil*
[c] *Recife Center of Advanced Studies and Systems (C.E.S.A.R), Recife, PE, Brazil*
[d] *Cesmac University Center, Maceió, AL, Brazil*
[e] *Federal Institute of Alagoas (IFAL), Arapiraca, AL, Brazil*

A R T I C L E  I N F O

*Article history:*
Received 18 April 2014
Received in revised form 21 August 2014
Accepted 27 September 2014
Available online 6 October 2014

*Keywords:*
Software process improvement
CMMI
Agile methodology
Benefits
Limitations
Systematic review

A B S T R A C T

*Background:* The search for adherence to maturity levels by using lightweight processes that require low levels of effort is regarded as a challenge for software development organizations.
*Objective:* This study seeks to evaluate, synthesize, and present results on the use of the Capability Maturity Model Integration (CMMI) in combination with agile software development, and thereafter to give an overview of the topics researched, which includes a discussion of their benefits and limitations, the strength of the findings, and the implications for research and practice.
*Methods:* The method applied was a Systematic Literature Review on studies published up to (and including) 2011.
*Results:* The search strategy identified 3193 results, of which 81 included studies on the use of CMMI together with agile methodologies. The benefits found were grouped into two main categories: those related to the *organization* in general and those related to the *development process*, and were organized into subcategories, according to the area to which they refer. The limitations were also grouped into these categories. Using the criteria defined, the strength of the evidence found was considered *low*. The implications of the results for research and practice are discussed.
*Conclusion:* Agile methodologies can be used by companies to reduce efforts in getting to levels 2 and 3 of CMMI, there even being reports of applying agile practices that led to achieving level 5. However, agile methodologies alone, according to the studies, were not sufficient to obtain a rating at a given level, it being necessary to resort to additional practices to do so.
© 2014 Elsevier B.V. All rights reserved.

**Contents**

1. Introduction . . . 21
2. Background . . . 21
   2.1. Capability Maturity Model Integration (CMMI) . . . 22
   2.2. Agile software development . . . 22
   2.3. Summary of previous reviews . . . 22
3. Review method . . . 23
   3.1. Research questions . . . 22
   3.2. Data sources . . . 23

* Corresponding author at: Center of Informatics, Federal University of Pernambuco (CIn/UFPE), Av. Jornalista Anibal Fernandes, S/N, Cidade Universitária, 50.740-560, Recife, PE, Brazil. Tel.: +55 81 8250 0470.

*E-mail addresses:* fss4@cin.ufpe.br (F. Selleri Silva), fsfs@cin.ufpe.br (F.S.F. Soares), alp5@cin.ufpe.br (A.L. Peres), ima2@cin.ufpe.br (I.M.d. Azevedo), aplfv@cin.ufpe.br (A.P.L.F. Vasconcelos), fkk@cin.ufpe.br (F.K. Kamei), srlm@cin.ufpe.br (S.R.d.L. Meira).

http://dx.doi.org/10.1016/j.infsof.2014.09.012
0950-5849/© 2014 Elsevier B.V. All rights reserved.

3.3. Search terms . . . 23
3.4. Criteria for selecting a study . . . 23
3.5. Procedures for selecting a study. . . . 24
3.6. Quality assessment . . . 24
3.7. Data extraction . . . 25
3.8. Data synthesis . . . 25
3.9. Conducting the review . . . 25
3.9.1. Automatic search . . . 25
3.9.2. Manual search. . . . 25
3.9.3. Potentially relevant studies . . . 25
3.9.4. Study inclusion and exclusion . . . 26
4. Results. . . . 26
4.1. Methodological quality . . . 26
4.2. Research methods of the studies . . . 27
4.3. Aim of the studies . . . 27
4.4. General characteristics . . . 28
4.5. CMM/CMMI levels and process area . . . 29
4.6. Agile methodologies and practices. . . . 29
5. Discussion. . . . 30
5.1. Benefits and limitations . . . 30
5.1.1. Organization . . . 30
5.1.2. Process understanding and communication . . . 31
5.1.3. Management and configuration. . . . 31
5.1.4. Requirements and tests . . . 31
5.1.5. Maturity . . . 32
5.1.6. Productivity and quality. . . . 32
5.2. Strength of evidence . . . 32
5.3. Implications for research and practice. . . . 33
5.4. Limitations of this review. . . . 33
6. Conclusion and future work . . . 34
Acknowledgements . . . 34
Appendix A. Studies included in the review . . . 35
Appendix B. Overview of studies included. . . . 38
Appendix C. Quality assessment result . . . 40
Appendix D. Distribution of studies by publication channel and occurrence . . . 42
References . . . 43

## 1. Introduction

Maturity models, such as *Capability Maturity Model Integration* (CMMI) [1], have been presented in recent times as a resource that software development organizations may be required to use if they are to win and retain more customers. An organization that offers a rating at the highest levels of these maturity models excels in competitions for software projects. These affirmations could be supported by the number of CMMI appraisals which have been increased over the years [2]. However, the direct implementation of maturity model practices, as opposed to the proper assimilation of the principles and objectives that underlie these models, should be avoided. Bearing these concerns in mind, many organizations have taken greater interest in *agile software development* in order to combine maturity and agility.

Agile software development has its main values stated in the Agile Manifesto [3] and is represented by agile methodologies, such as *Extreme Programming* [4] and *Scrum* [5], which are approaches, in general, that focus on collaboration in a flexible way. These methodologies arose from the need to establish processes that address the development of systems more quickly and with quality. They employ an iterative and incremental life cycle, with short iterations and requirements that can be modified throughout the development, with extensive participation by the customer.

The systematic review in this paper sets out to evaluate, synthesize, and present results on the use of CMMI in combination with agile software development, using studies published up to (and including) 2011, and thereafter to give an overview of the topics researched, which includes a discussion of their benefits and limitations, the strength of the findings, and the implications for research and practice. This review is important for Software Engineering because it analyses together the current main approaches regard software development and software process improvement. Some statistics, as [2,6], show that the use of CMMI and agile is increasing in the software industry. Moreover, the number of studies included in this review shows that their use together also has increased.

The focus in this review is on CMMI because it is a maturity model used by many organizations all over the world, as reported in [2], and its use has been one of the factors related to agile development, identified in previous reviews, such as that by Hasnain [7]. In future, research may well be extended to further study on other maturity models, such as the *Reference Model for Brazilian Software Process Improvement* (MR-MPS.BR) [8], which was the subject of another review, published in Souza et al. [9].

The remainder of this paper has the following organization. Section 2 gives a brief theoretical description of CMMI, agile methodologies and previous reviews in this area. Section 3 describes the research methodology. Section 4 reports the general results. Benefits and limitations, the strength of the evidence and implications for research and practice are discussed in Section 5. Conclusions and recommendations for future research are set out in Section 6.

## 2. Background

Initially, the CMMI will be briefly described and, then, the main concepts of agile methodologies. This section ends with a brief overview of previous reviews of these fields of work.

### 2.1. Capability Maturity Model Integration (CMMI)

*Capability Maturity Model Integration* (CMMI) [1] is a project that deals with the integration of maturity models in software production, originally designed by the Software Engineering Institute (SEI) of Carnegie Mellon University, and is supported by software development organizations and government entities. It aims to consolidate a framework for models, and evolve and integrate models derived from the Capability Maturity Model (CMM) for Software, also proposed by SEI, the focus being on organizational capacity.

When this review was being undertaken, CMMI for Development (CMMI-DEV), the model considered here, was in version 1.3, which was released in November 2010. However, the review considers related work on previous versions, including work on CMM for software. For the purposes of nomenclature, in this paper only the term CMMI is used.

The main objective of CMMI is to reduce the cost of implementing improvements in processes by eliminating inconsistencies and establishing guidelines to assist organizations at various stages of a software project (planning, management, and others). Its architecture is comprised by defining a set of 22 process areas, arranged in two representations: one per stage, in which the areas are grouped into five levels of maturity; and another continuous representation, where four levels of capability are set.

CMMI is compliant with other standards and quality models, such as CMM, ISO 12207 and ISO 15504 [1]. Its use has been widespread in assisting the conduct of software projects in organizations [2]. However, in certain projects some concepts of CMMI should be adapted or removed in order to facilitate development. This adaptation feature should be considered by organizations seeking to employ CMMI with agile methodologies in software production.

### 2.2. Agile software development

Agile methodologies (or lightweight methodologies) have been proposed as an alternative to traditional methodologies of software development, which are also referred to as plan-driven, heavy, systematic or Tayloristic methodologies. The common principles of agile methodologies are described in a document entitled the "Manifesto for Agile Software Development" or simply the "Agile Manifesto" [3]. This document was proposed during a meeting in Utah of seventeen experts on software development in 2001. According to the manifesto, agile methodologies emphasize the value of the following concepts [3]:

– *Individuals and interactions* over processes and tools;
– *Working software* over comprehensive documentation;
– *Customer collaboration* over contract negotiation;
– *Responding to change* over following a plan.

Agile methodologies do not totally ignore processes and tools, documentation, contract negotiation and planning, but they assume that the software, especially the working code, should be the focus of development. Thus, agile methodologies consider that individuals and interactions, working software, customer collaboration and responding quickly to changes are concepts of greater value [3]. With an approach that prioritizes real knowledge about system functionality, agile methodologies directly stimulate software production, seek constant improvements as they have short iterations, and see to it that team members exchange their knowledge and experience. Boehm [10] lists various agile methodologies available for use in software development projects, such as *Adaptive Software Development* (ASD) [11], *Crystal* [12], the *Dynamic Systems Development Method* (DSDM) [13], *eXtreme Programming* (XP) [4], *Feature Driven Development* (FDD) [14], *Lean Software Development* [15] and *Scrum* [5].

### 2.3. Summary of previous reviews

Dybå and Dingsøyr [16] present a systematic review of agile software development including qualitative and quantitative empirical studies published up to 2005. In this review, they cite three reports, which describe the then state of the art and state of practice in terms of features of various agile methodologies in the industry, namely: Abrahamsson et al. [17]; Cohen, Lindvall and Costa [18]; and Erickson, Lyytinen and Siau [19].

Another systematic review, conducted by Hasnain [7] set out to identify the type of research done on agile methodologies in studies published in the proceedings of Agile Conferences from 2003 to 2007. The review focused on the context of the paper (professional or academic), type of study (experience or empirical), the kind of agile methodology, and the factors (human or technical). CMMI was cited as one of the factors discussed in these papers.

Staples and Niazi [20] conducted a systematic review on why organizations adopt approaches to Software Process Improvement (SPI) based on CMM and how these motivations relate to the size of an organization. The results pointed to issues such as software quality, development time, development cost, and productivity. Reasons related to Product and Performance are given more prominence than those related to Process, Customer and People.

Sulayman and Mendes [21] identify the occurrence of SPI models and techniques used by small and medium web companies. Their paper emphasized that SPI models use an interactive approach influenced by CMM/CMMI. Unterkalmsteiner et al. [22] present a systematic review that sought to identify and characterize assessment strategies and measures used to assess the impact of different initiatives in SPI.

A *quasi*-systematic review (involving only automatic searches) was developed by Magdaleno, Werner and Araujo [23]. It set out to characterize reconciliation among plan-driven, agile, and free/open source software development models. The paper suggests, as opportunities for reconciliation, the following factors: collaboration; code availability; cost reduction; and quality improvement. The challenges they list are: organizational culture; number of artifacts; knowledge management; organizational structure; system of recognition and individual reward; communication; decision making; quality assurance; requirements; planning work; monitoring work in progress; customer relationships; maintenance of certified processes. Studies using XP and Scrum allied to CMMI and ISO were included.

With a view to finding a systematic review specifically on CMMI and agile, that had already been undertaken, an automatic search was performed in September 2011 in the following electronic databases: *IEEE*, *ACM*, *Scopus*, *Compendex*, *ISI*, *Science Direct*, *Wiley*, and *SpringerLink*. The search used the following terms: *CMMI*, *Capability Maturity Model* or *CMM*; and *agile*, *agility*, *Scrum*, *Extreme Programming*, *XP*, *Dynamic System Development*, *DSDM*, *Crystal Clear*, *Crystal Orange*, *Crystal Red*, *Crystal Blue*, *Feature Driven Development*, *FDD*, *Lean Software Development*, *Adaptive Software Development*, *ASD*, *Test Driven Development* or *TDD*; and *systematic review*, *systematic literature review* or *SLR*. Altogether, the search returned 471 individual studies, after which a check was made on each title and abstract. The outcome was that no published systematic review of CMMI and agile development was found. This reinforces the importance of the review undertaken in this paper. Thus, this paper offers as a contribution, in relation to previous reviews, a discussion that helps organizations and researchers in the definition of software development processes that obtain the benefits of both the maturity and agility approaches together.

**Table 1**
Search terms.

| Keywords | Synonyms or related words |
|---|---|
| CMMI | Capability Maturity Model, CMM |
| Agile | Agility, lightweight, scrum, extreme programming, XP, dynamic system development, DSDM, crystal clear, crystal orange, crystal red, crystal blue, feature driven development, FDD, lean software development, adaptive software development, ASD, test driven development, TDD |
| Software development | Software engineering, software production, software project, system development, system engineering, system production, system project, application development, application engineering, application production, application project |

## 3. Review method

According to Kitchenham and Charters [24], a systematic review seeks to identify, evaluate and support all the relevant studies currently available for a specific research question, subject area, or phenomenon of interest. The definition of a protocol is important and necessary to reduce the possibility of bias in the search because the protocol specifies the methods used to guide the systematic review. This protocol includes all elements of the analysis proposed and some additional planning. A summary of the protocol used in this review is given in the sub-sections below.

### *3.1. Research questions*

This systematic review seeks to answer the following three research questions (RQs), based on questions used in the paper by Dybå and Dingsøyr [16]:

*RQ1*. What is currently known about the benefits and limitations of using CMMI in combination with agile software development?
*RQ2*. What is the strength of the evidence supporting the claims for using CMMI and agile software development?
*RQ3*. What are the implications of the included studies for the software industry and the scientific community?

### *3.2. Data sources*

The search strategy to find the studies included automatic searches in electronic databases and a manual search in conference proceedings and journals, to ensure that the greatest number of studies were checked, even if this caused redundancy in the results. Automatic searches included the most relevant indexation mechanisms of scientific studies, and a search for PDF files on the SEI website, as suggested by Staples and Niazi [20]. The following electronic databases were searched:

– *ACM Digital Library*;
– *Compendex*;
– *IEEE Xplore*;
– *ISI Web of Science*;
– *ScienceDirect – Elsevier*;
– *Scopus*;
– *SEI – PDF Documents (searched with Google)*;
– *SpringerLink*;
– *Wiley Inter Science Journal Finder*.

Furthermore, some conference proceedings and journals were selected on which a manual search was performed. Initially, the conference proceedings listed in Dybå and Dingsøyr [16] were considered, and some important software engineering journals were added. To this end, a search by the category "Computer Science, Software Engineering" was conducted on the site of the Journal Citation Reports, classified by the criterion "5-Year Impact Factor", which gave the top five ranked. Journals clearly not about software development, such as ACM Transactions on Graphics (best rated) and IEEE Transactions on Dependable and Secure Computing were eliminated. The Empirical Software Engineering Journal, which is the top-ranked journal, focused on empirical data, and the Agile Journal, which according to Racheva et al. [25] is the most popular site of online publication focused on professionals in the agile community, were included. Thus, the list of conference proceedings and journals considered for the manual search consisted of:

– Conference Proceedings of: *XP Conference* (XP); *Agile Development Conference* (AGILE); *International Symposium on Empirical Software Engineering and Measurement* (ESEM); and *International Conference on Software Engineering* (ICSE).
– Journals: *IEEE Transactions on Software Engineering; Journal of the ACM* (JACM); *ACM Transactions on Software Engineering and Methodology* (TOSEM); *IEEE Software; Empirical Software Engineering Journal; Journal of Software Process: Improvement and Practice*; and the *Agile Journal*.

### *3.3. Search terms*

The search in electronic databases used keywords derived from the research questions. We also decided to include some synonyms or related words for composing search terms. Related words were obtained from previous studies in the area, such as Staples and Niazi [20] and Magdaleno et al. [23]. Table 1 presents the keywords and their synonyms or related words that correspond to the search terms.

The strategy used to construct the search string was:

1. Derive keywords of research questions;
2. Identify synonyms or related words for keywords;
3. Grouping synonyms and related words with the identifier "OR";
4. Group each set of terms with the identifier "AND".

Stages 1 and 2 were performed as described in Table 1. After stages 3 and 4, the following results for the search string were obtained:

*("CMMI" OR "capability maturity model" OR "CMM")*

*AND*

*("agile" OR "agility" OR "lightweight" OR "scrum" OR "extreme programming" OR "XP" OR "dynamic system development" OR "DSDM" OR "crystal clear" OR "crystal orange" OR "crystal red" OR "crystal blue" OR "feature driven development" OR "FDD" OR "lean software development" OR "adaptive software development" OR "ASD" OR "test driven development" OR "TDD")*

*AND*

*("software development" OR "software engineering" OR "software production" OR "software project" OR "system development" OR "system engineering" OR "system production" OR "system project" OR "application development" OR "application engineering" OR "application production" OR "application project")*

### *3.4. Criteria for selecting a study*

Studies in the review were selected as per the inclusion and exclusion criteria given in this sub-section.

The following *inclusion criteria* were considered:

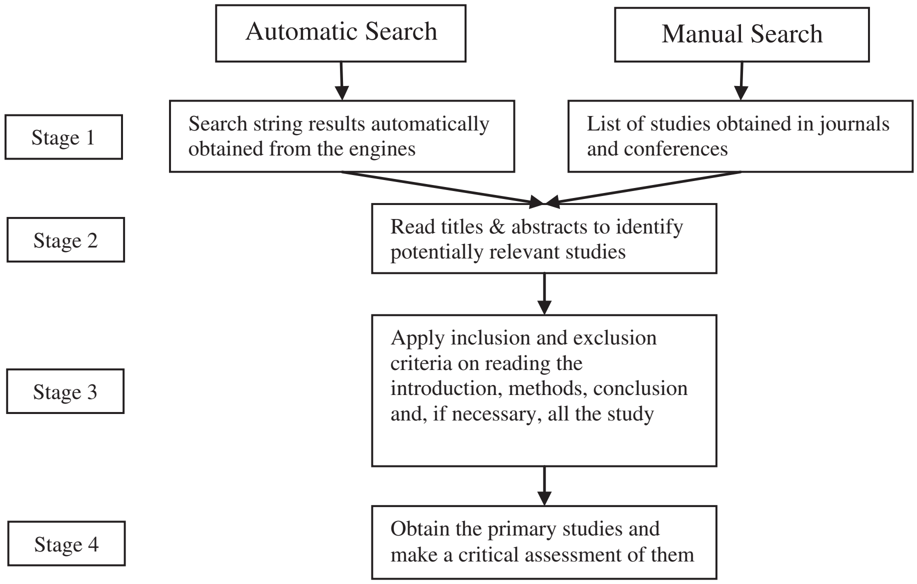


**Fig. 1.** Stages of study selection process.

– Academic and industry studies are included;
– Studies that show empirical data, theoretical or experience reports on CMMI and agile software development;
– Studies of qualitative and quantitative research;
– Only studies written in English;
– Studies published up to and including 2011.

As *exclusion criteria* were adopted:

– Studies not focused on CMMI and agile development;
– Studies that focus on simple techniques or practices such as pair programming, unit testing and refactoring, applied to a non-agile process, such as the Unified Process;
– Studies merely based on expert opinion without locating a specific experience;
– Editorials, prefaces, summaries of articles, interviews, news, analysis (reviews), correspondence, discussions, comments, readers' letters, summaries of tutorials, workshops, panels, and poster sessions.

This review opted to include experience reports and exclude expert opinion, when this expert opinion did not have a particular context. Given the importance of conducting a survey that mapped the subject of this review, we also included studies with theoretical discussions or proposals for models and tools, which have not been implemented in practice, but have been substantiated by the literature.

### 3.5. Procedures for selecting a study

Selecting a study was conducted by following four stages in order to obtain a set of primary studies. Each stage is shown in Fig. 1 and described below.

– *Stage 1*: conduct of the automatic and manual search in order to identify a preliminary list of studies. Duplicate studies were discarded.
– *Stage 2*: identification of potentially relevant studies, based on the analysis of title and abstract, discarding studies that are clearly irrelevant to the search. If there was any doubt about whether a study should be included or not, it was included for consideration at a later stage.
– *Stage 3*: selected studies in previous stages were reviewed, by reading the introduction, methodology section and conclusion and applying the inclusion and exclusion criteria presented in Section 3.4. If reading the above items was not enough to make a firm decision, the study was read in its entirety.
– *Stage 4*: thus, a list of primary studies was obtained and later subjected to critical examination using the criteria set out in Section 3.6.

### 3.6. Quality assessment

Quality assessment of this review followed 11 criteria established by Dybå and Dingsøyr [16]. These criteria are listed below:

1. Is the paper based on research (or is it merely a "lessons learned" report based on expert opinion)?
2. Is there a clear statement of the aims of the research?
3. Is there an adequate description of the context in which the research was carried out?
4. Was the research design appropriate to address the aims of the research?
5. Was the recruitment strategy appropriate to the aims of the research?
6. Was there a control group with which to compare treatments?
7. Was the data collected in a way that addressed the research issue?
8. Was the data analysis sufficiently rigorous?
9. Has the relationship between researcher and participants been considered to an adequate degree?
10. Is there a clear statement of findings?
11. Is the study of value for research or practice?

According to Dybå and Dingsøyr [16], these criteria include three important issues, related to quality, which were considered for this review, namely:

– *Rigor*: has a complete and adequate approach been applied to key research methods in the study?
– *Credibility*: are the results well-presented and in a meaningful way?

– *Relevance*: how useful are the results to the software industry and the scientific community?

For evaluation each of the 11 criteria was graded on a dichotomous scale ("yes" or "no"), receiving a score of "1" for "yes" and "0" for "no".

### 3.7. Data extraction

Strategies used in the data extraction included a spreadsheet editor (Microsoft Excel™). All relevant information on each study was recorded on a spreadsheet. This information was helpful when summarizing the data as it made it possible to map each datum extracted with its source.

The following data were extracted from the description of the studies:

(i) references;
(ii) type of article (journal, conference, workshop);
(iii) geographic location (country);
(iv) aim of the study;
(v) research question.

The following data were extracted on study results:

(i) type of design of the study (empirical, experience report, theoretical);
(ii) research method (case study, experiment, action research, survey);
(iii) methodology of the analysis (qualitative, quantitative);
(iv) research hypothesis;
(v) control group;
(vi) data collection;
(vii) sample description – subject, size, age, education, experience;
(viii) scenario;
(ix) project domain;
(x) project duration;
(xi) agile methodology and practices;
(xii) CMM/CMMI level and process areas;
(xiii) results and conclusions;
(xiv) benefits;
(xv) limitations and challenges;
(xvi) validity;
(xvii) relevance.

### 3.8. Data synthesis

This synthesis aimed to group findings from the studies in order to: identify the main concepts (organized in spreadsheet form), conduct a comparative analysis on the characteristics of the study, using demographic data, temporal distribution, type of agile methodology and practices, and CMM/CMMI level and process areas; seek answers to the research questions. Other information was synthesized when necessary. The meta-ethnography method, presented by Noblit and Hare [26], was found to be of help in the process of data synthesis.

### 3.9. Conducting the review

The review started with an automatic search and this was followed by a manual search, during which potentially relevant studies were identified, and inclusion/exclusion criteria were applied. A detailed description of each stage is given below. The search process, from testing to the final search lasted from November 2011 to March 2012, and included using the search strings previously defined in the search engines and conducting the separate manual search.

#### 3.9.1. Automatic search

The first tests using automatic search began in November 2011. In some engines, the search string had to be adapted, without losing its primary meaning and scope. After performing the search, the results were tabulated on a spreadsheet so as to facilitate the subsequent phase of identifying potentially relevant studies. In March 2012, a final search was performed in the mechanisms with a view to including studies published in 2011. Table 2 presents the results obtained from each electronic database used in the search, which gave an overall total of 3782 results.

#### 3.9.2. Manual search

The separate manual search was conducted in January and February 2012 by analyzing the titles and abstracts (if necessary) of studies published in conference proceedings and journals. Those considered potentially relevant were tabulated on an Excel spreadsheet. Table 3 presents the results for each source, an overall total of 61 results.

#### 3.9.3. Potentially relevant studies

Results obtained from the automatic search and manual search were included on a single spreadsheet: an overall total of 3843 results, namely 3782 from the automated search plus 61 from the separate manual search. The studies were sorted by title in order to eliminate redundancies. Studies for which the title, author(s), year and abstract were identical were considered redundant. After removing redundant items, we were left with *3193 results.*

The titles and abstracts of studies, resulting from the automatic search, were read to identify potentially relevant studies. Some studies were from medical or chemistry area, because these areas also use the acronym CMM. Many studies only cited terms related

**Table 2**
Automatic search results.

| Electronic database | Result |
|---|---|
| IEEE | 1042 |
| SpringerLink | 810 |
| Scopus | 675 |
| Science Direct | 314 |
| ACM | 309 |
| SEI | 306 |
| Wiley | 245 |
| Compendex | 72 |
| ISI | 9 |
| Total | 3782 |

**Table 3**
Manual search results.

| Source | Result |
|---|---|
| Agile Conference | 16 |
| XP Conference | 13 |
| IEEE Software | 12 |
| ESEM | 7 |
| IEEE Transactions | 4 |
| Agile Journal | 3 |
| ICSE | 3 |
| Software Process | 2 |
| TOSEM | 1 |
| ESEJ | 0 |
| Journal of the ACM | 0 |
| Total | 61 |

to agile and CMMI, but not discussed them together. These studies were discarded. At the end of this stage, we obtained a total of 438 potentially relevant studies, which were considered in the next phase of the review: the application of inclusion and exclusion criteria.

During data extraction, the bibliographical references of the studies included were checked to ensure greater coverage. The scan identified an additional of seven potentially relevant studies, not identified in the previous searches. Thus, the grand total of potentially relevant studies was *445 studies*.

#### 3.9.4. Study inclusion and exclusion

In this stage, we analyzed the introduction, methodology, conclusion, and in case of doubt, the other sections of each study. One researcher reviewed all the studies identified as potentially relevant, which gave rise to a total of *81 studies* being used for our systematic review. Subsequently, the list of excluded studies, with their title, abstract, author(s), year and link to complete text, was sent to a specialist in the area to check out whether some relevant work was deleted.

Access to the majority of potentially relevant studies was gained via the Federal University of Pernambuco portal of journals. Others, not available via the portal, were obtained by e-mail request to the author(s), most of whom responded promptly and expressed interest in our research study. One article had to be purchased by researchers.

Only three studies indicated as potentially relevant were not found for a full reading in order to apply the inclusion and exclusion criteria. For these studies, we adopted the following procedure:

1. We read the abstract, keywords, references and other information on the site to which the search engine referred.
2. We checked whether the references cited by the study included articles that might be relevant for our research purposes.
3. We checked whether or not the same authors or at least with one of the authors, in the case of co-authorship, had published any other potentially relevant articles.
4. We checked how often the study had been cited by other studies (verifiable via the search engine), especially for studies included or listed as potentially relevant.

After analyzing this information, it was deduced that none of these studies focused on the use of CMMI together with agile methodologies, and they were therefore excluded from the process.

In general, we excluded studies that compare results obtained from using CMMI to results from using agile methodologies, given that such papers are beyond the scope of this research, which investigates the use of CMMI and agile methodologies together. Studies published in 2012 or after were not included, but it is emphasized that among the results, obtained for this more recent period, at least the study of McMahon [27] includes contributions that fit within this research.

Five of the seven potentially relevant studies which were identified after reviewing the references of the 76 studies initially included, arising from the automatic and manual searches, were included. This reinforces the importance of the technique of reviewing the references given in papers initially identified with a view to finding other relevant studies.

## 4. Results

We identified 81 studies about the joint use of CMMI and agile methodologies, which are listed in Appendix A. Of these, 23 studies (28%) are considered empirical, 38 (47%) are experience reports and 20 (25%) are theoretical. This section discusses the (empirical and non-empirical) studies included. They present different approaches to the topic of this review; apply different research methods; were developed in different environments; and range from projects developed in large, medium and small enterprises to those arising from academic experiences. Appendix B lists an overview of the studies included. Data relating to the domain description in which each study was developed indicate that many refer to the financial, telecommunications, government or military, aerospace and aviation, and automotive areas, and complex and critical systems. However, areas related to small and medium enterprises, web development and even game development have also become interested in software process improvement with CMMI and agile methodologies.

The methodological quality, research methods applied, aims and general characteristics of the studies, such as publication data, CMMI levels and areas, and the agile methodologies and practices adopted, are described in the following sub-sections.

### 4.1. Methodological quality

Quality assessment used the criteria suggested in Dybå and Dingsøyr [16], as defined in Section 3.6. Although empirical and non-empirical studies are included, non-empirical studies are rarely evaluated positively in criteria 4, 5, 6 and 9, which correspond to the research design, sampling, control group and reflexivity. The 81 studies included were listed on a spreadsheet. Each study was thoroughly read and each of the eleven criteria evaluated. The results of quality assessment of the studies included are presented in Appendix C, where "1" means "yes" (or OK) to the criterion, while "0" indicates "no" (or not OK). Fig. 2 shows the score result for each quality criterion evaluated.

With regard to the inclusion of empirical studies, theoretical studies and experience reports, not all studies were scored with OK on the first criterion for the type of research. Altogether 23 studies (28%) were based on empirical research, while 58 (72%) are theoretical studies or experience reports from industry or academia. As to the second criterion of evaluation, 30 studies did not state a clear research aim, of which one was an empirical study, and the others were experience reports or theoretical studies. Twenty-six studies did not describe clearly the context in which the research was conducted. As measured against the third criterion, none of them are empirical.

The fourth criterion sought to assess whether the research design proved suitable for the defined aims. On this criterion, only 10 studies were evaluated positively. However, we emphasize that this criterion is applied only to empirical studies and that the only studies that received an OK were those studies which spell out in

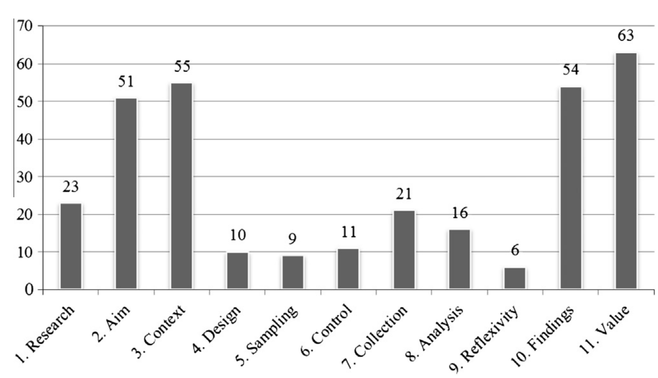


**Fig. 2.** Score for quality criteria evaluated.

the text a rationale for the choice of research method, which therefore restricted these results. On the fifth criterion, also applied to empirical studies, only nine studies were considered to have had a recruitment strategy appropriate to the aims set for the research. Eleven empirical studies include one or more control groups, for comparison purposes, as judged against the sixth criterion.

It was considered that 21 studies adequately described the procedures for data collection, as assessed on the seventh criterion. Of these, only three studies are non-empirical (2 experience reports and 1 theoretical). As to the eighth criterion, 16 studies adequately described the procedures for data analysis, there being only one experience report. The possibility of bias having been introduced by the researchers, as well as the strategies adopted to avoid it, was mentioned only in six empirical studies, based on the ninth criterion.

With regard to the tenth assessment criterion, it was considered that 27 studies, of which one is an empirical study and the others non-empirical, did not give a clear description of their findings. In 18 of the non-empirical studies, it was not possible to identify the value of the study to research or practice, as required by the last criterion.

Only two studies obtained the maximum total score in the quality evaluation. Four empirical studies had only one negative response. Six empirical studies had two or three negative responses, while 11 studies, one of which was non-empirical, had four or five negative responses. The remaining 58 studies had six or more negative responses. Among the non-empirical studies, the fewest number of negative responses was five, and this was obtained by only one study. Other non-empirical studies had seven or more negative responses. The highest number of negative responses for empirical studies was six (one study), while all the responses for seven non-empirical studies were negative.

### 4.2. Research methods of the studies

For identification of the research method or type of study, the classification presented by Dybå and Dingsøyr [16] for empirical studies was considered. They classified empirical studies as experiment, survey, case study (single-case and multi-case) and action research. The article type or genre of articles described by Montesi and Lago [28] was also considered mainly for non-empirical studies, but with some observations. From them, we consider experience reports and theoretical studies. However, as survey, we consider studies that applied some kind of opinion research with participants answering questionnaires or interviews and, thus, these studies were considered as empirical studies, according to [16]. Literature surveys were considered as theoretical papers, as described below. Short papers and opinion papers were not included.

The number of studies included by research method is shown in Fig. 3. Altogether, there were 38 experience reports, 20 theoretical studies, 9 case studies (single-case), 6 case studies (multi-case), 6 surveys, 1 action research and 1 study with various methods (mixed). The total of empirical research studies (23) was smaller than that of non-empirical studies (58). This highlights the importance of conducting empirical studies with the aim of increasing evidence of the effects of using CMMI and agile methodologies jointly based on the findings of such studies. Appendix B details the method used for each study.

Of the 38 experience reports, only one study [s37] was conducted with students as subject. The others involved industry-based professionals and experiences of which six dealt with the responses of an experienced team.

The 20 theoretical studies refer to work with a bibliographic basis, which presented proposals not yet implemented and therefore which have not been evaluated in practice. They represent early-stage research, with preliminary assessments, or are in process of development, and were included in order to provide a broader view of what has been discussed in relation to the subject of this review, although they do not present more concrete evidence about their considerations. Purely bibliographical or secondary and tertiary studies (systematic literature reviews) which did not put forward proposals were excluded.

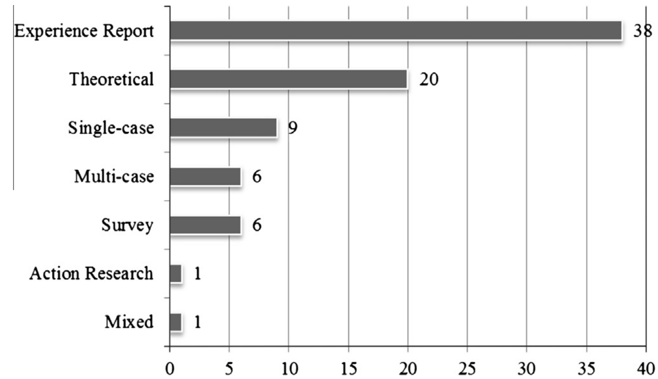


**Fig. 3.** Studies by research method.

All surveys were conducted with professionals. One study [s65] highlighted the case of experienced professionals.

The single-cases involved industry professionals, three of which emphasized that the professionals were experienced. The multi-cases were also conducted with professionals, with two studies being about experienced teams.

One study [s14] that used action research as a method was developed with professionals. One study [s39] brought together a multi-case method and action research, which was considered a mixed study. The participants were students and professionals, both groups being experienced in agile software development projects.

### 4.3. Aim of the studies

The studies included are related to each other in various ways, either because they address (a) common issues related to the various stages of software development (management, requirements, development, deployment, and other), (b) the same agile practices or CMMI process areas, or (c) different organizational aspects (adoption, human factors, perspectives, comparisons, and other). Studies related to various stages of software development were found, ranging from characterizing the process, its actual development, to maintenance; from configuration management support to database system support; from small and medium-sized enterprises to large corporations; from companies that develop small systems and web systems to complex systems of high precision and reliability; and from companies that develop products for their own use or for sale in the market, to companies acting as subcontractors in the production of software or operating in global software development.

Some studies also establish a linear complementarity relation for reporting different aspects of the same research or experience. As an example, study [s24], which focuses on the process of obtaining CMM Level 2 using agile methodologies, is complemented with a study [s15], which focuses on the development process considering an environment of high volatility, and [s1], which describes the experience of constituting a team of agile software development over seven years, combining CMM and agile methodologies. Studies [s40,s46,s54,s55,s78] complement each other because they report, in the context of a large organization, an experience of using CMMI with agile methodologies, starting with Scrum

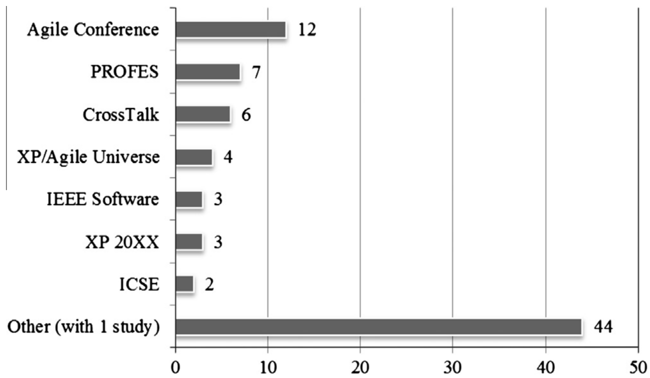


**Fig. 4.** Publication channels with the most studies.

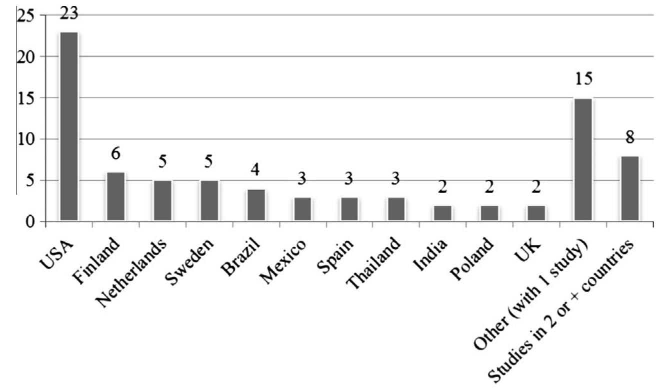


**Fig. 5.** Countries of origin of the studies.

[s55,s46], through Lean [s54,s78], and ending with the inclusion of feature-driven practices in [s40]. Other studies that are complementary are: [s5,s14], which propose and implement, respectively, a method for assessing software process improvement for small and medium enterprises; [s8,s9], which discuss and evaluate CMMI components with respect to their support for agile methodologies; [s33,s34], which report on and expand to other levels the experience of implementing CMMI and agile in a large organization in the energy sector. For repetitive studies or studies, the results of which were included in their entirety in another study, only the latest study or the most complete study was considered, depending on the situation.

In the context of this review, an analysis was made of how the studies are related to each other based on their aim, a methodology which was also adopted by Dybå and Dingsøyr [16] and is very common in the literature. Based on their objectives, studies can be associated with four major groups: (1) fifty-two studies describe experiences of using CMMI and agile methodologies effectively in industry or academia; (2) twenty-nine studies propose models, frameworks, tools or theoretical discussions (approaches) on the topic; (3) twenty studies analyze the compatibility between CMMI and agile, and map CMMI process areas with agile practices; and (4) five studies focus on models for evaluation in CMMI, considering agile practices. Some studies were placed in more than one group, according to their aims. Motivations for adopting CMMI and agile together were given such as: improving the software process [s2,s24]; meeting business goals [s68]; not only having the seal of level rating [s2,s24,s68]; market recognition [s1]; getting new contracts, especially in the government sector [s16] and in outsourcing [s2]; strengthening of project management [s73]; reducing costs on extra hours and rework [s70]; maintaining the level of quality assurance [s66]. To avoid extending this paper, details about the studies that make up each group will not be given here.

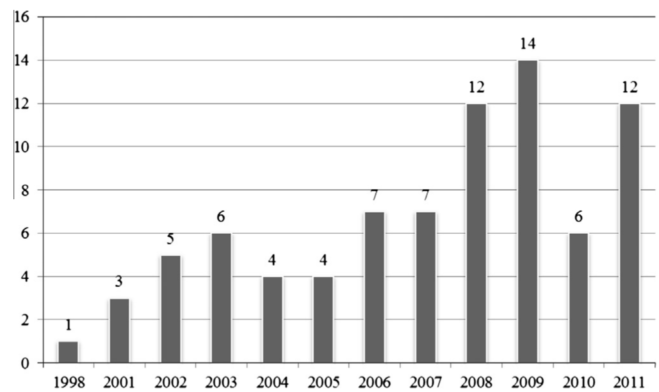


**Fig. 6.** Published studies by year.

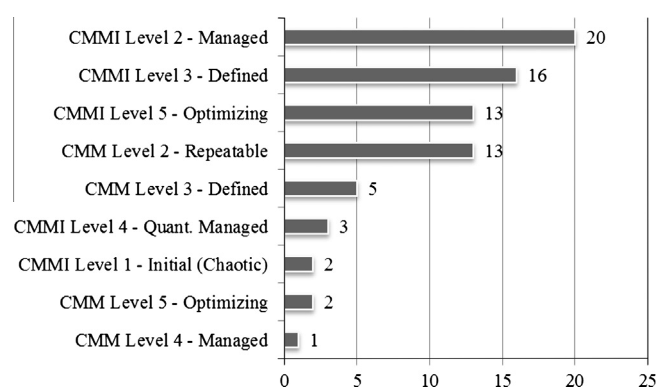


**Fig. 7.** CMM/CMMI Levels addressed in the studies.

### 4.4. General characteristics

Fig. 4 gives an overview of the studies by publication channel. *Agile Conference* had the largest number of studies (12 in total). Other conferences such as the *International Conference on Product Focused Software Process Improvement (PROFES)*, *XP/Agile Universe*, the *International Conference on Agile Software Development: XP* and the *International Conference on Software Engineering (ICSE)* had at least two studies. The journals *CrossTalk The Journal of Defense Software Engineering* and *IEEE Software* had respectively six and three studies. Fifty-eight studies (72%) were published in conferences, while twenty-one (26%) were published in journals and two (2%) published in books belonging to the Springer series. It is interesting that many studies (23%) came from proceedings of conferences about agile development. This could signal that the agile community sees these two approaches (CMMI and agile) as complementary to each other not as opposites. Appendix D has the full distribution of studies by publication channel and type.

The studies came from various countries. Fig. 5 shows the distribution of the studies by country. The country of origin of the study was considered to be the author's institution, and the country in which the research was conducted. Most studies were conducted in the country of origin of the author's institution. However, eight studies involved projects in more than one country. Twenty-three studies (28%), the largest number, were conducted in the United States (USA), probably because it was the cradle of both approaches. There were six studies (7%) in Finland, five studies (6%) in Netherlands and Sweden, and four studies (5%) in Brazil.

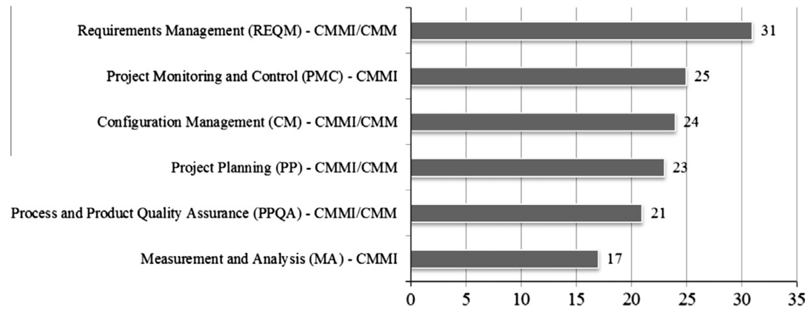


**Fig. 8.** CMM/CMMI process areas most cited in the studies.

Only one study was conducted in 15 countries, namely, Germany; Saudi Arabia; Bulgaria; Canada; Singapore; Chile; China; Colombia; Korea; Croatia; Denmark; Estonia; Philippines; Ireland; and Pakistan. Studies involving more than one country were carried out in: Denmark and the USA (3 studies); Denmark, Finland, USA and the United Kingdom (1); England and Germany (1); Ireland and Finland (1); Jordan and the United Arab Emirates (1); Sweden, Northern America, North-Eastern Europe and Eastern Europe (1).

Regarding the year of publication of the studies, no studies were found on CMM/CMMI and agile methodologies before 1998, the first year in which one study was published. Then there was a jump to 2001 and onwards before other studies were published. Fig. 6 shows the number of studies published in each year.

Note that, in general, the trend was for the number of published studies to increase. However, the number of published studies fell in 2004 and did not recover in 2005 and there was a sharp fall in 2010 followed by a recovery to the 2009 total in 2011. A cause for variations in the number of published studies was not identified. A CMMI report [2] recorded that the number of CMMI appraisals also decreased in 2010, in relation to earlier years. 2010 saw the release of a new version of CMMI, but at first it was not possible to establish a relationship between this and the decrease in the number of studies published in that year. 2009 was the year that most studies on the subject were published.

### 4.5. CMM/CMMI levels and process area

Although the term CMMI has been used in the general scope in this review, data on CMM and CMMI levels were treated separately, according to the version referenced in the studies. Most of them referred to CMMI Level 2, which was cited in 20 studies, as can be seen in Fig. 7. Next, quoted in 16 studies, is CMMI Level 3. CMMI Level 5 tied third with CMM Level 2, both being cited in 13 studies. An interest in demonstrating that agile methodologies are compatible with the highest level of CMMI was noted in the studies. However, the same did not happen with CMMI Level 4, which was cited in only 3 studies, behind CMM Level 3, which was quoted in 5 studies. A possible cause for this could be the fact that companies prefer to implement levels 4 and 5 together. Few companies implement only level 4 [2], so level 5 is the most cited. Two studies mentioned CMMI Level 1 and CMM Level 5. CMM Level 4 was cited in only one study. Many studies referred to more than one level of CMMI and CMM.

To complement the information presented in Fig. 7, 18 studies referred to CMMI in general, and did not describe the levels properly. The same occurred with nine studies that referred to CMM in general. Two studies included as an approach the People CMM version and another study addresses SSE-CMM (the Systems Security Engineering Capability Maturity Model). All the others focused on SW-CMM (Software CMM) or CMMI-DEV (CMMI for Development).

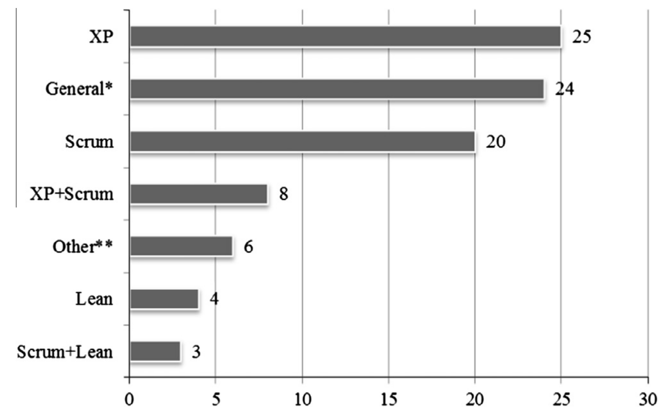


**Fig. 9.** Agile methodologies most cited in the studies. * "General" refers to studies about agile practices, which do not define a specific methodology. ** "Other" refers to company internal agile methodology.

The CMMI process area most cited in studies was Project Planning (cited in 28 studies), followed by Project Monitoring and Control (25 studies), Requirements Management (23 studies), Configuration Management (18 studies), Measurement and Analysis (17 studies) and Process and Product Quality Assurance (16 studies). Altogether, 21 CMMI process areas were cited. The areas related to planning, monitoring and requirements management may have been most cited because these issues are also addressed by Scrum, and it is the agile methodology currently most used according to [6] and one of the most cited in this review. We can also see that the three Basic Support Process Areas (Configuration Management, Measurement and Analysis, and Process and Product Quality Assurance), which address fundamental support functions for other process areas [1], were the most cited. From CMM the following areas were cited, with the number of studies that cited them in brackets: Requirements Management (8); Project Tracking and Oversight (7); Configuration Management (6); Project Planning (5); Quality Assurance (5); Project Management (3). By way of counting, for studies that describes generally involve all areas of a certain level of CMM/CMMI each area of the referred level was considered. For studies that reported having used broadly all areas of CMM/CMMI counting was considered for the category "All Areas". Four studies reported used all areas of CMMI, while three studies reported used all areas of CMM. Fig. 8 summarizes the areas cited by combining similar areas of CMM/CMMI.

### 4.6. Agile methodologies and practices

As to the agile methodologies applied in the studies included, Fig. 9 shows that Extreme Programming (XP) was the most mentioned, being used in 25 studies. Twenty-four studies cited agility in "General", i.e. applying agile practices, without defining a

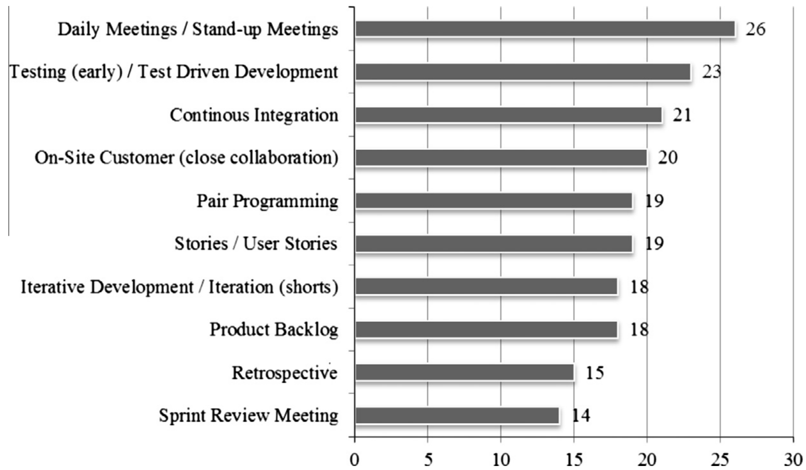


**Fig. 10.** Agile practices most cited in the studies.

specific methodology. Scrum was cited in 20 studies, while eight studies reported the use of XP and Scrum together (Scrum + XP). Specifically agile methodology created internally in the context of projects described in the studies, nominated as "Other", were mentioned in six studies. Lean was reported in four studies, and in combination with Scrum (Scrum + Lean) was cited in three studies.

The following methodologies were mentioned only once in the studies: Agile RUP, Crystal Clear, and DSDM; besides agile combinations of Lean + Scrum + TDD, Scrum + XP + Other, and XP + Scrum + Lean; and combinations with traditional methodologies Agile General + RUP and Scrum + PMBOK. Thirteen studies cited two or more agile methodologies; they refer to more than one project. Recent methodologies like Kanban [29] and others were not cited. Although they were not included explicitly in the search string, generic terms like agile, agility and lightweight could have been found these methodologies. Kanban was to a certain extent considered since it is part of the Lean Software Development, but no study mentions it explicitly.

We attempted to pinpoint the agile practices discussed in the studies. Fig. 10 highlights the ten most cited practices. Daily Meetings or Stand-up Meetings present in XP and Scrum methodologies were the most cited agile practice, appearing in 26 studies, followed by Testing or Test Driven Development, cited in 23 studies, and Continuous Integration, cited in 21 studies. Practices coming from XP such as On Site Customer (cited in 20 studies), Pair Programming (19), User Stories (19) and Iteration Development (18) also appear among the ten most cited practices. Besides these, the other classical practices of XP were identified as the Planning Game (cited in 13 studies), Small Releases (11), Collective Ownership (11), Refactoring (11), Simple Design (10), Coding Standards (10), 40-h Week/Sustainable Pace (8), and Metaphor (6). At least, five studies reported having used all XP practices.

The following practices specific to Scrum were identified, for which the respective number of studies that cited them is given in brackets: Product Backlog (18); Retrospective (15); Sprint Review Meeting (14); Sprint Development (11); Sprint Planning Meeting (11); Burndown Charts (7); Sprint Backlog (6); and Removal of Impediments (6). Two studies reported having used all Scrum practices.

Agile practices in general were identified such as: Tasks (cited in 7 studies); Estimative (7); Continuous Measurement/Monitoring (6); Self Manageable Team (5); Incremental Delivery (5); and Intensive Communication (3). Five studies reported having used agile practices in general, although they do not specify which ones.

Agile practices related to other methodologies such as Lean (Remove Waste), FDD (Domain Object Modeling, Inspections, Feature Teams and Progress Reporting) and Crystal (Staging, Revision and Control, Methodology Tuning Technique and Information Radiators) were cited in at least one of the studies included. In ten studies it was not possible to identify agile practices used.

## 5. Discussion

To establish a discussion that seeks to answer the research questions defined for this review, both type of studies (empirical and non-empirical) are considered. This decision is motivated by the fact that, although experience reports do not have scientific rigor, they can help identify benefits and limitations in the context of CMM/CMMI and agile methodologies (Section 5.1), in order to answer the First Research Question (RQ1) of this review. Therefore, the reliability of the information contained in these studies is discussed (Section 5.2) in answer to the Second Research Question (RQ2). And the implications for research and practice are set out (Section 5.3) in response to the Third Research Question (RQ3). Finally, Section 5.4 discusses the limitations of this review.

### *5.1. Benefits and limitations*

The benefits and limitations of using CMMI together with agile are discussed in the sub-sections below. In order to facilitate the synthesis, the benefits and limitations were grouped into two main categories: those related to the *organization* in general and those associated with the *development process*, which were organized into the following subcategories: *process understanding or knowledge*; *communication*; *management*; *configuration*; *requirements*; *tests*; *maturity*; *productivity*; and *quality*. We tried to keep the same nomenclature that the studies included used for all the benefits and limitations that they cited. Some limitations were originally deemed as challenges by studies, but in this review the term limitations was used generally to refer to both limitations and challenges.

#### *5.1.1. Organization*

The benefits related to the *organization* reported on were: business objectives achieved successfully [s2,s20]; organizational growth [s1]; improvement of business performance [s25] and of the company's position in the market [s47]; stability in long-term goals [s47]; and better relationships with suppliers [s13]. Improvements in the work environment of the organization are reported in [s11,s51], and improvements added to the client with respect to satisfaction, value and achievement for their needs in [s40,s15,s30,s17,s55,s71]. The benefits extend to the development team and employees, and these were deemed as providing: satisfaction [s15,s52]; top management support [s18]; and integrated work across teams, departments and customers [s69,s13,s8,s22]. Five studies reported organizational benefits, related to cost and price reduction [s24,s12,s10,s5,s11].

As regards the *organization*, some limitations conflict with the benefits found. Studies [s30,s13] reported that disagreements occur in the team, introducing new members is difficult and stress increases. Other studies [s30,s69,s12] pointed out as challenges, related to the team, the following issues: offshoring with a centralized team; replacing team members; inadequate infrastructure; team behavior; and culture of the sequential development. To [s79,s18], resistance to change is a factor of difficulty when implementing CMMI and agile methodologies. Factors such as all stakeholders reaching a consensus [s76], promoting training for new members [s1] and involving senior management [s24], present challenges for deployment. The limitations on CMMI contain no practice of or guidance on business goals [s17] and establishing effective collaboration with the client [s80,s22,s68,s12,s1,s15] conflict with the benefits found. Difficulties related to performance at the organizational level were reported in several studies [s19,s7,s8,s9,s68,s31]. These difficulties are complementary to the issue of which items of information are concentrated in teams on specific projects, it being difficult to outsource this experience at an organizational level [s25,s9,s44]. On the organizational level, the following challenges still remain: organizational learning [s15,s39,s11]; developing control while maintaining agility [s73,s2,s8]; little emphasis on process, documentation, contracts and planning [s80]. In terms of costs, [s2] pointed out that the training costs are more than expected and [s47] that the process of implementing CMMI was time consuming.

The disagreements among the studies do not allow strong affirmations to be made that reflect the views of a large majority, but we can note that for the clients and the organization, in terms of market gain and recognition, the adoption of CMMI and agile provided more benefits than limitations. Within the context of team activity, but arguably less clearly so at the level of the organization as a whole, the benefits also were present. The limitations impact the aspect of externalizing the experience from the team to the organization as a whole. Some cultural and social aspects also represent challenges to be overcome in the teams and in the organization.

#### 5.1.2. Process understanding and communication

A better *understanding* of the *development process*, promoting its visibility and transparency, were deemed as benefits in [s16,s79,s4,s15]. The fact that the process became more understandable was highlighted, as were facilitating in-service training, team learning, the introduction of new members, the evaluation of new methods and tools, mentoring and conflict resolution [s79,s13,s15,s51,s16,s30]. According to [s42,s46,s43], the use of CMMI and agile methodologies contributes to institutionalizing the process, defining an optimized framework and improving the process of designing artifacts. Some other benefits discussed were: collaboration with stakeholders [s68,s73] and the strong inclusion of the client, with regard to identifying and improving relationship problems [s58,s60].

On the limitations related to the *process* with regard to *understanding* it, the difficulty of generalizing the results was reported, since the way that agile practices are associated with process areas may vary according to the context/domain [s4,s65]. Difficulties related to mapping, which provides full adjustment, and to guidance on how to take advantage of the best agile practices and to find resources/responsibilities for process improvement are highlighted in [s11,s61,s76,s2]. The results vary because of human factors and extremism [s72,s20,s80]. Support for development in specific technical environments, such as emphasis on hardware [s63], medical [s1] and open source software [s15], was highlighted as a limitation. The lack of dealing with dependencies was discussed as a challenge in [s15,s23].

As to understanding the process, the limitations contradict the benefits mainly in relation to the absence of a general solution that fits all cases. Human factors and resistance to change may lead to limitations. Some specific technical areas and external dependencies may represent challenges. However, the benefits point towards a more viable process in order to meet the organization's expectations, client satisfaction and team needs.

The *communication* and interaction between the team, in the context of the process, were listed as benefits by studies [s4,s16,s63,s71], and stressed that using a common language and interaction increased. Studies [s69,s22] reported the benefit of an increase in feedback. Difficulties in *communication* were reported in five studies [s63,s73,s69,s33,s54]. However, [s63] pointed out that the limitation occurred with stakeholders still unfamiliar with agile development. [s73] refers to a globally distributed project and [s54] to a cross-functional team, situations where communication difficulties of course arise. The training material is cited by [s33] as a challenge related to communication. Training is a recurrent aspect in other studies, which indicates that the organization seeking to combine CMMI and agile must invest in training.

#### 5.1.3. Management and configuration

Benefits related to *management* emphasize that the project is monitored as a whole and daily [s46,s73], thereby enhancing project management in situations of great dynamism [s50]. Limitations, on the other hand, highlighted issues about the role of management, such as little attention given to reviews and meetings [s15], little support during the early stages [s63], little familiarity with agile practices [s69] and an excessively large view of the risks associated with adopting agility [s81]. This indicates that CMMI plus agile requires the project manager to engage fully with and to be open to agile practices. Others challenges were: not providing visibility to management nonconformities [s24]; lack of integrated project [s23]; need to manage artifacts other than code [s34]; and difficulties with the management components interface [s68].

For *configuration*, as a benefit, the overheads incurred on configuration management are reduced [s15]. There were no limitations associated with this subcategory. However, the CM process area was one of the areas deemed to have little coverage [s73,s77,s30,s24].

#### 5.1.4. Requirements and tests

Several benefits, described in [s18,s36,s60,s63,s69,s77,s22], indicated improvements in aspects of *requirements*, such as elicitation and specification, change management, change in priorities, identification of stakeholders, managing client demands, documentation and evaluation. Limitations associated with *requirements* pointed out the difficulty in managing these [s16], and difficulties in: dealing with their histories and estimates/management [s16,s69], generating the traceability matrix [s45], managing change [s79], performing analysis [s79], documenting requirements/changes [s32,s69] and dealing better with secure software requirements [s57]. Some limitations conflict with the benefits, in one sense, the agile manner in dealing with requirements lead to improvement, but the lack of knowledge about some agile particularities, such as user stories and estimates, could lead to limitations.

In *tests*, [s63] describes improvement in unit tests as a benefit. According to [s21], there was difficulty with the agile practice of building *tests* first. Nor did the creation of constant testing have a positive performance in [s63]. For [s21], it is necessary to increase the number of automated tests. The lack of documentation of tests was reported in [s76]. Test-first could represent a difficulty, as pointed out in other studies, which deal only with agile methodologies [16], and when combining agile with CMMI the difficulties remain. However, when agile brings testers and clients closer together with the development team, in order to perform tests, this generates improvements in the process.

### 5.1.5. Maturity

The very progress in process *maturity* and achievement of level rating in the Standard CMMI Appraisal Method for Process Improvement (SCAMPI), such as level 2, or better performance in specific process areas, were cited as benefits by [s2,s3,s34,s13]. Associated with maturity, an increase in the flexibility and adaptability of the process [s18,s55] and process weight reduction [s64] were also indicated. Some studies indicated benefits from process improvement [s34,s52,s17,s28], the results from this [s21], choosing the optimal process in accordance with the project [s26] and evaluating this [s37]. Improvements related to planning [s63], response to changes [s58], task prioritization [s8], the accuracy of estimates [s79], predictability of activities [s43], consolidated progress [s43], measurements [s46], using a systematic and quantitative approach [s16] and real performance data [s22] are pointed out. For [s56,s31], the focus on the best software practices is maintained.

However, the category *maturity*, related to CMMI level rating, had most associated limitations. The appraiser's dependence with respect to his/her understanding of agile methodologies was highlighted in two studies [s4,s3]. The need for additional practice was mentioned by four studies [s11,s29,s81,s50]. The need for practices of other methodologies (e.g. RUP and XP) to complement Scrum was pointed in [s23,s61]. Seven studies [s62,s30,s19,s71,s47,s11,s13] reported conflicts at higher levels (levels 4 and 5), while studies [s77,s7,s29,s8] reported little coverage of process areas: REQM; MA; PPQA; CM; SAM; RM; DAR; OPF; QPM; ISM; CAR; PMC. The areas PPQA, RSKM and CM also were quoted in [s42,s23,s24], respectively. The SAM area was most mentioned (3 studies), followed by PPQA and REQM (or RM), cited by 2 studies. Other studies [s10,s23,s8] pointed out limitations of the coverage of process areas, product acquisition and management process. [s5] warns that if one process area is omitted, this could imply that the company cannot claim maturity level 2, for example.

For [s11,s80], there is a difficulty in adopting values and practices of agile methodologies. Limitations related to measurement and quantitative metrics were highlighted by [s15,s18,s22,s43]. Making estimates was seen as a problem by [s63,s25,s23,s26]. Limitations associated with documentation were identified by several studies [s12,s63,s18,s31]. They were concentrated mostly in the category of maturity, but some limitations were classified in other categories such as requirements [s32] and testing [s76], indicating that limitations related to documentation permeate the development process in its various stages. In this aspect, data management [s45], availability in electronic format [s1], support of automated tools [s50,s62], absence of a historical basis [s23], lack of planning and control of data [s23] and the effort to generate evidence [s61] were also identified as limitations. The question of producing documentary evidence is taken up by [s6,s79,s34,s53,s80]. For [s75,s60], the limitation is to maintain agile practices during the project, mainly in response to unpredictable changes. Agile practices are criticized in [s30,s43,s38,s9] due to the lack of engineering aspects. Divergences in practices such as refactoring and retrospectives, lack of global vision, lack of learning techniques, restricted domain and difficulty of establishing a software engineering process group (SEPG), complement the limitations associated with maturity [s38,s39,s58,s47].

With regard to process maturity, while benefits indicate that agile empowers aspects such as flexibility, process improvement, response to change, that is helpful for level rating, the limitations lead to aspects such as measurement and documentation, that reinforce the sense that agile on its own is not enough to obtain all CMMI levels.

### 5.1.6. Productivity and quality

*Productivity* was a prominent factor in several benefits identified. According to [s11,s65,s74,s10], there was an increase in productivity and performance. For [s54], productivity doubled, while [s55] had the milestones delivered on or before the deadline. The release time, rework, effort and interruptions decreased, according to [s18,s55,s11,s43]. Only two studies reported challenges related to *productivity*. One reported the feeling of having more work [s18] and the other the need to improve the ability to deliver to the market with greater frequency [s52]. Thus, a positive aspect of combining CMMI and agile is that overall productivity is improved as evidenced by the larger number of benefits that were reported in the studies.

Benefits related to *quality* highlighted the ease with which defects and deviations were identified, corrected and reduced [s15,s55,s20,s11,s18,s8]. Management and reduction of risk were listed in [s46,s69]. Other benefits focused on analyzing the improvement of quality [s55,s46,s13,s14,s47,s51]. Limitations associated with the *quality* approach showed: lack of focus on SPI [s22]; absence of support for quality assurance [s24]; difficulty in meeting audit requirements [s81]; weakness in dealing with the PPQA area [s42]; difficulties in the relationship between the quality team and the agile team [s76]; the need for a group of quality assurance [s24]; achieving a balance between quality guidelines and reliability goals [s52]; and in adopting Lean [s41]. As regards to risk, limitations indicate little risk management service [s8,s23,s24,s46]. The need to support secure software development was identified [s57]. Despite the limitations, some studies [s2,s4,s15,s24,s47,s56,s78] discuss how quality assurance could be conducted in environments with XP or Scrum, and that these are very helpful to overcome these difficulties.

## 5.2. Strength of evidence

To define the strength of evidence Dybå and Dingsøyr [16] used the GRADE (Grading of Recommendations Assessment, Development and Evaluation) system, which classifies the strength of evidence as *high*, *moderate*, *low* and *very low*, based on evaluating four elements: study *design*; *quality* of the study; *consistency*; and *objectivity* [30].

Regarding the *design* of studies, the biggest factor in reducing the strength of evidence is the absence of empirical data, clearly detected in experience reports and in theoretical studies. Among the empirical studies, most are observational studies (case studies and action research), which have a *low* rating, according to the GRADE system, while the evidence from experience reports has *very low* ratings.

As to *quality*, the following issues imply a negative effect on the strength of evidence: the absence of assessment of proposals; superficial approaches, with no details about the corresponding areas of CMMI and agile practices; informal methods of data collection; no presentation of how the data were collected or details about the sample; very short studies, resembling posters and expert opinion. On the other hand, the limitations of research were addressed in 11 studies. This discussion contributes to increasing the strength of evidence, since it allows scaling challenges for future research. The following limitations were cited: not presenting improvements after the method was implemented [s14]; not detailing the merger between CMMI OPF and agile practices [s28]; study author involved in the development team [s39]; failure to provide concrete data on CMMI [s39]; results based only on one software project [s44]; small number of business respondents in the sample [s48]; study based only on quantitative data [s59]; the use of a single-case study [s59]; conclusions based on observations and informal discussions [s62]; limited amount of data collected and related to small organizations [s72]; limited number of team members involved in the research [s75]; context-specific results [s79]. One factor that reduced the limitations of [s75] was using actual documentation of the project as a source of information. The collection of data in just one organization was also reported as a limitation in [s60].

The quality issue was the subject of further analysis in Section 4.1. According to this assessment, the overall rating of the quality of the studies was *very low* to *low*, when considering the quality assessment on the following scale: 0–2 criteria, very low; 3–5 criteria, low; 6–8 criteria, moderate; 9–11 criteria, high. In the case of experience reports (38 studies) the quality was *very low* to *low*, but consideration must be given to the fact that the experience reports were produced mostly by professionals working in the industry, whose perceptions stem from everyday experience and collaborate with the strength of the information highlighted. In theoretical studies (20 studies) the quality assessment was *very low* to *low*. However, the quality of the empirical studies (23 studies) was *moderate* to *high*.

The disagreements between the benefits and limitations identified in aspects such as team satisfaction, requirements and other, reflect negatively on the *consistency* of the studies. However, in other categories the benefits show different aspects from limitations, and this contribute to increasing consistency.

With regard to *objectivity*, three studies were considered as not directly related to CMMI and agile methodologies or having a rather superficial approach. DSDM methodology is mentioned in one study, but it does not give details of the practices and areas. Some studies focus more on agile practices, and do not address CMMI areas. Other studies focus primarily on other approaches, e.g.: ISO; Global Development; PMBOK; RUP; and Lean. Two studies do not clearly explain agile methodologies.

In this review, it is possible that the GRADE system is not adequate, especially because the review included experience reports and theoretical studies. However, according to elements provided in GRADE (design, quality, consistency and objectivity), discussed above, it is considered that the strength of evidence found is *low*, indicating that further research is very likely to have an important impact on confidence in the estimation of effects (results of the application of CMMI and agile methodologies) and is likely to change the estimate. As this is a preliminary survey on the topic, the evidence found can serve as a starting point for insights.

### 5.3. Implications for research and practice

Sixty-three studies were considered to have value for research or practice, as per the quality assessment in Section 4.1. Among them: thirty studies were considered to have a direct implication for *research*, nine of which were empirical studies (5 surveys, 2 single-case studies and 2 multi-case studies), five were experience reports and sixteen theoretical studies; twenty-four studies were considered to be applicable in *practice*, of which five were empirical (4 single-case studies and 1 multi-case study) and nineteen experience reports; and nine studies were considered relevant for both research and practice, all of which were empirical studies (1 action research, 3 single-case studies, 3 multi-case studies, 1 survey and 1 mixed). The value of the research was not clear for quality assessment in eighteen studies.

This review demonstrates the scarcity of empirical studies in the area of software process improvement, even though there are consolidated models in the industry, such as CMMI, when it is considered that only 28% of the studies included are considered empirical. Another area for future research is related to higher levels of CMMI. The application of agile practices for rating at the highest levels is interesting for industry. A research study using a combination of two or more agile methodologies, such as happens in practice in some companies, would be welcome.

Some CMMI areas were taken, by studies included, with low coverage or with difficulties in complying with agile methodologies, e.g. SAM, PPQA, MA and others. Future research can contribute by proposing solutions that add established agile practices to meet the objectives of these areas.

The fact of strength from identified evidence being assessed as low reinforces the importance that in future research, studies should seek to be linked to criteria for better performance in quality assessment, especially as regards researchers clearly describing the design of the research, selecting the sample, using a control group, collecting data, analysis and reflexivity on possible influences on the results.

From the industry point of view, despite the need for additional practices, among other limitations detected in specific fields, agile methodologies helped companies attain, in a less bureaucratic way, rating at CMMI levels 2 and 3. Registers include reports on adopting agile practices in companies with level 5. Thus, companies and professionals interested in improving the software process and obtaining a CMMI level rating should consider applying agile methodologies in their contexts. The data presented in this review, by providing an overview of research on the subject, can be assessed by companies for the purpose of identifying similarities and differences between the results reported by the studies and their own situation.

In this scenario, the proposal of models and guidelines to guide the application of agile values and practices for software process improvement with CMMI also represents an opportunity for future studies. Is important that these proposals are prepared in conjunction with professionals and companies in order to effectively meet software industry needs.

### 5.4. Limitations of this review

The main limitation of this review is the possibility of bias. However, it was supervised by other researchers whose knowledge of the subject is broad. Two reviewers assessed each criterion in quality assessment independently. When they disagreed about the assessment, a third reviewer was called on to resolve the conflict. Disagreements about the inclusion/exclusion of a particular study and data extraction were resolved by referring to the original study and discussing it to establish a consensus. When necessary, the authors of the studies were contacted for additional information. We also seek to minimize the possibility of bias by defining a protocol and by constantly revising the spreadsheets containing the data at each stage of the review (study selection, quality assessment, data extraction and synthesis).

It is possible that relevant studies were not included. However, we have tried to minimize this possibility by having also made manual and automatic searches in the indexation of major studies. The formulation of the search string took into account the terms used in previous revisions both in the SPI as in agile methodologies, in order to make this fit the scope. Additionally, we checked the reference lists of previously studies included to identify relevant studies not found in searches. This technique was positive, since 7 studies were detected as potentially relevant, of which 5 were included. No studies published after 31 December 2011 were considered, since the searches were consolidated in 2012.

The proposal to map published studies on CMMI and agile development, in order to carry out an overview of research on the subject, may have contributed to the inclusion of studies that made secondary reference to the topic, yet these focused on software process improvement. The data from these studies, such as benefits and limitations identified, had the same treatment with respect to relevance as the data from studies that focused on CMMI and agile methodologies in their essence. No distinction was made of the relevance in the treatment of data from studies with different performances in quality assessment. Results can suffer variations when a critical analysis of them is made that takes into account treating studies differently according to their quality.

## 6. Conclusion and future work

This systematic literature review initially identified 3193 studies, of which 445 were considered potentially relevant and 81 studies were included on the use of CMMI in combination with agile software development. The studies included were evaluated according to quality criteria and data and results from them were extracted and characterized. The number of studies included and the fact that the number of published studies is growing indicate that discussion and research on this theme is relevant and current. The results enabled benefits and limitations related to the use of CMMI and agile methodologies to be identified.

Agile methodologies have been used by companies to reduce their efforts to reach levels 2 and 3 of CMMI, there even being reports of applying agile practices to obtain level 5. Among the benefits, improvements in organizational aspect, greater team and customer satisfaction, further integration, cost reduction, process assimilation, increasing productivity and reducing defects stand out. The feasibility of using CMMI together with the agile development is manifested on both sides. From the CMMI, the latest release of the CMMI-DEV Technical Report [1] incorporates a set of suggestions for applying the model in environments with agile methodologies and these suggestions were also considered in its appraisal method (SCAMPI). Depending on the lead appraiser, evidence of statements, for example, can replace the need for a direct artifact. This is a breakthrough in the appraisal method and favors the adoption of agile methodologies without the need to create (or generate) direct and objective evidence, although it is still subject to a relative dependence on the appraiser. On the agile side, the large number of papers published in events on agile development, also signals that the agile community is open to seeking rapprochement with CMMI.

However, agile methodologies alone, according to the studies, were not sufficient to obtain the level desired, it being necessary to resort to additional practices. Organizations interested in combining CMMI and agile development should not neglect the documentation and evidence required by the formal model of process improvement, even if this action demands solutions not available in the market, especially when it comes to automation, quality assurance and metrics in line with agile values. Documentation should demonstrate that the process is followed, document decisions, and document specific activities, e.g. database [s6,s38,s44]. Organizational training and learning also poses a challenge, mainly for organizations that have no prior knowledge about agile, while seeking consultancy services from professionals and companies with recognized expertise in CMMI and agile methodologies contributes to this adoption occurring smoothly. The definition of a software engineering process group (SEPG) could also help, but it must be made up of process executors, and not be an isolated or outside group [s17,s60].

Organizations should seek to ensure that how CMMI and agile can be combined is understood and undertaken by those involved, whether these be senior management, project managers, development teams, customers or evaluators [s22,s63]. Training, process visibility (through face-to-face communication, project web sites, wikis, forums, guides, manual or cookbooks), and tools for automation of complex activities or quantitative attributes, were highlighted as success factors by studies [s2,s18,s28,s44,s49,s56,s66,s69]. Metrics should add improvements to the agile process and assist in the improvement or correction of future results of the project, it being necessary to focus critically on the analysis, reporting and improvement actions [s16,s20]. The combination of different agile methodologies, like XP for operational/engineering, Scrum for management, and Lean for tactical/strategic, seems to be a way forward [s69,s78]. The solutions implemented in the organization should not be disconnected from the teams. They should emerge in the team context, and give special attention to the values of the Agile Manifesto (individuals and interactions, working software, customer collaboration and responding to change) [3], so that the teams define the best solution, by combining both approaches i.e. maturity and agility. As highlighted before, there is no generic solution that fits all cases. The teams and the organization as a whole, based on their own goals, should focus on two basic aspects: process quality and product quality. Process quality is manifested through the satisfaction of the team and the organization with the practices used for software development, like sustainable place, self-organization and return on investment. Product quality, for its part, manifests itself through customer satisfaction with the achievement of the desired (but sometimes not clearly stated) functional and non-functional software requirements.

As a contribution in relation to the results obtained in this review, an agile maturity model for software development organizations has been defined, as per [31]. The model guides the setting up and running of agile methodologies, based on Capability Maturity Model Integration (CMMI), into the five levels of maturity. Based on the results of this review, we are proposing as current work a set of strategies and approaches so as to combine CMMI and agile in some specific areas (including the most cited here), considered strategic for many organizations, such as project management, quality assurance and, in a new field, user experience design.

Due to the low strength of evidence, we cannot say the extent to which combining CMMI and agile methodologies is feasible so as to foster improvements in the software industry. This question requires further research. A smaller number of empirical studies (23), compared with the number of experience reports (38) found, suggests investment in future empirical research to be developed together with companies. We consider the software industry as a *locus* in which the interaction between CMMI and agile methodologies occurs, and where its benefits and limitations emerge. Greater in-depth analysis of specific process areas of CMMI, especially at the highest levels, and defining proposals and guidelines to assist in combining agile practices and values in serving these areas, will contribute to expanding benefits to the industry, this being an open field for future research.

## Acknowledgements

We would like to thank the many authors of studies who, in response to our requests made via e-mail, promptly sent their studies, so that they could be analyzed in the context of this review.

This work was supported by the Foundation for Science and Technology of the State of Pernambuco (FACEPE), under Process No. IBPG-0136-1.03/11.

## Appendix A. Studies included in the review

[s1] C. Vriens, R. Barto, 7 Years of Agile Management, in: Proceedings of the Agile Conference 2008 (AGILE'08), IEEE, Toronto, Canada, August 4–8, 2008, pp. 390–394. doi:10.1109/Agile.2008.97.

[s2] B. Shen, T. Ruan, A Case Study of Software Process Improvement in a Chinese Small Company, in: Proceedings of the International Conference on Computer Science and Software Engineering 2008, IEEE, Wuhan, China, December 12–14, 2008, vol.2, pp. 609–612. doi:10.1109/CSSE.2008.701.

[s3] N. Porrawatpreyakorn, G. Quirchmayr, W. Chutimaskul, A Prototype for the Support of Integrated Software Process Development and Improvement, in: Proceedings of the 4th International Conference on Advances in Information Technology (IAIT'10), Springer, Bangkok, Thailand, November 4–5, 2010, pp. 94–105. doi:10.1007/978–3-642–16699-0_11.

[s4] T. Kähkönen, P. Abrahamsson, Achieving CMMI Level 2 with Enhanced Extreme Programming Approach, in: Proceedings of the 5th International Conference on Product Focused Software Process Improvement (PROFES'04), Springer, Kansai Science City, Japan, April 5–8, 2004, pp. 378–392. doi:10.1007/978–3-540–24659-6_27.

[s5] F. Mc Caffery, P.S. Taylor, G. Coleman, Adept: A Unified Assessment Method for Small Software Companies, IEEE Software 24 (1) (2007) 24–31. doi:10.1109/MS.2007.3.

[s6] Collabnet, Agile CMMI at a Large Investment Bank, Agile Journal (2009), http://www.agilejournal.com/whitepapers/2492-agile-cmmi-at-a-large-investment-bank, last accessed in February 2012.

[s7] Omran, AGILE CMMI from SMEs perspective, in: Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA'08), Springer, Damascus, Syria, April 7–11, 2008, pp.1–8. doi:10.1109/ICTTA.2008.4530352.

[s8] R. Turner, Agile Development: Good Process or Bad Attitude?, in: Proceedings of the 4th International Conference on Product Focused Software Process Improvement (PROFES'02), Springer, Rovaniemi, Finland, December 9–11, 2002, pp. 134–144. doi:10.1007/3–540-36209–6_13.

[s9] R. Turner, A. Jain, Agile Meets CMMI: Culture Clash or Common Cause?, in: Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods (XP/Agile Universe 2002), Springer, Chicago, USA, August 4–7, 2002, pp. 153–165. doi:10.1007/3–540-45672–4_15.

[s10] M.I. Khan, M.A. Qureshi, Q. Abbas, Agile methodology in software development (SMEs) of Pakistan software industry for successful software projects (CMM framework), in: Proceedings of the 2010 International Conference on Educational and Network Technology (ICENT'10), IEEE, Qinhuangdao, China, June 25–27, 2010, pp. 576–580. doi:10.1109/ICENT.2010.5532104.

[s11] C. Santana, C. Gusmão, L. Soares, C. Pinheiro, T. Maciel, A. Vasconcelos, A. Rouiller, Agile Software Development and CMMI: What we do not know about dancing with Elephants, in: Proceedings of the 10th International Conference on Extreme Programming (XP 2009), Springer, Pula, Italy, May 25–29, 2009, pp. 124–129. doi:10.1007/978–3-642–01853-4_15.

[s12] N.P. Singh, R. Soni, Agile Software: Ensuring Quality Assurance and Processes, in: Proceedings of the International Conference on High Performance Architecture and Grid Computing (HPAGC'11), Chandigarh, India, July 19–20, 2011, pp. 640–648. doi:10.1007/978–3-642–22577-2_86.

[s13] R. Leithiser, D. Hamilton, Agile versus CMMI - process template selection and integration with Microsoft Team Foundation Server, in: Proceedings of the 46th Annual Southeast Regional Conference on XX (ACM-SE'08), ACM, Auburn, USA, March 28–29, 2008 pp. 186–191. DOI: 10.1145/1593105.1593153.

[s14] F. McCaffery, M. Pikkarainen, I. Richardson, Ahaa - agile, hybrid assessment method for automotive, safety critical SMEs, in: Proceedings of the 30th International Conference on Software Engineering (ICSE'08), ACM, Leipzig, Germany, May 10–18, 2008, pp. 551–560. doi:10.1145/1368088.1368164.

[s15] E. Bos, C. Vriens, An Agile CMM, in: Proceedings of the 4th Conference on Extreme Programming and Agile Methods (XP/Agile Universe 2004), Springer, Calgary, Canada, August 15–18, 2004, pp. 129–138. doi:10.1007/978–3-540–27777-4_13.

[s16] S. Cohan, H. Glazer, An Agile Development Team's Quest for CMMI® Maturity Level 5, in: Proceedings of the Agile Conference 2009 (AGILE'09), IEEE, Chicago, USA, August 24–28, 2009, pp. 201–206. doi:10.1109/AGILE.2009.24.

[s17] J.L. Dutton, An integrated framework for performance excellence, CrossTalk 23 (1–2) (2010) 6–9.

[s18] M.M. Trujillo, H. Oktaba, F.J. Pino, M.J. Orozco, Applying agile and lean practices in a software development project into a CMMI organization, in: Proceedings of the 12th International Conference on Product Focused Software Process Improvement (PROFES'11), Springer, Torre Canne, Italy, June 20–22, 2011, pp. 17–29. doi:10.1007/978–3-642–21843-9_4.

[s19] S. Suwanya, W. Kurutach, Applying Agility Framework in Small and Medium Enterprises, in: Proceedings of the International Conference on Advanced Software Engineering and Its Applications (ASEA'09), Springer, Jeju Island, Korea, December 10–12, 2009, pp. 102–110. doi:10.1007/978–3-642–10619-4_13.

[s20] P.E. McMahon, Are the right people measuring the right things? A lean path to achieving business objectives, CrossTalk 21 (5) (2008) 16–20.

[s21] J. Surdu, D.J. Parsons, Army simulation program balances agile and traditional methods with success, CrossTalk 19 (4) (2006) 4–8.

[s22] G. Alleman, Blending agile development methods with CMMI®, Cutter IT Journal 17 (6) (2004) 5–15.

[s23] S.C. Marçal, B.C.C. Freitas, F.S.F. Soares, M.E.S. Furtado, T.M. Maciel, A.D. Belchior, Blending Scrum practices and CMMI project management process areas, Innovations in Systems and Software Engineering 4 (1) (2008) 17–29. doi:10.1007/s11334-007-0040-1.

[s24] Vriens, Certifying for CMM Level 2 and IS09001 with XP@Scrum, in: Proceedings of the Agile Development Conference 2003 (ADC'03), IEEE, June 25–28, 2003, pp.120–124. doi:10.1109/ADC.2003.1231461.

[s25] M. Lepmets, M. Nael, Comparison of Plan-driven and Agile Project Management Approaches: Theoretical Bases for a Case Study in Estonian Software Industry, in: Proceedings of the Ninth International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2010), IOS Press, Amsterdam, The Netherlands, 2011, pp. 296–308. doi:10.3233/978–1-60750–688-1–296.

[s26] Geras, M. Smith, J. Miller, Configuring hybrid agile-traditional software processes, in: Proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2006), Springer, Oulu, Finland, June 17–22, 2006, pp. 104–113. doi:10.1007/11774129_11.

[s27] J.C. Ruiz, Z.B. Osorio, J. Mejia, M. Munoz, A.M. Chavez, B.A. Olivares, Definition of a Hybrid Measurement Process for the Models ISO/IEC 15504-ISO/IEC 12207: 2008 and CMMI Dev 1.3 in SMEs, in: Proceedings of the 2011 IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA'11), IEEE, Cuernavaca, Mexico, November 15–18, 2011, pp. 421–426. doi:10.1109/CERMA.2011.74.

[s28] T. Galinac, Empirical evaluation of selected best practices in implementation of software process improvement, Inf. Softw. Technol. 51 (9) (2009) 1351–1364. doi:10.1016/j.infsof.2009.05.002.

[s29] S.W. Lee, H.-K. Kim, R.Y. Lee, Enterprise Process Model for Extreme Programming with CMMI Framework, in: R. Lee, H.-K. Kim (Eds.), Computer and Information Science, Springer, Berlin, pp. 169–180. doi:10.1007/978–3-540–79187-4_15.

[s30] U. Banerjee, E. Narasimhan, N. Kanakalata, Experience of executing fixed price off-shored agile project, in: Proceedings of the 4th India Software Engineering Conference (ISEC'11), ACM, Thiruvananthapuram, India, February 23–26, 2011, pp. 69–75. doi:10.1145/1953355.1953364.

[s31] M.C. Paulk, Extreme programming from a CMM perspective, IEEE Software 18 (6) (2001) 19–26. doi:10.1109/52.965798.

[s32] J.R. Nawrocki, M. Jasiñski, B. Walter, A. Wojciechowski, Extreme Programming Modified: Embrace Requirements Engineering Practices, in: Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (ICRE'02), IEEE, Washington, USA, September 9–13, 2002, pp. 303–310. doi:10.1109/ICRE.2002.1048543.

[s33] S.W. Baker, Formalizing Agility, Part 2: How an Agile Organization Embraced the CMMI, in: Proceedings of the Agile Conference 2006 (AGILE'06), IEEE, Minneapolis, USA, July 23–28, 2006, pp. 147–154. doi:10.1109/AGILE.2006.30.

[s34] S.W. Baker, Formalizing agility: an agile organization's journey toward CMMI accreditation, in: Proceedings of the Agile Development Conference 2005 (ADC'05), IEEE, July 24–29, 2005, pp. 185–192. doi:10.1109/ADC.2005.27.

[s35] N. Kitiyakara, J. Graves, Growing a Build Management System from Seed, in: Proceedings of the Agile Conference 2007 (AGILE'07), IEEE, Washington, USA, August 13–17, 2007, pp. 401–407. doi:10.1109/AGILE.2007.32.

[s36] J. Babuscio, How the FBI learned to catch Bad Guys one iteration at a time, in: Proceedings of the Agile Conference 2009 (AGILE'09), IEEE, Chicago, USA, August 24–28, 2009, pp. 96–100. doi:10.1109/AGILE.2009.52.

[s37] G. Alkhatib, Z. Maamar, G. Issa, D. Daoud, A. Turani, M.I. Zaroor, Incorporating innovative practices in software engineering education, in: Proceedings of the Global Engineering Education Conference (EDUCON'11), IEEE, Amman, Jordan, April 4–6, 2011, pp.136–141. doi:10.1109/EDUCON.2011.5773127.

[s38] P. Hoffman, C. Scott, Incorporation of Spiral and Agile Development into an Existing CMM Level 3 Process, in: Proceedings of the 49th International Instrumentation Symposium, ISA, Orlando, USA, May 4–8, 2003, pp. 105–117.

[s39] O. Salo, P. Abrahamsson, Integrating agile software development and software process improvement: a longitudinal case study, in: Proceedings of the 2005 International Symposium on Empirical Software Engineering, IEEE, November 17–18, 2005, pp.193–202. doi:10.1109/ISESE.2005.1541828.

[s40] C.R. Jakobsen, T. Poppendieck, Lean as a Scrum Troubleshooter, in: Proceedings of the Agile Conference 2011 (AGILE'11), IEEE, Salt Lake City, USA, August 7–13, 2011, pp. 168–174. doi:10.1109/AGILE.2011.11.

[s41] S. Raman, Lean software development: is it feasible?, in: Proceedings of the 17th Digital Avionics Systems Conference (DASC'98), The AIAA/IEEE/SAE, Bellevue, USA, October 31-November 7, 1998, pp.C13/1-C13/8. doi:10.1109/DASC.1998.741480.

[s42] N. Perez, E. Ambrose, Lessons learned in using agile methods for process improvement, CrossTalk 20 (8) (2007) 7–11.

[s43] H. Glazer, Love and marriage: CMMI and agile need each other, CrossTalk 23 (1–2) (2010) 29–34.

[s44] L. Rusu, Y. Lin, G. Hodosi, Management Guidelines for Database Developers' Teams in Software Development Projects, in: Proceedings of the Second World Summit on the Knowledge Society (WSKS'09), Springer, Chania, Greece, September 16–18, 2009, pp. 377–386. doi:10.1007/978–3-642–04757-2_40.

[s45] J. Diaz, J. Garbajosa, J.A. Calvo-Manzano, Mapping CMMI Level 2 to Scrum Practices: An Experience Report, in: Proceedings of the 16th European Conference on Software Process Improvement (EuroSPI'09), Springer, Alcala, Spain, September 2–4, 2009, pp. 93–104. doi:10.1007/978–3-642–04133-4_8.

[s46] C.R. Jakobsen, K.A. Johnson, Mature Agile with a Twist of CMMI, in: Proceedings of the Agile Conference 2008 (AGILE'08), IEEE, Toronto, Canada, August 4–8, 2008, pp. 212–217. doi:10.1109/Agile.2008.10.

[s47] J. Martinsson, Maturing XP through the CMM, in: Proceedings of the 4th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2003), Springer, Genova, Italy, May 25–29, 2003, pp. 80–87. doi:10.1007/3–540-44870–5_11.

[s48] M. Rönkkö, A. Järvi, M.M. Mäkelä, Measuring and Comparing the Adoption of Software Process Practices in the Software Product Industry, in: Proceedings of the International Conference on Software Process (ICSP'08), Springer, Leipzig, Germany, May 10–11, 2008, pp. 407–419. doi:10.1007/978–3-540–79588-9_35.

[s49] J. Fecarotta, MyBoeingFleet and Agile Software Development, in: Proceedings of the Agile Conference 2008 (AGILE'08), Springer, Toronto, Canada, August 4–8, 2008, pp. 135–139. doi:10.1109/Agile.2008.72.

[s50] T. Kovacheva, N. Todorov, Optimizing software development process: A case study for integrated Agile-CMMI process model, in: Proceedings of the International Conference on Computer as a Tool (EUROCON'11), IEEE, Lisbon, Portugal, pp. 1–2, April 27–29, 2011. doi:10.1109/EUROCON.2011.5929328.

[s51] G. Srinivasa, P. Ganesan, Pair Programming: Addressing Key Process Areas of the People-CMM, in: Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods (XP/Agile Universe 2002), Springer, Chicago, USA, August 4–7, 2002, pp. 221–230. doi:10.1007/3–540-45672–4_21.

[s52] K. Lebsanft, Process Improvement in Turbulent Times - Is CMM Still an Answer?, in: Proceedings of the Third International Conference on Product Focused Software Process Improvement (PROFES'01), Springer, Kaiserslautern, Germany, September 10–13, 2001, pp. 78–85. doi:10.1007/3–540-44813–6_10.

[s53] F. Navarrete, P. Botella, X. Franch, Reconciling Agility and Discipline in COTS Selection Processes, in: Proceedings of the Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07), IEEE, Banff, Canada, February 26 - March 2, 2007, pp.103–113. doi:10.1109/ICCBSS.2007.29.

[s54] C.R. Jakobsen, J. Sutherland, Scrum and CMMI going from good to great, in: Proceedings of the Agile Conference 2009 (AGILE'09), IEEE, Chicago, USA, August 24–28, 2009, pp. 333–337. doi:10.1109/AGILE.2009.31.

[s55] J. Sutherland, C.R. Jakobsen, K. Johnson, Scrum and CMMI Level 5: The Magic Potion for Code Warriors, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS'08), IEEE, Waikoloa, Hawaii, January 7–10, 2008, pp. 466–466. doi:10.1109/HICSS.2008.384.

[s56] P. Santos, SCRUM meets CMMi, Dr. Dobb's Journal 32 (9) (2007) 28–33.

[s57] J. Wäyrynen, M. Bodén, G. Boström, Security Engineering and eXtreme Programming: An Impossible Marriage?, in: Proceedings of the 4th Conference on Extreme Programming and Agile Methods (XP/Agile Universe 2004), Springer, Calgary, Canada, August 15–18, 2004, pp. 117–128. doi:10.1007/978–3-540–27777-4_12.

[s58] E.S. Visconti, E. Spina, Software Development: Planning x Agility, in: Proceedings of the International Conference on Software Engineering Research and Practice (SERP'03), Las Vegas, USA June 23–26, 2003, pp. 165–170.

[s59] W. Spoelstra, M. Iacob, M.V. Sinderen, Software reuse in agile development organizations: a conceptual management tool, in: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC'11), ACM, New York, USA, March 21–24, 2011, pp. 315–322. doi:10.1145/1982185.1982255.

[s60] Aaen, A. Börjesson, L. Mathiassen, SPI agility: How to navigate improvement projects, Software Process: Improvement and Practice 12 (3) (2007) 267–281. doi:10.1002/spip.v12:3.

[s61] D. Homchuenchom, C. Piyabunditkul, H. Lichter, T. Anwar, SPIALS: A light-weight Software Process Improvement self-assessment tool, in: Proceedings of the 5th Malaysian Conference on Software Engineering (MySEC'11), IEEE, Johor Bahru, Malaysia, December 13–14, 2011, pp. 195–199. doi:10.1109/MySEC.2011.6140668.

[s62] C. Patel, M. Ramachandran, Story Card Maturity Model (SMM): A Process Improvement Framework for Agile Requirements Engineering Practices, Journal of Software 4 (5) (2009) 422–435. doi:10.4304/jsw.4.5.422–435.

[s63] M. Pikkarainen, O. Salo, R. Kuusela, P. Abrahamsson, Strengths and barriers behind the successful agile deployment - insights from the three software intensive companies in Finland, Empirical Software Engineering 17 (6) (2012) 675–702. doi:10.1007/s10664–011-9185–5.

[s64] D.J. Anderson, Stretching Agile to fit CMMI Level 3 - the story of creating MSF for CMMI® Process Improvement at Microsoft Corporation, in: Proceedings of the Agile Development Conference 2005 (ADC'05), IEEE, Washington, USA, July 24–29, 2005, pp. 193–201. doi:10.1109/ADC.2005.42.

[s65] N. Ramasubbu, R.K. Balan, The impact of process choice in high maturity environments: An empirical analysis, in: Proceedings of the 31st International Conference on Software Engineering (ICSE'09), IEEE, Vancouver, Canada, May 16–24, 2009, pp. 529–539. doi:10.1109/ICSE.2009.5070551.

[s66] H. Dahlberg, F.S. Ruiz, C.M. Olsson, The Role of Extreme Programming in a Plan-Driven Organization, in: Proceedings of the IFIP TC8 WG 8.6 International Working Conference, Springer, Galway, Ireland, June 7–10, 2006, pp. 291–312. doi:10.1007/0–387-34410–1_20.

[s67] J. Nawrocki, B. Walter, A. Wojciechowski, Toward maturity model for Extreme Programming, in: Proceedings of the 27th Euromicro Conference, IEEE, Warsaw, Poland, September 4–6, 2001, pp.233–239. doi:10.1109/EURMIC.2001.952459.

[s68] M. Pikkarainen, Towards a Better Understanding of CMMI and Agile Integration - Multiple Case Study of Four Companies, in: Proceedings of the 10th International Conference on Product Focused Software Process Improvement (PROFES'09), Springer, Oulu, Finland, June 15–17, 2009, pp. 401–415. doi:10.1007/978–3-642–02152-7_30.

[s69] D. Wildt, R. Prikladnicki, Transitioning from Distributed and Traditional to Distributed and Agile: An Experience Report, in: D. Šmite, N.B. Moe, P.J. Ågerfalk (Eds.), Agility Across Time and Space: Implementing Agile Methods in Global Software Projects, Springer, Berlin, pp. 31–46. doi:10.1007/978–3-642–12442-6_3.

[s70] D. Knudson, A. Radermacher, Updating CS capstone projects to incorporate new agile methodologies used in industry, in: Proceedings of the 24th IEEE-CS Conference on Software Engineering Education and Training (CSEET'11), IEEE, Honolulu, Hawaii, May 22–24, 2011, pp. 444–448. doi:10.1109/CSEET.2011.5876122.

[s71] R. Sison, T. Yang, Use of Agile Methods and Practices in the Philippines, in: Proceedings of the 14th Asia–Pacific Software Engineering Conference (APSEC'07), IEEE, Aichi, Japan, December 4–6, 2007, pp. 462–469. doi:10.1109/ASPEC.2007.35.

[s72] Garcia, C. Pacheco, J. Calvo-Manzano, Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting, IET Software 4 (4) (2010) 237–251. doi:10.1049/iet-sen.2009.0045.

[s73] J. Gómez, Using Agile Practices and the CMMI to achieve High Project Management Capability in Small Settings, in: Proceedings of the First International Research Workshop for Process Improvement in Small Settings, SEI, Pittsburgh, USA, October 19–20, 2005, pp. 102–107.

[s74] J. Schalken, S. Brinkkemper, H.V. Vliet, Using linear regression models to analyse the effect of software process improvement, in: Proceedings of the 7th International Conference on Product Focused Software Process Improvement (PROFES'06), Springer, Amsterdam, The Netherlands, June 12–14, 2006, pp. 234–248. doi:10.1007/11767718_21.

[s75] D. Smite, C. Gencel, Why a CMMI Level 5 Company fails to meet the Deadlines?, in: Proceedings of the 10th International Conference on Product-Focused Software Process Improvement (PROFES'09), Springer, Oulu, Finland, June 15–17, 2009, pp. 87–95. doi:10.1007/978–3-642–02152-7_8.

[s76] D.J. Reifer, XP and the CMM, IEEE Software 20 (3) (2003) 14–15. doi:10.1109/MS.2003.1196314.

[s77] J.A.H. Alegría, M.C. Bastarrica, Implementing CMMI using a Combination of Agile Methods, Clei Electronic Journal 9 (1) (2006) 1–15.

[s78] C.R. Jakobsen, J. Sutherland, Mature Scrum at Systematic, Methods & Tools 17 (3) (2009) 2–14.

[s79] M. Pikkarainen, A. Mäntyniemi, An Approach for using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies, in: Proceedings of the 6th International Conference on Software Process Improvement and Capability dEtermination (SPICE'06), Luxemburg, May 4–5, 2006, pp. 1–11.

[s80] M.C. Paulk, Agile methodologies and process discipline, CrossTalk 15 (20) (2002) 15–18.

[s81] M. Lycett, R.D. Macredie, C. Patel, R.J. Paul, Migrating Agile Methods to Standardized Development Practice, IEEE Computer 36 (6) (2003) 79–85. doi:10.1109/MC.2003.1204379.

## Appendix B. Overview of studies included

| Id | Research Method | Agile Method | Team Experience | Professional/ Student | Project Duration | Team Size | Domain, Comment |
|---|---|---|---|---|---|---|---|
| s1 | Experience Report | XP + Scrum | Mature | Professional | – | 18 | Healthcare, lifestyle and technology |
| s2 | Experience Report | Other | – | Professional | – | 30 | Outsourcing and application system solutions |
| s3 | Theoretical | PMBOK + Scrum | NA | NA | NA | NA | Telecommunications industry |
| s4 | Single-case | XP | Mature | Professional | 8 weeks | 4 | Intranet application for managing research information |
| s5 | Theoretical | General | NA | NA | NA | NA | Indigenous Irish software organizations |
| s6 | Experience Report | General | – | Professional | 6 months | 72 teams | Front office trading software |
| s7 | Experience Report | XP | – | Professional | – | – | Small and mid-size enterprises |
| s8 | Theoretical | General | NA | NA | NA | NA | – |
| s9 | Experience Report | General | Mature | Professional | – | 19 participants | – |
| s10 | Theoretical | XP | NA | NA | NA | NA | Small and mid-size enterprises from Pakistan |
| s11 | Experience Report | Scrum/XP | Mature | Professional | – | – | – |
| s12 | Theoretical | General | NA | NA | NA | NA | – |
| s13 | Experience Report | General | – | Professional | – | – | – |
| s14 | Action Research | General | – | Professional | – | 8 | Automotive software |
| s15 | Experience Report | XP + Scrum | Mature | Professional | – | 28 | Internal software for Phillips |
| s16 | Experience Report | Other | – | Professional | – | 1–10 | Work for the federal government |
| s17 | Theoretical | Lean | NA | NA | NA | NA | – |
| s18 | Single-case | Lean + Scrum + TDD | – | Professional | 35 days/40 days | 7 | Software related to receipt, control and administration of funds |
| s19 | Theoretical | XP + Scrum | NA | NA | NA | NA | Software development and system integrator – government agency/Software development – general industry |
| s20 | Single-case | Lean/General | – | Professional | – | – | – |
| s21 | Experience Report | XP/General | Mature | Professional | 4 years | 100 | System to support research, development and train future U.S. military leaders |
| s22 | Experience Report | XP | – | Professional | – | – | Information and communication technology |
| s23 | Survey | Scrum | – | Professional | NA | NA | NA |
| s24 | Experience Report | XP + Scrum | – | Professional | – | 26 | Ambient intelligence, storage, networks, copy protection and robotics |
| s25 | Survey | Scrum | – | Professional | 6 months | Average 5 | – |
| s26 | Theoretical | General | NA | NA | NA | NA | NA |
| s27 | Experience Report | Scrum | Mature | Professional | – | 12 | Electronic billing |
| s28 | Single-case | General | Mature | Professional | 1–2 years | ~300 | Telecommunication equipment |
| s29 | Theoretical | XP | NA | NA | NA | NA | Small and medium enterprises |
| s30 | Experience Report | Scrum | – | Professional | 90 days | 10 | Financial services, insurance, travel, transport, retail, distribution and government sectors |
| s31 | Theoretical | XP | NA | NA | NA | NA | NA |
| s32 | Theoretical | XP | NA | NA | NA | NA | NA |

**Appendix B** (*continued*)

| Id | Research Method | Agile Method | Team Experience | Professional/ Student | Project Duration | Team Size | Domain, Comment |
|---|---|---|---|---|---|---|---|
| s33 | Experience Report | General | – | Professional | 1 year | 350+ | Energy-related businesses and services worldwide |
| s34 | Experience Report | XP/General | – | Professional | 6 years | 860 | Energy-related businesses and services worldwide |
| s35 | Experience Report | XP | – | Professional | – | 2–20 | Build management system from a simple version control system |
| s36 | Experience Report | Scrum | – | Professional | – | 5 | Midsize technology services company |
| s37 | Experience Report | General | – | Student | – | – | Application life cycle management (ALM) tools |
| s38 | Experience Report | General | – | Professional | – | 130 | Aerospace ground test facilities |
| s39 | Mixed | XP/Other | Mature | Both | 9 weeks/9 weeks/9 weeks/11 weeks/8 weeks | 8.5/10/5.5/ 5.2/7.1 | Intranet app and mobile app |
| s40 | Multi-case | Lean/Scrum/Other | Mature | Professional | 5 years | 450+ | Complex and critical IT solutions |
| s41 | Theoretical | Lean | NA | NA | NA | NA | Aerospace industry |
| s42 | Experience Report | General | – | Professional | 9 months | 75 | Information technology, engineering and operations services |
| s43 | Experience Report | Scrum/XP | – | Professional | 9 months | ~12 | – |
| s44 | Single-case | Scrum | – | Professional | – | 10+ | Games |
| s45 | Single-case | Scrum | – | Professional | 15 weeks | 8 | Test and operation environment |
| s46 | Experience Report | Scrum | – | Professional | – | 450+ | Complex and critical IT solutions |
| s47 | Theoretical | XP | NA | NA | NA | NA | – |
| s48 | Survey | Scrum/XP | – | Professional | – | – | – |
| s49 | Experience Report | Scrum | – | Professional | 24 months | – | – |
| s50 | Experience Report | General | – | Professional | – | – | – |
| s51 | Theoretical | XP | NA | NA | NA | NA | – |
| s52 | Experience Report | XP | – | Professional | – | – | – |
| s53 | Theoretical | General | NA | NA | NA | NA | Commercial off-the-shelf |
| s54 | Experience Report | Scrum + Lean | – | Professional | – | 500+ | Complex and critical IT solutions |
| s55 | Experience Report | Scrum + Lean | – | Professional | – | 4/5/10/19 | Large systems used in the defense, healthcare, manufacturing and service industries |
| s56 | Experience Report | Scrum | – | Professional | 14 months | – | Configuration-management and version-control tool |
| s57 | Theoretical | XP | NA | NA | NA | NA | – |
| s58 | Theoretical | XP | NA | NA | NA | NA | – |
| s59 | Single-case | General | – | Professional | – | 9 | Medium sized Dutch software development organization |
| s60 | Multi-case | General | – | Professional | – | 18 projects | Telecom company |
| s61 | Theoretical | Scrum | NA | NA | NA | NA | Software process improvement adaptive learning system |
| s62 | Experience Report | XP | – | Professional | – | 11–28/9–23/12–19 | Flyer design/Software house/ Web development and hosting |
| s63 | Multi-case | XP + Scrum/ Scrum + XP + Other | Mature | Professional | 2.5 years | 57 | Financial/Telecom/Information security |
| s64 | Experience Report | Other | – | Professional | – | – | Tool for process improvement |
| s65 | Survey | Agile RUP/XP/ Scrum | Mature | Professional | – | 112 projects | Development centers |

(continued on next page)

**Appendix B** (*continued*)

| Id | Research Method | Agile Method | Team Experience | Professional/ Student | Project Duration | Team Size | Domain, Comment |
|---|---|---|---|---|---|---|---|
| s66 | Single-case | XP | Mature | Professional | – | 4–15 | Transportation and vehicle industry |
| s67 | Theoretical | XP | NA | NA | NA | NA | – |
| s68 | Multi-case | XP + Scrum | – | Professional | – | 33 | Telecom, information security and financial sectors |
| s69 | Experience Report | XP + Scrum + Lean | – | Professional | 5 months/14 months | 7/20 | Hardware manufacturing |
| s70 | Experience Report | General | – | Professional | – | – | – |
| s71 | Multi-case | XP/Scrum | – | Professional | – | –100/12 | Integrator of IT solutions with clients/Game |
| s72 | Survey | General | – | Professional | 221 h./400 h./600 h. | 20/15/17 | Web-based portals/Software solution providers/Software solution providers and consultancy |
| s73 | Experience Report | Scrum/Crystal Clear | – | Professional | 6 months | Average 10 | – |
| s74 | Survey | DSDM | – | Professional | 4 years | 410 projects | Financial institution |
| s75 | Single-case | Scrum | – | Professional | 10 months | 4 teams | Web-based software application for a call centre |
| s76 | Experience Report | XP | – | Professional | – | – | Large web-based projects |
| s77 | Experience Report | XP + Scrum | – | Professional | – | 9 | – |
| s78 | Experience Report | Scrum + Lean | – | Professional | 5 years | 500+ | Complex and critical IT solutions |
| s79 | Multi-case | General/ XP + Scrum/Scrum/ Other | – | Professional | – | 2–5/6/~6 | Embedded software development branch |
| s80 | Theoretical | General | NA | NA | NA | NA | – |
| s81 | Experience Report | RUP + General | – | Professional | – | – | Information management |

## Appendix C. Quality assessment result

| Study | 1 Research | 2 Aim | 3 Context | 4 Design | 5 Sampling | 6 Control | 7 Collection | 8 Analysis | 9 Reflexivity | 10 Findings | 11 Value | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| s2 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 6 |
| s3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| s4 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 8 |
| s5 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| s6 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s7 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s8 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s10 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s11 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| s13 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s14 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 8 |
| s15 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s16 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| s18 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 6 |
| s19 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 4 |

**Appendix C** (*continued*)

| Study | 1 Research | 2 Aim | 3 Context | 4 Design | 5 Sampling | 6 Control | 7 Collection | 8 Analysis | 9 Reflexivity | 10 Findings | 11 Value | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s20 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 6 |
| s21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| s22 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s23 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 7 |
| s24 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s25 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 6 |
| s26 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s27 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s28 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 10 |
| s29 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s30 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s31 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s32 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s33 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s34 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s35 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s36 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| s38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| s39 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |
| s40 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 6 |
| s41 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| s42 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 |
| s43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| s44 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 5 |
| s45 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| s46 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s47 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| s48 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |
| s49 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s51 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| s52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| s53 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s54 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 |
| s55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| s56 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| s57 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s58 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s59 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 6 |
| s60 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 9 |
| s61 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 3 |
| s62 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s63 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 |
| s64 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s65 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 |
| s66 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 7 |
| s67 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s68 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 10 |
| s69 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 4 |
| s70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s71 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| s72 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 |
| s73 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s74 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 10 |
| s75 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 7 |
| s76 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s77 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| s78 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| s79 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 9 |

(*continued on next page*)

**Appendix C** (*continued*)

| Study | 1 Research | 2 Aim | 3 Context | 4 Design | 5 Sampling | 6 Control | 7 Collection | 8 Analysis | 9 Reflexivity | 10 Findings | 11 Value | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s80 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s81 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 23 | 51 | 55 | 10 | 9 | 11 | 21 | 16 | 6 | 54 | 63 | |

## Appendix D. Distribution of studies by publication channel and occurrence

| Publication channel | Type | Number |
|---|---|---|
| Agile Conference | Conference | 12 |
| PROFES | Conference | 7 |
| CrossTalk | Journal | 6 |
| XP/Agile Universe | Conference | 4 |
| IEEE Software | Journal | 3 |
| XP 20XX | Conference | 3 |
| ICSE | Conference | 2 |
| ACM-SAC | Conference | 1 |
| ACM-SE | Conference | 1 |
| Agile Journal | Journal | 1 |
| Agility Across Time and Space | Book | 1 |
| APSEC | Conference | 1 |
| ASEA | Conference | 1 |
| CERMA | Conference | 1 |
| CLEI Electronic Journal | Journal | 1 |
| Computer and Information Science | Book | 1 |
| CSEE&T | Conference | 1 |
| Cutter IT Journal | Journal | 1 |
| DASC | Conference | 1 |
| DB&IS | Conference | 1 |
| Dr. Dobb's Journal | Journal | 1 |
| EDUCON | Conference | 1 |
| Empirical Software Engineering | Journal | 1 |
| EUROCON | Conference | 1 |
| EuroMicro Conference | Conference | 1 |
| EuroSPI | Conference | 1 |
| HICSS | Conference | 1 |
| HPAGC | Conference | 1 |
| IAIT | Conference | 1 |
| ICCBSS | Conference | 1 |
| ICCSSE | Conference | 1 |
| ICENT | Conference | 1 |
| ICSP | Conference | 1 |
| ICTTA | Conference | 1 |
| IEEE Computer | Journal | 1 |
| IEEE-RE | Conference | 1 |
| IET Software | Journal | 1 |
| IFIP Working Conference | Conference | 1 |
| IIS | Conference | 1 |
| Information and Software Technology | Journal | 1 |
| Innovations in Systems and Software Engineering | Journal | 1 |
| ISEC | Conference | 1 |
| ISESE | Conference | 1 |
| Journal of Software | Journal | 1 |
| Methods & Tools | Journal | 1 |
| MySEC | Conference | 1 |
| SEI-IRWPISS | Conference | 1 |
| SERP | Conference | 1 |
| Software Process: Improvement and Practice | Journal | 1 |
| SPICE | Conference | 1 |
| WSKS | Conference | 1 |
| Total | | 81 |

## References


[1] SEI, Software Engineering Institute, CMMI® for Development, Version 1.3, Technical Report, CMU/SEI-2010-TR-033, November 2010, 470 p.
[2] CMMI Institute, Maturity Profile Reports, March 2013. <http://cmmiinstitute.com/assets/presentations/2013MarCMMI.pdf> (accessed March 2014).
[3] Agile Manifesto, Manifesto for Agile Software Development, 2001. <http://www.agilemanifesto.org/> (accessed September 2013).
[4] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, Boston, 2000. 224 p.
[5] K. Schwaber, M. Beedle, Agile Software Development with Scrum, Prentice Hall, 2002.
[6] VersionOne, 7th Annual State of Agile Development Survey, 2013. <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf> (accessed March 2014).
[7] E. Hasnain, An overview of published agile studies: a systematic literature review, in: Proceedings of the 2010 National Software Engineering Conference (NSEC'10), ACM, New York, NY, USA, 2010, pp. 6. doi:10.1145/1890810.1890813.
[8] SOFTEX, Association for Promoting Excellence in Brazilian Software, MPS.BR – Brazilian Software Process Improvement, SPI General Guide for Software, December 2012, 59 p.
[9] R.A. Souza, F. Selleri, F.S.F. Soares, S.R.L. Meira, Benefits and Limitations of Using the MPS.BR Model with Agile Methodologies: a survey based on a systematic literature review, in: Proceedings of the Eighth International Conference on Software Engineering Advances (ICSEA'13), IARIA, Venice, Italy, October 27–31, 2013, pp. 373–379.
[10] B. Boehm, A view of 20th and 21st century software engineering, in: Proceedings of the 28th International Conference on Software Engineering (ICSE'06), ACM, New York, NY, USA, 2006, pp. 12–29. doi:10.1145/1134285.1134288.
[11] J.A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, Dorset House, New York, 2000. 392p.
[12] A. Cockburn, Agile Software Development, Addison-Wesley, 2000.
[13] J. Stapleton, DSDM: Business Focused Development, Pearson Education, 2003. 272p.
[14] S.R. Palmer, J.M. Felsing, A Practical Guide to Feature-Driven Development, Prentice Hall, 2002.
[15] M. Poppendieck, T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003.
[16] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Inform. Softw. Technol. 50 (9–10) (2008) 833–859, http://dx.doi.org/10.1016/j.infsof.2008.01.006.
[17] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods: review and analysis, VTT Technical report, 2002, 107 p.
[18] D. Cohen, M. Lindvall, P. Costa, An introduction to agile methods, in: M.V. Zelkowitz (Ed.), Advances in Computers, Advances in Software Engineering, vol. 62, Elsevier, Amsterdam, 2004, p. 66. doi: 10.1016/S0065-2458(03)62001-2.
[19] J. Erickson, K. Lyytinen, K. Siau, Agile modeling, agile software development, and extreme programming: the state of research, J. Database Manage. 16 (4) (2005) 88–100, http://dx.doi.org/10.4018/jdm.2005100105.
[20] M. Staples, M. Niazi, Systematic review of organizational motivations for adopting CMM-based SPI, Inform. Softw. Technol. 50 (7–8) (2008) 605–620, http://dx.doi.org/10.1016/j.infsof.2007.07.003.
[21] M. Sulayman, E. Mendes, A systematic literature review of software process improvement in small and medium web companies, in: Proceedings of the International Conference on Advanced Software Engineering and Its Applications (ASEA'09), Springer, Jeju Island, Korea, 2009, pp. 1–8. doi:10.1007/978-3-642-10619-4_1.
[22] M. Unterkalmsteiner, T. Gorschek, A. Islam, C. Cheng, R. Permadi, R. Feldt, Evaluation and measurement of software process improvement – a systematic literature review, IEEE Trans. Softw. Eng. 38 (2) (2012) 398–424, http://dx.doi.org/10.1109/TSE.2011.26.
[23] A.M. Magdaleno, C.M.L. Werner, R.M. Araujo, Reconciling software development models: a quasi-systematic review, J. Syst. Softw. 85 (2) (2012) 351–369, http://dx.doi.org/10.1016/j.jss.2011.08.028.
[24] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007, 57p.
[25] Z. Racheva, M. Daneva, K. Sikkel, Value creation by agile projects: methodology or mystery? in: Proceedings of the 10th International Conference on Product-Focused Software Process Improvement (FROFES'09), Springer, Oulu, Finland, 2009, pp. 141–155. doi:10.1007/978-3-642-02152-7_12.
[26] G.W. Noblit, R.D. Hare, Meta-Ethnography: Synthesizing Qualitative Studies, Sage Publications, London, 1988. 88p.
[27] P.E. McMahon, Taking an agile organization to higher CMMI maturity, CrossTalk 25 (2012) 19–23.
[28] M. Montesi, P. Lago, Software engineering article types: an analysis of the literature, J. Syst. Softw. 81 (10) (2008) 1694–1714, http://dx.doi.org/10.1016/j.jss.2007.11.723.
[29] D. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press, Washington, 2010. 280p.
[30] D. Atkins, D. Best, P.A. Briss, M. Eccles, Y. Falck-Ytter, S. Flottorp, G.H. Guyatt, R.T. Harbour, M.C. Haugh, D. Henry, S. Hill, R. Jaeschke, G. Leng, A. Liberati, N. Magrini, J. Mason, P. Middleton, J. Mrukowicz, D. O'Connell, A.D. Oxman, B. Phillips, H.J. Schünemann, T. Edejer, H. Varonen, G.E. Vist, J.W. Williams Jr., S. Zaza, Grading quality of evidence and strength of recommendations, BMJ 328 (2004) 1490–1494.
[31] F.S.F. Soares, S.R.L. Meira, An agile maturity model for software development organizations, in: Proceedings of the Eighth International Conference on Software Engineering Advances (ICSEA'13), IARIA, Venice, Italy, October 27–31, 2013, pp. 324–328.