*Master Thesis*
*Software Engineering*
*Thesis no:*
*January 2014*

# Agile Methodologies and Software Process Improvement Maturity Models, Current State of Practice in Small and Medium Enterprises

**Vasileios Koutsoumpos – Iker Marinelarena**

Dept. of Software Engineering (DIPT)
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Dept. of Software Engineering (DIPT) at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 10 weeks of full time studies.

**Contact Information:**
Author(s):
Vasileios Koutsoumpos – 870702-7598
Iker Marinelarena – 891124-1290
E-mail:
bkoutsoub@gmail.com
icka69@gmail.com

University advisor:
Stefanie Betz
Dept. of Software Engineering (DIPT)

Dept. of Software Engineering
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

# Agile Methodologies and Software Process Improvement Maturity Models, Current State of Practice in Small and Medium Enterprises

Koutsoumpos Vasileios, Marinelarena Iker

*Abstract—Background:* **Software Process Improvement (SPI) maturity models have been developed to assist organizations to enhance software quality. Agile methodologies are used to ensure productivity and quality of a software product. Amongst others they are applied in Small and Medium – sized Enterprises (SMEs). However, little is known about the combination of Agile methodologies and SPI maturity models regarding SMEs and the results that could emerge, as all the current SPI models are addressed to larger organizations and all these improvement models are difficult to be used by Small and Medium – sized firms. Combinations of these methodologies could lead to improvement in the quality of the software products, better project management methodologies and organized software development framework.**

*Objectives:* **The aim of this study is to identify the main Agile methodologies and SPI maturity models applied in SMEs, the combinations of these methodologies, and the results that could emerge. Through these combinations, new software development frameworks are proposed. What is more, the results of this study can be used as a guide with the appropriate combination for each SME, as a better project management methodology or as improvement in the current software engineering practices.**

*Methods:* **A Systematic Literature Review was conducted, resulting in 71 selected relevant papers ranging from 2001 to 2013. Besides, a survey has been performed from June 2013 to October 2013, including 49 participants.**

*Results:* **Seven Agile methodologies and six different SPI maturity models were identified and discussed. Furthermore, the combination of eight different Agile methodologies and Software Process Improvement maturity models is presented, and as well as their benefits and drawbacks that could emerge in Small and Medium – sized firms.**

*Conclusion:* **The majority of the Agile methodologies and SPI maturity models are addressed to large or very large enterprises. Thus, little research has been conducted for SMEs. The combinations of the Agile methodologies and SPI maturity models are usually performed in experimental stages. However, it has been observed that such type of combination could present numerous benefits, which can also be applicable in SMEs as well. The combinations that are most common are the CMMI and XP, CMMI and Scrum, CMMI and Six Sigma, and the PRINCE2 and DSDM.**

*Index Terms*—**Agile methodologies, combination, Software Process Improvement, SMEs, survey, Systematic Literature Review.**

## I. Introduction

Small and Medium Enterprises (SMEs), are the most widespread type of enterprises around the different global economies [7][8] of this rapidly changing world [9], and they can be resembled as foundation and "motor" of the industrial growth [2][10]. SMEs are the enterprises, in which the number of employees is less than 50, for small enterprises, and less than 250 in the case of the medium – sized ones [12].

A lot of these SMEs are focusing on developing software. All of these enterprises implement Software Engineering practices to a greater or lesser extend in order to develop their products. Software Engineering practices have achieved a lot of importance, because if they are not applied and performed adequately, various problems can arise during the software development [13][14][15].

Performing incorrectly the Software Engineering practices can for example lead to the development of a system that contains properties, which were not requested [13]. The major impact is the rework that has to be done; it has been proved that the rework can cost up to 40% of the total project cost [14]. If errors are discovered late in a Software Engineering processes the cost can be 200 times more, than catching them in the early phases of the development process [14][15].

As it was mentioned before, this world is changing rapidly; the same happens with software projects. In order to be able to satisfy the new necessities of the market, Agile methodologies were implemented, providing enterprises with a group of faster, more flexible, with a continue and easy learning and responsiveness [2].

In addition, applying Agile methodologies is a step forward in the software development environment, but still there are many other aspects that are not fully addressed by implementing Agile methodologies, such as quality assurance, time management, and so forth. Software Process Improvement techniques are applied to enterprises in order to fulfill these needs. Their objective is to manage and improve software processes to satisfy the customer's requirements within the time frame at a lower cost, while maintaining the quality of the software product [16][17][18].

### A. Contributions

The aim of this thesis project is to identify the current state of practice of the Agile methodologies and the SPI maturity models and as well as their combinations that could be applied in Small and Medium – sized firms. In order to meet the aim, the following objectives are defined:

- Investigate:
    - Search through the literature for the identification on the most common methodologies.
    - Identify the benefits, challenges and drawbacks for each methodology.
- Find out the "real situation" in Small and Medium Enterprises.
- Compare the Literature and the "real examples".

### B. Research Questions

To achieve the aims and objectives the following research questions will be addressed:

RQ1: What agile methodologies combined with Software Process Improvement maturity models crafted for Small and Medium Enterprises exist?

RQ2: Under which situations and how could these methodologies be applied?

- What are the results – benefits – drawbacks that each methodology could provide?
- When should each methodology be applied?
- What are the reasons for failure?

RQ3: Are these methodologies really applied in SMEs?

### C. Research methodology

Details of the specific research that were used are presented in **Fig. I.1**. The overall goal of the thesis was to identify the current state of practice of the Agile methodologies and the SPI maturity models in Small and Medium sized – firms, which are described not only in articles and books, but also in real examples, by including practitioners. This goal was achieved by the research questions described in **Table I.1**.

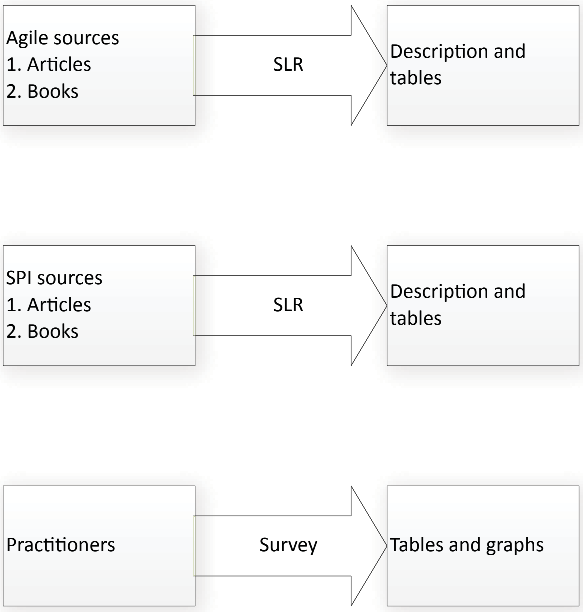

Figure I. 1 Overview of research

TABLE I.1
RESEARCH QUESTIONS AND THEIR CORRESPONDING METHODS

| Research Question | Methodology |
|---|---|
| RQ1 | Systematic Literature Review [1] |
| RQ2 | Systematic Literature Review, Survey [1][3] |
| RQ3 | Survey [3] |

### D. Outline of the thesis

**Section II** presents the background of the thesis and provides descriptions and definitions of the main concepts used in the thesis. In addition, it identifies the research gap that this thesis will cover. **Section III** describes the most common Agile methodologies, the most common SPI maturity models and the most common combinations of Agile methodologies and SPI maturity models, based on the SLR. The SLR is used to answer the **RQ1** and **RQ2,** by providing details regarding the way the combinations could be adopted by SMEs, and as well as their benefits and drawbacks that could emerge. Next **Section IV** presents the results of the survey. The survey is used to answer the **RQ2** and **RQ3**, by describing what is the "real" situation in enterprises, and if these combinations could actually be adopted by SMEs. **Section V** discusses the threats to validity, concerning this research study. Finally, **Section VI** concludes the paper, by answering briefly all the research questions.

## II. BACKGROUND

Various papers have been published the last years focusing on the domain of Software Engineering practices and methods for improvement [19][20][21][22][23][24]. Through these studies, it has been proven that the Software Engineering (SE) practices are performed differently in SMEs, than the way larger enterprises perform, or the way textbooks describe [25]. For example, in SMEs, although different roles exist, they are not so clear and every employee performs many different kinds of tasks. Thus, the work is done in a rather informal environment heavily relying on collaboration.

The literature that exists regarding improvement models for SMEs is limited; this is because SMEs present unique characteristics that originate from their make and ownership [7][25]. SMEs have got specific challenges, due to their size and budget under which they operate. In addition, they present low maturity levels and fewer resources, in order to consider quality and process improvements. Only a few SMEs document their software engineering processes. Kamsties et al. [10] in their study prove that there are not clear ways in SMEs, to perform the software engineering practices properly.

Simon et al. [26] show that SMEs do not emphasize on training, as they have pressing deadlines, and the time that is available for improvement is limited. Instead, SMEs follow a simplified process lifecycle, where they pay particular attention on developing and testing. Even more, they present lack in control procedures, project management and risk management skills [26]. Mishra et al. [27] present that SMEs cannot measure the benefits and the process progress that can be derived from such processes.

Many Small and Medium Enterprises are willing to improve their software engineering processes, in order to develop successful software systems [7][27]. However, they find it difficult to implement these improvement processes, due to the high cost that is demanded to implement them. Furthermore, the limited resources and the strict time constraints that the SMEs operate make the adoption of these techniques even harder [27].

What is more, the research in software process is motivated by the fact that process quality is related with the quality of the software product [28][29]. The aim of the Software Process Improvement is not only to increase product quality, but also to increase reliability, consistency and predictability and reduce time to market [28][30].

Although several software process improvement techniques have been introduced to be applicable in SMEs such as CMMI [31], ISO/IEC 15504 [32], PRINCE2 [33], and so forth, the problem is that usually all these improvement methods are suitable only for large enterprises [16]. Changes have to be done in these methodologies, so that they can fit the SMEs needs, due to the fact that they operate in strict budget, and they are highly affected by the customer's needs [34]. In addition, as the conditions in which these techniques can be applied are not well defined, the benefits that each method can provide are not well stated [35].

Thus, experienced practitioners came up with another approach for improvement. The approach was labeled as Agile software development. This method, presents high impact on how a software product is developed worldwide [36]. However, although the past years many Agile methodologies have been developed, little is known about how these methods are carried out in practice and what are their effects [36]. For example, Agile methodologies can affect negatively the project's main characteristics, such as scope, time, cost and quality [37]. In general, the Agile methodologies suffer from the lack of disciplined planning [38].

Since, the Software Process Improvement methods are shown to be unsuitable for the SMEs, studies suggest that Small and Medium – sized companies can adopt Agile development methodologies, while following the SPI maturity models [39]. Through this new way of working, SMEs can gain a new competitive environment [39]. These methodologies take advantage of the flexibility and adaptability of Agile methodologies, and of SPI methods main value: control [38]. The combination of SPI maturity models and Agile methods would be beneficial, since the former tell us what to do, and the latter tell us how to do it [40]. The main issue is that there are no studies that analyze and compare the different possible models and combinations. For this reason, it is not easy for the SMEs to decide which of these models they could adopt.

In the next section, the structure and the results of the Systematic Literature Review are analyzed and discussed.

## III. SYSTEMATIC LITERATURE REVIEW

The Systematic Literature Review provides answers to the **research questions 1 and 2**. In order to answer the questions, the findings from the Systematic Literature Review will be discussed. According to Kitchenham **[1][5]**, an SLR is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. Therefore, it could be useful to a systematic understanding of scientific domain is required.

What is more, an SLR could be used to summarize the existing evidence concerning a specific domain, to identify possible gaps in current research and to suggest areas of further research and finally to provide a framework in order to appropriately position new research activities.

### *A. Review questions*

The research questions that will be addressed by this study are the **RQ1** and **RQ2** of the thesis. These questions are: **"What Agile methodologies combined with Software Process Improvement maturity models crafted for Small and Medium Enterprises exist?"** and **"Under which situations and how could these methodologies be applied?, What are the benefits and drawbacks for each methodology?, How could they be applied and what are the reasons for failure?"**

### *B. Data Sources and Search Strategy*

The process that was used to identify relevant studies and details of the search strategy are presented in **Fig. III.1**. The process was as follows. In the beginning relevant keywords were identified. Keywords were synonyms for the terms used in the topic of the thesis and the research questions. Next, search strings were formulated based on the identified keywords. Later, a trial search was performed on the BTH e-libraries. If the results were not sufficient, the research questions or the keywords were modified. The databases that were used were the ACM Digital Library, the IEEE Xplore, the Science Direct and the Inspec.

In order to narrow the number of results, filters were used to find the appropriate papers in the field of Software Engineering or Computer Science. Additionally, the papers that were collected are from 2001 and later on, due to the Agile Manifesto [41]. Finally, the duplicities were removed with the help of Zotero [4]. Literature for further reading was extracted based on the inclusion / exclusion criteria.

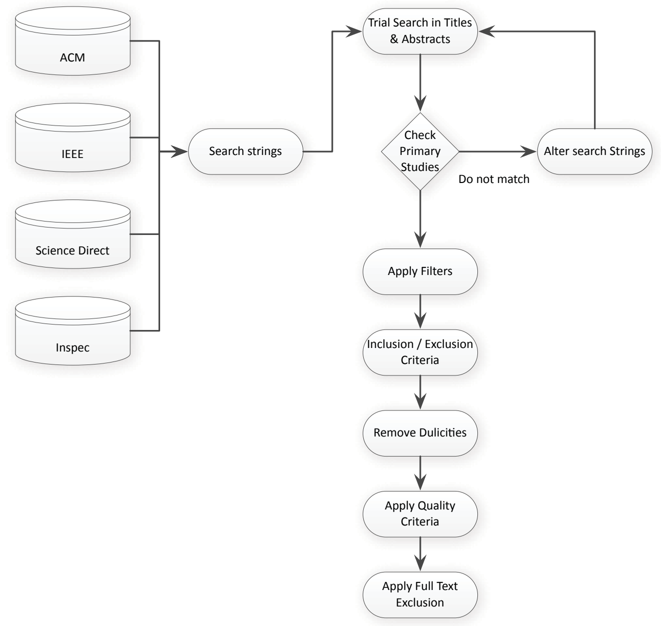

Figure III. 1 Search strategy to identify the relevant literature

### *C. Study Selection*

During the research process, various criteria have been used. They determined which studies should be included or excluded.

#### *1) Inclusion / Exclusion Criteria*

The choice of studies was based on the inclusion / Exclusion criteria listed bellow.

- Papers published from 2001 and later on due to the Agile Manifesto [41].
- Filters in the databases to limit the results in the field of Software Engineering and Computer Science domains.
- Papers that were not written in English were removed.
- Duplicates were removed with the help of Zotero.
- Peer reviewed articles were only included in the study.
- The abstract and keywords in a big number of papers was read.
- The number of citations that each article had was taken into consideration as a supporting method.
- In case of disagreement between the authors, a study of the articles was performed and then a discussion was carried out.

#### *2) Quality Assessment*

Once the selected articles for the research were gathered, quality assessment had to be performed, based on specific questions that are presented in **Appendix A**. Additionally, the questions were scored as follows: Y (Yes), P (Partially) and N (No). The scoring procedure was Y=1, P=0,5 and N=0, or unknown (i.e. the information is not specified). The articles that were scored bellow 5 were excluded from the Literature Review. Both the authors performed the scoring procedure individually and when there was a disagreement, they were discussing the issues, until they reached an agreement.

What is more, papers that were not conference or journal articles, such as symposium proceedings, section introductions, book chapters were excluded.

### *D. Data extraction*

#### *1) Keywords*

The keywords that were used for search were synonyms of the research topic and the research questions. Specifically, the keywords are described bellow:

- Agile methodologies,
- Software Process Improvement,
- SPI,
- SMEs,
- Maturity models.

#### *2) Search strings*

Pilot searches have proved that the keywords listed above can provide too broad searching results. Therefore, search queries were built from the identified keywords and their modifications (e.g. plural form) and Boolean operators such as the “AND” and the “OR”. The search strings were applied only for the Titles and Abstracts.

The authors decided initially to search literature for the main Agile methodologies and the main Software Process Improvement maturity models. For this reason, they used the first two search strings that are listed bellow. Next, they proceed to the combination, by using the third search string and to identify the most common models that are referred to the combinations. After identifying the most common methodologies, they proceed to specific literature search, by using the last three search strings. Specifically, the search queries are described bellow:

- (Agile method*) AND SMEs
- ((SPI OR software process improvement) AND method*) AND SMEs
- ((Agile method*) AND ((SPI OR software process improvement) AND method*)) AND SMEs
- (XP OR Extreme Programming OR Scrum OR ASD OR adaptive software development OR Lean development OR Dynamic systems development methodology OR Crystal methods) AND SMEs
- (CMM OR CMMI OR SPICE OR ISO 15504 OR P3M3 OR OPM3 OR PRINCE2) AND SMEs
- (XP OR Extreme programming OR Scrum OR ASD OR adaptive software development OR Lean development OR Dynamic systems development methodology OR Crystal methods) AND (CMM OR CMMI OR SPICE OR ISO 15504 OR P3M3 OR OPM3 OR PRINCE2) AND SMEs

#### *3) Search results*

After applying the search queries in the BTH e-libraries, the number of papers that were found, is presented in the following table.

TABLE III. 1
RELEVANT ARTICLES FOR THE SLR

| Sources | Discovered | Criteria application | Relevant studies |
|---|---|---|---|
| IEEE | 64 | 60 | 35 |
| ACM | 272 | 126 | 26 |
| Science Direct | 818 | 150 | 43 |
| Inspec | 354 | 221 | 59 |

Due to the fact that many of the relevant studies were found in more than one database, duplicated findings had to be removed. The total number of studies that were obtained by the authors for the needs of the thesis was 71.

### 4) *Data collection form*

The authors, in order to assist the data collection process, created a form that was applied to all the relevant studies. According to Kitchenham et. al. [1], the use of data collection forms is proposed, in order to remove bias in the data collection process.

The collection form contained the following data:

- Article's title,
- The research methodology that was used in the article,
- The database that was retrieved,
- The type of the article (e.g. conference, journal),
- Publication year,
- Link to the database,
- Comments,
- Quality score.

## E. *Results presentation*

This section summarizes the results from the Systematic Literature Review. The articles that were used for the purpose of the SLR are presented in **Appendix B**. The authors identified articles for the three parts of the thesis, first Agile methodologies, second SPI maturity models and third the combination of Agile and SPI. The results of the study are presented in the subsections bellow.

### 1) *Agile methodologies*

In the beginning of the 1990s, many developers found the traditional methodologies frustrating and sometimes impossible to apply on the day - to - day more dynamic environment [9]. The industry and the technology move too fast, requirements "change at rates that swamp traditional methods" [42]. In order to satisfy the necessities of this new environment, a group of methodologies that would address the challenge of an unpredictable world by relying on "people and their creativity rather than on processes" [17][43][44] was created, the alternative to heavyweight processes: Agile methodologies [41].

On February 11-13, 2001, in Utah, USA, seventeen representatives from Extreme Programming, SCRUM, Dynamic System Development (DSDM), Adaptive Software Development (ASD), Crystal Methodologies, Feature-Driven Development (FDD), Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened. What emerged from this meeting was the Agile Manifesto, which core ideas are the following [41]:

- *"Individuals and interaction over process and tools,*
- *Working software over comprehensive documentation,*
- *Customer collaboration over contract negotiation,*
- *Responding to change over following a plan".*

From these core ideas many definitions [45][46] of the meaning of being Agile have been given, but definitely according to these definitions the key attributes of an Agile organization are: speed, flexibility, learning and responsiveness [2]. Still, practitioners agree that being Agile involves more than simply following the methodology that makes a project Agile. Being Agile is more than a collection of practices; it's a way of thinking. As is said by Andrea Branca, *"many processes may look Agile, but they will not feel Agile"* [11, p.2].

Through the Systematic Literature Review, the main Agile methodologies that were identified are the following:

- Lean Software Development,
- Scrum,
- Extreme Programming (XP),
- Crystal Methodologies,
- Dynamic System Development Method (DSDM),
- Feature Driven Development (FDD),
- Adaptive Software Development (ASD).

In the following subsection the results of applying Agile methodologies on SMEs are analyzed.

#### *a) Agile on SMEs*

The new environment is forcing enterprises in adopting Agile methodologies. This can be clearly seen on a 2005 survey of the US and Europe, which revealed that 14 percent of companies were using Agile methods, and 49 percent of the companies aware of Agile methods were interested in adopting them [60].

Many studies have proved the validity of the main methodologies. When they are discussing about Agile in general, the main remarked benefits are: the improved communication and coordination [61], greater teamwork and effectiveness in the rapid production of software that meets customers' requirements [62].

It seems that the Agile methodologies are an approach to perfection. But this is not the reality, some practitioners claim that the studies investigate XP almost exclusively and that there is an increasing need for bigger variety (Scrum, Lean, etc.) and better research to determine the validity of these methodologies [63][64]. For others, the fact that *"there is a lack of literature describing projects where Agile failed to produce good results"* [40] attracts their attention. In order to try to make an approach to the reality, some studies state that several research studies have been published on Agile adoption by large enterprises, but the adoption of Agile in regulated environments has not been yet addressed [65] and therefore Agile methods and regulated environments are often seen as incompatible [63].

It has usually been said that Agile processes are more suitable for small and low risk projects [40][66] but the issue of best practices within SMEs has always been a challenge [66]. Thus, many practitioners claim that adopting Agile methodologies has a high cost of implementing new ideas [37][67][68] and still many time tools and techniques are applied in a wrong way [69].

### 2) *SPI maturity models*

Software Process Improvement is "*a systemic procedure for improving the performance of an existing process system by changing or updating the process*" [17, p.1][70]. Unfortunately, there is no specific SPI model for these firms as all the current SPI models were developed for large firms, and these improvement models are difficult and in many cases not suitable to be used by small software development firms, due to the fact that they are too complicated and expensive to be implemented [8]. However, some researchers indicated that SPI could be used as a competitive advancement strategy for both small and large organizations [17][71].

Through the SLR that the authors conducted, the most common SPI maturity models are the following:

- CMM,
- CMMI,
- ISO / IEC 15504 or else known as SPICE,
- PRINCE2,
- OPM3,
- P3M3.

In the following subsection, the results of applying Software Process Improvement maturity models on SMEs are discussed.

#### *a) SPI maturity models on SMEs*

The purpose of the several maturity models for software process improvement such as the CMM, the CMMI, the SPICE, and so forth, is to provide quality patterns and management frameworks that an enterprise could implement to improve its software development process [8].

Unfortunately, it has been observed that the successful implementation of such models is generally not possible within the context of Small and Medium – sized software organizations, as they are not capable of bearing the cost of implementing these software process improvement programs [17][72][73]. The proper implementation of software engineering techniques is a difficult task for SMEs, since they often operate on limited resources and with strict time constraints [73]. Small companies generally need external assistance in planning and implementing process improvement to keep abreast of state-of-the-art Software Engineering research and practice [74].

Many SMEs have recognized that the need to improve and evaluate their software product alone seems insufficient, since it is known that product's quality is largely dependent on the process that is used to create it [75]. Many researchers support that SMEs are characterized by lack of resources, lack of development and supporting environment, lack of budget and dependency on large organizations [76].

Dyba et al. [77] and [17] indicated that SPI maturity models can be used as a competitive advancement strategy for both small and larger organizations [77]. Furthermore, Cater and Steel [78] in their study proved that the software process improvement program was effective in improving the process capability of many of the SMEs.

Today, software industry is one of the most rapidly growing sectors and this situation stimulates especially the constant creation of small companies, which play an important role in the economy [73]. In the last few years, a great number of organizations have been interested in the SPI [8][79].

### 3) *Combination of Agile methodologies and SPI maturity models*

Nowadays software industry represents an important economical activity in both developed and underdeveloped countries [80]. This is the reason why the quality aspect in software development products represents one of the most important activities that has been applied, in order to ensure the software product quality for some years now [80]. Besides, organizations implement these activities to increase the quality and capability of its processes, products and services [81]. In this context, the measurement and analysis process in quality models and standards such as CMMI Dev 1.2 [82] and ISO/IEC 15504 [83] are highly adopted by the software industry to provide their software products and services [84].

However, in smaller industries such as the SMEs, the difficulty in adopting them increases, due to the wide of this models and standards. As a result, SMEs have a greater interest in adopting agile methodologies, which guarantee them to deliver software according to the capability [84].

Some possible benefits of a combination between the Agile methodologies and the SPI maturity models that could emerge for a SME, would be the improvement in the quality of the software product, more efficient project management methods, clear process of the software development and reduce the development cost [36][84][85].

Although research has been done regarding the Agile methodologies and the SPI maturity models, little is known about their relationship [86]. It is possible to support that the issues on the "marriage" between agile methods and SPI standards have not been investigated in sufficient depth and breadth [86].

Thus, continuous improvement of organizational software processes is important in enhancing the capabilities of an organization. The traditional approaches for organizational SPI, however, need to be altered to enable the co-existence of agile projects and organizational SPI [86]. Currently, there seems to be lack of empirical evidence on how the Agile approaches for SPI integrate to the organizational SPI activities. The existing methods for iterative adaptation and improvement of Agile project teams do not seem to address the organizational learning aspects [86].

Through the SLR, the most common combinations of SPI maturity models and Agile methodologies that the authors came up are the following:

- CMMI and XP,
- CMM and XP,
- PRINCE2 and XP,
- CMMI and Scrum,

- CMM and Scrum,
- CMMI and Lean,
- CMMI and Six Sigma,
- PRINCE2 and DSDM.

In the next subsection, the combinations presented above are analyzed, and as well as their adoption on the SMEs is also discussed. Specifically, the authors begin by presenting a small overview of the combinations, then they proceed in the discussion of their benefits and drawbacks and finally, they analyze if these models can be adopted by SMEs.

#### *a) CMMI and XP*

Some people think that XP and CMMI are like oil and water [39]. XP, as many of the Agile methodologies needs a framework in order to improve the software processes [17][39]. Still, the main reason for these two methodologies to be combined is the facilities that XP gives for learning, applying and adapting it. In addition XP already fulfills the CMMI level 2 [17].

To combine the methodologies, two different methodologies have been followed:

- Combine CMMI's Key Process Activities (KPA) combined with each XP principles to see if the methods are combinable [39][87]. The steps to be followed, in order to apply this combination are listed below:
  - Define the objective of the CMMI level; define which is the level that the enterprise wants to reach.
  - Analyze which are the different problems when combining the methodologies.
  - Check what is the actual state of the enterprise, both in CMMI and XP.
  - Finally, start applying the combination solving the conflicts analyzed before, until the previous state is reached.
- In order to fulfill as much as possible the CMMI specification, extends the XP method to fit all the CMMI KPA's [17]. In addition, an enterprise has to follow the steps described below, in order to adopt this combination.
  - Define the objective of the CMMI level; define which is the level that the enterprise wants to reach.
  - Analyze which are the different problems when combining the methodologies.
  - Check what is the actual state of the enterprise, both in CMMI and XP.
  - Finally, apply the combination by making the needed changes and extensions on the XP methodology in order to fully fulfill the desired CMMI level on its whole.

Combining the two models provides developers with a comprehensive spectrum of tools and options [17], besides it were shown that workshops and observation were productive ways for collecting assessment data from the Agile projects [87]. Finally, by adding CMMI to XP there was a significantly decrease of risk failures [39].

However, there are some KPA's that cannot be addressed by XP (such as, Organizational Process Performance or Quantitative Project Management) [17][39], so the XP methodology must be extended [17].

#### *b) CMM and XP*

Studies proved that CMM could be applicable in SMEs and benefits improvements in cost, development time, quality and customer satisfaction [88][89]. XP pays attention in customer focus and their satisfaction. Prefers teamwork and decentralized approach, where all the stakeholders have equal rank. The team focuses on the problems, and move towards the solution with mutual understanding in an efficient and effective manner [90].

It could be easily observed that the features of Agile are embedded in CMM. Research has shown that the combination of these two methodologies could reduce the cost of training and documentation that enterprises need. In this way, SMEs could be able to save valuable resources [90].

In XP the major phases are planning, managing, designing, coding and testing. CMM is consisted of five different levels – initial, repeatable, defined, managed and optimizing. However, some key process activities are common – the features of Agile are embedded in CMM – for these two models [90]. These activities are: defect prevention, organization process focus, software product engineering, intergroup coordination, software project planning, software quality assurance, and software configuration management.

Specifically, an enterprise, in order to apply this combination has to apply the XP features in the different CMMI levels [90]. In the CMMI level 5 – Optimizing, the enterprise has to apply the Continuous integration along with the Defect Prevention. In the CMMI level 3 – Defined, the XP features that have to be applied are team focus along with the organization process focus, simple design, coding standard and rested have to be applied along with the software product engineering key process activity. Moreover, the feature pair programming has to be applied along with the intergroup coordination of the CMMI. Finally, in the CMMI level 2 – Repeatable, the XP features small version and pair programming have to be applied with the CMMI key process activities software project planning and software quality assurance accordingly.

What is more, the good points from these two methodologies that could be used by SMEs are some of the following [90]:

- Improvements in the development cost and development time, customer satisfaction, increase in the quality of the software product, and reduce of the bureaucracy.
- Some other benefits that have been noticed are innovative and productive products.
- Better bonding on the teamwork and pair programming. Furthermore, CMM claims to be a flexible model that can be tailored and adopted to many lifecycles [91][92].

On the other hand, drawbacks have been noticed for this combination, as well.

- CMM seems to be expensive, as it requires training and skilled personnel.

- It increases the documentation.
- Furthermore, XP is not suitable for large projects and is more code centric rather than project centric.

Despite the drawbacks and other possible challenges, the adaption of these practices will reduce the cost of training and no documentation demanded will exist at early stages of software development. In this way, Small and Medium – sized firms will save more than the expenses. These practices make them to reserve capital and to make more profit [90].

### *c) PRINCE2 and XP*

PRINCE2 is a very popular management method known for its rigidness, strictness and document-centric nature. PRINCE2 has been a very useful method in planning and tracking software process. On the other hand, XP as an Agile method, suffers from lack of disciplined planning [38]. The main objective of this combination is to extract the best out of XP and PRINCE2. This will lead to a new method that will take advantage of the main XP's principles: flexibility and adaptability, and of PRINCE2's main value: control [38].

The steps that an enterprise has to follow in order to apply this combination are described bellow [38]. This combination emphasizes on simple design, because it is easier to schedule, the budget and to help the communication between the development team and the customers. Moreover, the enterprise should also have to apply an analysis phase, as it will be beneficiary in making more realistic and accurate designs. What is more, the enterprise should apply the XP features, such as continuous integration and testing, in order to reduce the risk probability. In addition, PRINCE2 requires very controlled procedures for testing and integration. Thus, it is necessary that all the procedures be properly documented. The enterprises should pay particular attention in the project assurance and the management. These two features should have grater focus on the quality plan in the beginning of the project and then to verify the production.

Finally, another major issue is the standardization. The project managers should concentrate on standard coding structure and naming conventions, as in an opposite case it is very difficult to control the quality.

The benefits that could emerge from this combination are described bellow:

- PRINCE2 heavily depends on paper work – almost every step of PRINCE2 is documented and filed. On the contrary, Agile methodologies and especially XP, rely on oral communications rather than documents [54][93].
- What is more, in XP there is no hierarchy, while PRINCE2 depends on a 4 – level hierarchical structure.
- PRINCE2 is considered to be as less flexible as any Agile methodology can ever tolerate, due to the fact that it requires a control method for every step in the lifecycle of the software project. Thus, XP requires the maximum possible flexibility in adopting the changes [38].

Nevertheless, this combination presents some drawbacks.

- PRINCE2 requires heavy bureaucracy.
- PRINCE2 is a project management method and supports the quality of the product; on the other hand, XP does not measure or plan the quality aspect of the software development. In order to overcome this, training is required for the development teams [38].

Despite the drawbacks, agility and discipline are not mutually exclusive. Rather, they can be complementary. Agility can contribute to creativeness and improve customer relationship, and discipline will keep the project on track and within budget, time, and quality constraints. Thus, there is no wrong answer for which management method to use in XP. PRINCE2 can be a complementary method, however it is preferred in unstructured environments [38].

### *d) CMMI and Scrum*

Through the SLR, the authors found that the CMMI maturity model and the Agile methodology Scrum can be compatible. CMMI focuses at a high level of abstraction, for example on what projects do, and not on what development methodology was used. Instead, Scrum focuses on how projects develop software products. Therefore, Scrum and CMMI can co – exist [94][95][96]. Scrum and CMMI together bring a more powerful combination of adaptability and predictability than either one alone, and suggest how enterprises can adopt them [95]. In addition, Scrum provides software development the "how – to", that is missing from CMMI. CMMI provides the systems engineering practices that help Scrum on large projects [94]. Furthermore, CMMI provides also the process management and supports practices that could help deploying, sustaining and continuously improving the deployment of Scrum in Small and Medium Enterprises [94].

Besides, this combination can substantially improve the dismal software project statistics, experienced by many companies [97]. In some cases, it has even also been used to migrate development staff to Agile development methods [87].

Mapping the Scrum practices into CMMI process areas have been the main way to combine these two methodologies. In order to make this mapping effective, few adaptations on Scrum practices have been made. These changes are mainly related to the Agile risk management, issues management and estimates methods [94][98].

The main ideas on how to apply this combination are described bellow [94][98]:

- CMMI is focusing on the enterprise. It is more beneficial when it is implemented in the organization level, so that all the processes of the development are addressed by the improvement effort. On the other hand, Scrum does not refer to the level of the organization.
- Scrum is not able to cover all the project management process areas. However, it can be tailored to be compliant with the CMMI. Furthermore, the CMMI's plan driven processes could be improved, by using Scrum practices.
- As far as the risk management is concerned, Scrum is able to identify the potential risks, but it does not

have any practices to identify the source, the parameters and to analyze and control the risk management effort. Thus, is does not provide any strategies to mitigate the risks. On the other hand, CMMI provides various strategies in order to confront potential risks. Therefore, particular attention should be paid on CMMI for the risk management.

- Finally, Scrum does not have any specific practices to address the areas of configuration management and quality assurance. CMMI however, provides mechanisms to support them. Specifically, it establishes and maintains the integrity of work, by using configuration control, identification and status. In addition, in order to assure the product quality, CMMI provides specific practices to evaluate the software development, the products and the services.

The benefits the this combination is able to provide are:

- The main benefit is that the best things of the two methodologies are adapted on the methodology, improvements such as better risk, issues and estimation assessment due to the CMMI principles [94][95].
- Adopting the CMMI, the combination also gains better-documented requirements, which leads to an improvement on the product quality [87][97].
- On the other hand, by applying Scrum, the communication inside the team and with the customer is improved, by being reflected on the enhancement of the performance [87][97].

The good news is that there are no specific drawbacks for this methodology. As happens in all the combinations the major drawbacks are the time that has to be invested in order to be able to get the best things from the methodology.

Concluding, projects that combine Agile methodologies with CMMI are more successful in producing higher quality software, which meets customers' needs at a faster pace [95].

### *e) CMM and Scrum*

CMM's purpose is to assure the quality of the software project. Furthermore, it minimizes the risk, as it is responsible to keep track of the development process of the project, and the development has to stick on the plans. Instead, Scrum is responsible to encourage the bold commitments and to provide benefits, such as establishing good communication between the software company and the clients – commit and deliver.

The most important selection criterion for this combination is the willingness of the consultants to interpret the CMM requirements from an Agile point of view. The freedom in this interpretation exists, since this combination refers to what enterprises should do and not how they should do it. Through this combination, it is ensured the quality of the project, the benefits of the communication and the maintaining of the sustainable pace of the development that the Agile methodologies can provide [99].

What is more, it is advisable for an enterprise in order to adopt this combination, firstly to apply the Scrum methodology to resolve in a quick way the different issues. Then they could apply the CMM to benchmark and measure the development process.

The benefits that could emerge from the combination of CMM and Scrum are some of the following [99]:

- The assessment teams can identify the possible strengths and weaknesses in the organization.
- The evaluation teams can identify the risks such as deadlines, quality of the product, contracts, and so forth and provide solutions to prevent them.
- Managers and staff can understand the necessary activities to plan and implement software process improvement for their organization.
- Cost reduction and cost accuracy in the software development process.
- Increase in the product quality and the productivity.

Although the benefits of this combination were analyzed before, however there are some drawbacks that should be taken into consideration. These drawbacks are discussed below [99]:

- CMM does not always specify a particular way of achieving the improvement goals, just because if one organization follows the rules, set by the CMM, it does not guarantee that it will be successful, as there are other factors involved.
- CMM says what you need and not how to do it. In addition CMM revers to processes and not to people.
- With this combination, it is difficult for the project manager to structure, organize and plan a project that lacks a clear definition.
- Frequent changes, frequent product delivery and uncertainty regarding the precise nature of the finished product make for a rather intense project life cycle for everyone involved.
- Training is required, so that the members of the enterprise can learn how to use the CMM and the Scrum. In an opposite case, if the members are not well equipped or committed, the project can even fail.

Despite the fact that the drawbacks of this combination are enough, the benefits that could emerge for an enterprise are more. First of all, this combination can be used in a Small and Medium – sized organization, with very small adjustments, such as creating smaller artifacts, and making the processes that the CMM demands simpler [101]. What is more, improvements in the quality of the software product and in the development cost have been noticed. Finally, proper usage of the development time has been performed [99].

### *f) CMMI and Lean*

Both approaches – CMMI and Lean – motivate the thinking in *"perfect lean processes"* and allow the use of a common terminology. Through the combination of these two methods, the authors came up with the following results [100][101]:

- LEAN allows seeing the "waste".
- CMMI has built in mechanisms to avoid the "waste".
- CMMI forms an enhanced toolbox to implement LEAN thinking in development / service / acquisition environments.

- CMMI provides a clear roadmap for process orientation both on project and organizational level.
- LEAN is supported by CMMI via the concept of institutionalization and organizational learning.
- Both methodologies require commitment and pro-active leadership on all management hierarchies.
- If done right, you are never done implementing either.

Both the authors tried to find more information regarding how to proceed to the combination, what would be the benefits, the drawbacks, potential challenges in the adoption and the impact that this combination can present in an enterprise. However, the literature that exists for this combination is very limited. They were able to identify only 4 papers, but the information was insufficient. In addition, they also tried the Google search engine, in order to check if there are examples of this combination in white papers, in presentations or in real enterprises.

From the entire search that has been performed in the 4 databases, in Google scholar, and finally in the Google search engine, they were not able to find the appropriate information for this combination. All the studies refer to the CMMI and Six Sigma, or the combination of Lean and Six Sigma [47][102][103], since these two methodologies present many similarities.

### *g) CMMI and Six Sigma*

CMMI is used to create an organizational process infrastructure by addressing particular domains, such as software and systems engineering [104][105]. Six Sigma is a top-down initiative that cuts across the entire enterprise, including areas such as engineering, sales, marketing, and research. Six Sigma is intended to be implemented with a focus on problems and opportunities, often with narrow scopes, that will yield significant business benefits [104][105]. It focuses on the performance of processes and practices as implemented rather than checking for compliance against a definition or model. While these two improvement initiatives are different by design, they are interdependent in their use. In practice, a back and forth focus is often effective [105][106]. For instance, Six Sigma could be used to discover the processes' needs to be more repeatable, CMMI could be used to institute processes based on community best practice, and then Six Sigma could be used to optimize those processes [105].

CMMI offers institutionalization features that are lacking in Six Sigma [107]. Six Sigma reinforces mission focus, and its enterprise deployment strategy fosters culture change that is supportive of CMMI implementation [101].

There are four different strategies that could be used to combine CMMI and Six Sigma. These strategies are described below [104][105][108]:

- *Implement CMMI process areas as Six Sigma projects.*

With this strategy, the objective of the Six Sigma development team is to implement a process area or a group of process areas. Their task is to define the problem or opportunity and to use the available data to inform the improvement or the design of processes that will serve the organizational mission and meet the model requirements.

- *Use Six Sigma as the tactical engine for high capability and high maturity.*

As far as the process definition is concerned, there is natural cooperation between the high maturity process areas and the Six Sigma's framework. As such, the tactics of Six Sigma can be used to directly enrich the defined processes that address the high maturity process areas.

- *Apply Six Sigma to improve or optimize an organization's improvement strategy and processes.*

Six Sigma could be used in making decisions about the adoption of improvement initiatives and in the management and overhead, associated with the adoption. In addition, using CMMI for guidance and possibly as governance for specific improvements, the organization could then employ Six Sigma for each improvement effort and push itself towards *"control"* and *"optimization"* one project at a time.

- *Integrate CMMI, Six Sigma, and all other improvement initiatives to provide a standard for the execution of every project throughout its life cycle.*

This is an approach for setting an organization's strategy. It is a longer term and more visionary. It supports the idea that an organization should take control of its destiny and manages its initiatives rather than be managed by them. Particularly, regardless of the label, the idea remains the same: the organization establishes a set of standard processes that incorporates all the features of the initiatives of choice. This idea assumes that conscious decisions have to be made at the organizational level to adopt these initiatives. Also it is assumed that the processes are adaptable with time, and robust to the realities of the organization.

As far as the benefits are concerned, they are described in the following bullets.

- CMMI and Six Sigma together, provide a strong foundation for performance – driven improvement.
- Six Sigma's focus can help to mitigate the risks of pursuing improvements. In addition, it also provides improvement frameworks and analytical methods that enable the achievement of the CMMI objectives [106].
- Six Sigma gives the organization a snapshot of the enterprise's current performance, which can be used as a roadmap towards future performance and improvement.
- On the other hand, CMMI's process infrastructure offers a foundation for Six Sigma's efforts. Furthermore, it helps an enterprise's engineering processes relate to its business processes [106][109].

Nevertheless, there are some drawbacks that have to be taken into consideration.

- Both of these models present a tendency to the minimum – avoid the hard stuff.
- They are designed for larger organizations and modifications have to be made, so that SMEs can use them [101].

Determining what is appropriate requires an understanding of the selected initiatives and their differences, synergies, and connections. CMMI and Six Sigma cannot subsume one another, because they are different types of models. Their joint deployment is synergistic. The potential value that is added is the accelerated achievement of performance goals, accelerated achievement of CMMI adoption, stronger foundational measurement and analysis skills to enable better quantification of results, and all of the corresponding culture change that goes along with these improvements [110].

### *h) PRINCE2 and DSDM*

The combination of DSDM with PRINCE2 seems that the two methodologies are complementary. PRINCE2 can provide control and DSDM can provide agility. This combination could be found in concepts like product-based planning, involved partnership of users and developers, and strong emphasis on underlying business case [38].

The combination of PRINCE2 and DSDM seems to be a safe approach, since these two models have many things in common [38][111].

The DSDM with the PRINCE2 Task Group [111] see the two methodologies as complementary. PRINCE2 can provide control and DSDM can provide agility. Moreover the paper [111] claims that DSDM developers had PRINCE2 in mind. This could be found in concepts like product-based planning, involved partnership of users and developers, and strong emphasis on underlying business case.

PRINCE2 is a project management method, intended for all types of project, whereas DSDM is a rapid application development method. For an organization using PRINCE2 for IT, ensures commonality between DSDM and other types of projects.

An enterprise in order to apply this combination has to follow a series of steps [111]:

- In the initial phase of the project development, the early stages of the PRINCE2 overlap the stages of the DSDM. In addition, both methodologies have a major control point in the end of the initiation. In this point, a decision to proceed must be confirmed, and the option of abandoning the project should be considered.
- In the running phase of the development, PRINCE2 does not require any management stages to match the technical ones. Furthermore, a management stage could be consisted of a number of DSDM time boxes. The number of the required stages should be determined by the ratio of the needed management control towards the potential overhead.
- In the end of the project, the PRINCE2 close down overlaps with the implementation phase of the DSDM. In addition, the project review is done in every increment of the DSDM and is related to the End Stage Assessment of PRINCE2, which is the last procedure on the project.

Through the combination of PRINCE2 and DSDM, the benefits that could emerge are some of the following [112]:

- Good communication between the project team and other stakeholders.
- Mechanisms to handle with deviations to the project plan.
- Flexible decision points.
- Prioritization being clearly defined, and performed early in the project.
- Quick visibility of the development process and handling of potential problems.
- The use of both models – PRINCE2 and DSDM is for free.
- On-cost and on-time delivery of the software projects.
- Time boxing to keep the project on track. This simplifies the use of tolerance. The only tolerance used extensively is scope, and this is flexed under the control of the empowered business representatives. This gives the business what they often need the most – on time and on budget delivery of a product which meets the business objective;
- Delivery of business products during the project, not just at the end;
- Welcomes changing requirements, even late in the project, using prioritization and time boxing to control this within time and budget, to harnesses change for the customer's competitive advantage.
- Small teams with empowered user representatives as fully resourced and continuous team members.
- Facilitated workshops and face-to-face communication, minimizing documentation wherever possible.

Although the benefits that could come up through such a combination are enough, however, there are some drawbacks, such as [38]:

- There is little information about this combination on the industry.
- Bureaucracy is needed, although DSDM as an Agile technique uses workshops, and face-to-face communication.

In a world where speed of delivery is often more important than having 100% of the functionality and where projects have to deliver within the time and budget constraints, to take advantage of market opportunity or to comply with complicated requirements, DSDM delivers [113].

Additionally, in an environment where many organizations are constrained to demonstrate that they are controlling their projects effectively and that they are giving the best value for money, PRINCE2 performs [113].

Since both of these needs often run concurrently, the use of PRINCE2 for its control and DSDM for its agility is a powerful combination. What is more, their combination produces a result, where the whole is greater than the sum of the constituent parts [111][113].

### *i) Agile methodologies and SPI maturity models on SMEs*

In the previous section the different parts in the combination of the Agile methodologies and the SPI maturity models were analyzed. The authors tried to find information regarding these combinations for SMEs, however the results they found were very limited. Although Small and Medium – sized firms represent a high portion of the enterprises all over the world, most of the SPI maturity models are designed for large or very large enterprises [17].

The continuous improvement of the software processes is important in enhancing the capabilities of an organization. The condition for change can be described properly in the following equation [114]:

If $D * V * F > R$<br>
Then "Change will occur"<br>
Where<br>
D: Dissatisfaction with status quo.<br>
V: Vision of a future state.<br>
F: First steps towards the vision.<br>
R: Resistance to change.

If the organizations follow this equation, they can improve their software engineering practices. However, the traditional approaches for the SPI need to be altered, so that the co-existence of the Agile methodologies and the SPI could be enabled [86].

What is more, SMEs suffer from the lack of research studies to solve the problem of improving their software development processes [115]. Therefore, from recent studies [17][38][84] the authors found out that currently organizations and especially SMEs are increasingly using Agile methodologies – most common methodologies are XP, Scrum, and DSDM – and SPI maturity models – most common maturity models are CMM, CMMI, and PRINCE2 – in their software projects. However, a large scale and systematic adoption of both Agile methodologies and SPI maturity models, is quite difficult, since they are mainly focused on project level activities [86].

#### *j) Conclusion*

In the Systematic Literature Review the authors tried to identify the most common Agile methodologies, the most common SPI maturity models and especially the most common combinations regarding Agile methodologies and SPI maturity models that could be applied in SMEs. They identified 1508 studies from the e-databases, from which 71 were relevant for the topic.

From the research that was performed, the authors found that the most common Agile methodologies are the Scrum, the XP and the DSDM. Additionally, the most common SPI maturity models are the CMM, the CMMI, the ISO / IEC 15504 and the PRINCE2. Finally, the most common combinations of Agile methodologies and SPI maturity models are the CMMI and XP, the CMMI and Six Sigma, the CMMI and Scrum and PRINCE2 and DSDM.

What is more, there are various benefits that could emerge from these combinations. All of them plan for the quality increase, for the minimization of the risk, for the on-cost and on-time development, for better communication between managers, development team and customers, and for clear visibility in the software development. Moreover, these models are frameworks, and they provide instructions for the management and the development. In addition, they provide mechanisms, such as small artifacts, documentation, control processes and others, in order to avoid possible deviations from the project plans.

To summarize, the results show that the SPI standards could fit with Agile methods, and that it is better if organizations can embrace both, since they benefit each other (by for example establishing better communication in the enterprise, reduce the bureaucracy and focus on the communication, avoid the large projects phases and create smaller artifacts) and bring more customer satisfaction, as well as reducing waste and creating higher software quality and higher value creation. Furthermore, although larger organizations and Small and Medium sized – firms present significant differences, the authors presented that the combinations with small adjustments [8][116] such as reduce the documentation and create smaller artifacts, could be used in SMEs.

In the next section, the structure and the results of the survey are analyzed.

## IV. SURVEY

### *A. Survey study design*

This section aims to answer **RQ2: "Under which situations and how could these methodologies be applied? What are the benefits and drawbacks that each methodology could provide?** and **RQ3: "Are these methodologies really applied in SMEs?"** through a survey.

The survey was consisted of two parts; questionnaire and interviews. Respondents were given the questionnaire in the form of a link to the website with the online version [6]. After the questionnaire responses, the authors performed interviews either face to face or through Skype. In total, **49 respondents participated in the survey.**

### *B. Sample*

The intended audience of the survey was practitioners who are working with Agile methodologies and Software Process Improvement maturity models. The authors were based on their personal contacts and through the Internet (e.g. LinkedIn) in order to find a suitable sample, including experienced practitioners in the field of Software Engineering. They conducted the survey from June 2013 until October 2013.

**49 practitioners** answered the questionnaire and from them 26 were project managers, 20 were software engineers and 3 were SMEs' owners. As far as the interviews are concerned, **13 practitioners** participated; 10 were project managers and 3 were software engineers. All of them had more than 10 years experience in the fields of project management and software engineering.

What is more, the survey was conducted in four European countries – Greece, Spain, France and the UK. It could be claimed that the results of the survey could be representative, because not only the authors gathered results from these different countries, but also they included participants with different backgrounds and positions in various enterprises.

Finally, the majority of the participants (almost 90%) work in Small and Medium – sized firms, and more than half of them work in enterprises that exist for more than 10 years.

The questions of the questionnaire are presented in **Appendix C**.

An analysis of the questionnaire and interview results is presented in the following subsections.

*C. Questionnaire Results*

**<u>Question 4:</u>** The fourth question asked about the most common Agile methodologies.

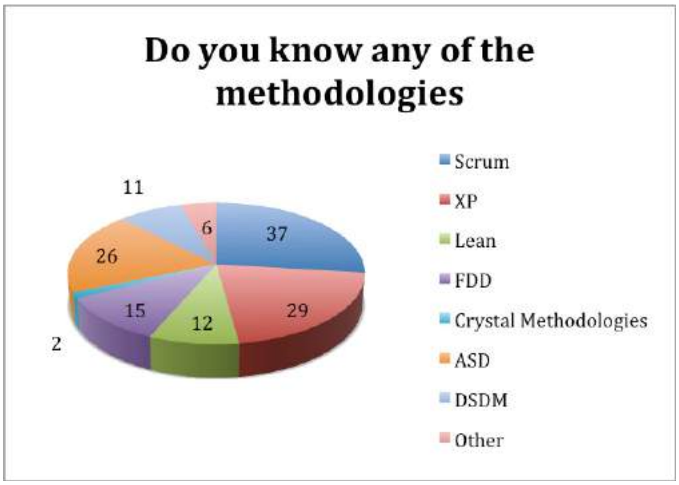

Figure IV. 1 Most common Agile methodologies

As presented in **Fig. IV.1**, the most common Agile methodologies are the Scrum, the XP and the Adaptive Software Development.

**<u>Question 5:</u>** This question refers to the time that the participants have used the Agile methodologies.

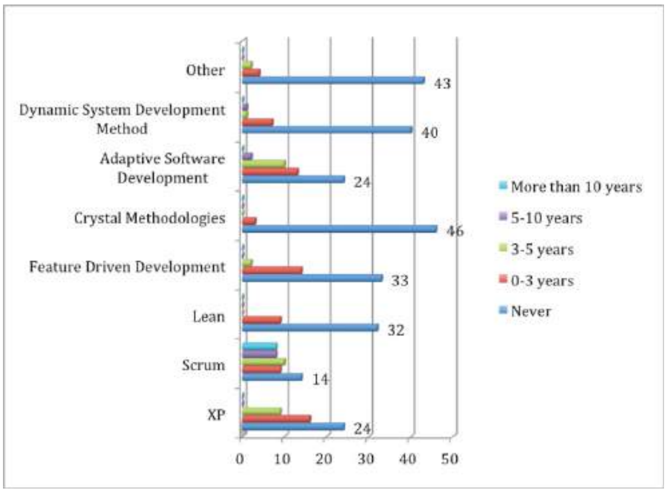

Figure IV. 2. Time of applying the most common Agile methodologies

As seen in the figure presented above, the majority of the participants have never used the Crystal methodologies, the Dynamic Systems Development Method and the Feature Driven Development. Instead, they have used the XP for more than 3 years, the Scrum for more than 5 years and the Adaptive Software Development for more than 5 years as well. The reader can easily observe that the most common Agile methodology to the participants is the Scrum model.

**<u>Question 6:</u>** This question refers to the results of applying Agile methodologies.

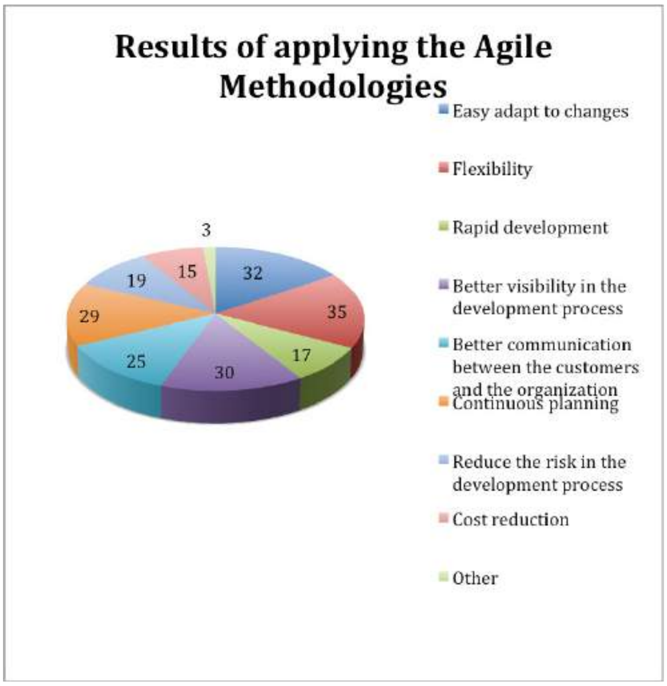

Figure IV. 3 Results of applying Agile methodologies

In the picture presented above, the results of applying the most Agile methodologies are analyzed. They present flexibility and better visibility during the software development process, they adopt easily the changes, and finally they enhance the communication between the customers or other stakeholder and the enterprise.

**<u>Question 7:</u>** This question asked the practitioners about how familiar they are with the most common SPI maturity models. In **Fig IV.4**, the reader can notice that for the participants the most common SPI maturity models are the CMMI, the CMM and the ISO / IEC 15504.

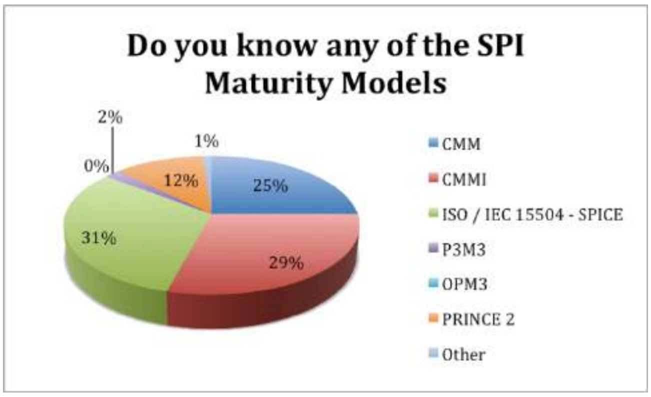

Figure IV. 4. Most common SPI maturity models

**<u>Question 8:</u>** This question refers to the time that the participants have used the most common SPI maturity models.

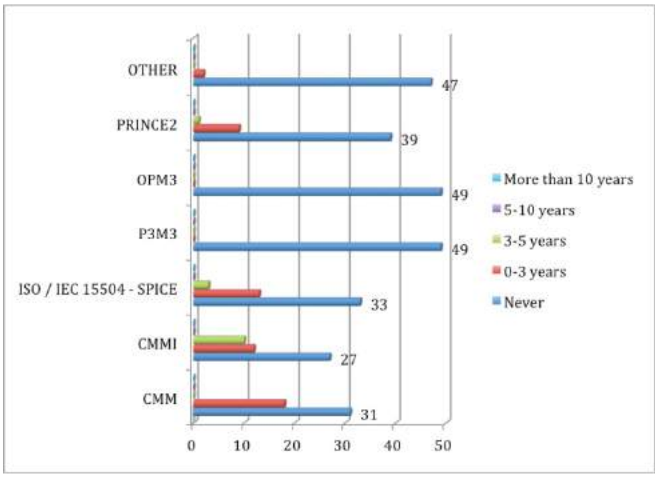

Figure IV. 5 Time of applying the most common SPI maturity models

As seen in **Fig. IV.5**, almost half of the participants have never used any of the SPI maturity models. However, around 10 participants have used the ISO / IEC 15504, the PRINCE2 and the CMMI from months to 3 years, while 10 have used the CMMI for more than 3 years. As far as the CMM is concerned, 18 participants have used is however, for less than 3 years time.

Special attention has to be paid to the fact that none of the participants have ever used the OPM3 and the P3M3.

**Question 9:** This question refers to the possible results that the SPI maturity models could present. In the following figure, it can be noticed that the participants claimed that the SPI maturity models offer better control and visibility during the development process. Additionally, they enhance the quality improvement of the software product and the development is performed within the time constraints.

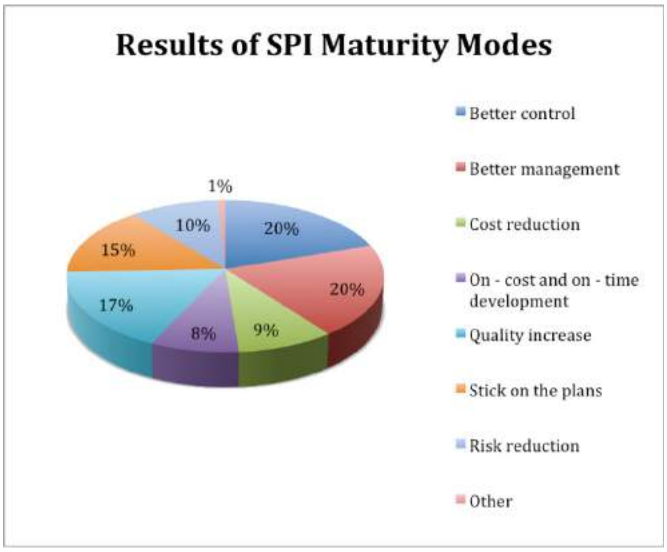

Figure IV. 6. Results of the SPI maturity models

**Question 10:** This question asked the practitioners about the most common combinations of Agile methodologies and SPI maturity models.

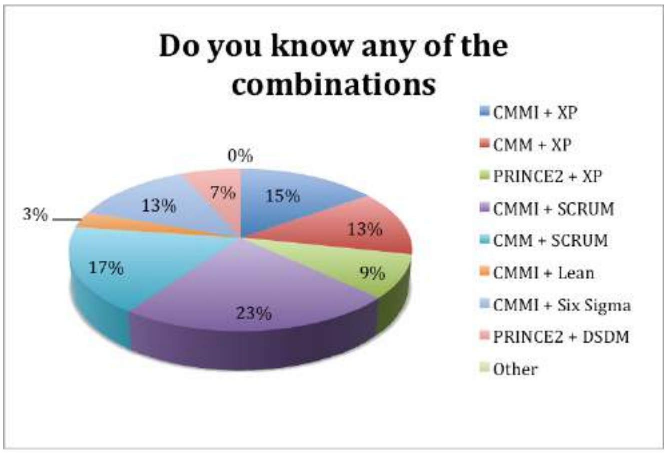

Figure IV. 7. Most common combinations of Agile methodologies and SPI maturity models

In **Fig. IV. 7**, the reader could notice that the most common combination according to the participants is the CMMI and Scrum. The combinations that follow are the CMM and Scrum, the CMMI and XP, the CMMI and Six Sigma and the CMM and XP.

**Question 11:** This question refers to the time that the participants have used the combination of Agile methodologies and SPI maturity models.

In the following picture **Fig IV. 8**, it can be easily observed that the most common combinations are the CMM and Scrum and the CMMI and Scrum. Special attention has to be paid, as the majority of the survey participants have never used any of the most common combinations.

Nevertheless, around 8 participants have used the combinations of CMMI and Six Sigma, CMM and Scrum, CMM and XP and CMMI and XP for time less than 3 years, while 13 have used the CMMI and Scrum from months to 3 years.

Additionally, around 5 participants have used the combinations of CMMI and Six Sigma, CMM and Scrum and CMMI and Scrum for time less than 5 years.

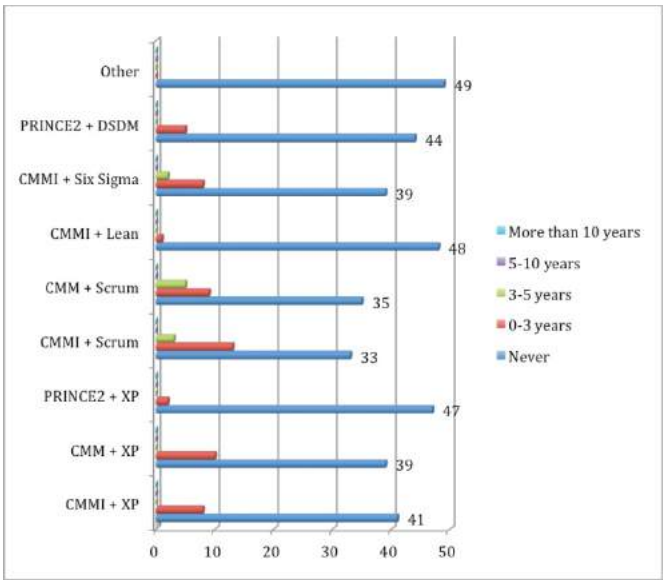

Figure IV. 8. Time of applying the combination of Agile methodologies and SPI maturity models

**Question 12:** This question refers to the possible results that the combinations of Agile methodologies and SPI maturity models could emerge. As presented in **Fig. IV.9**, the participants claimed that the combinations enhance the communication between the customers, managers and the development team; they increase the quality of the software product, the development is performed within the time and cost constraints and there is better visibility, management and control of the development process.

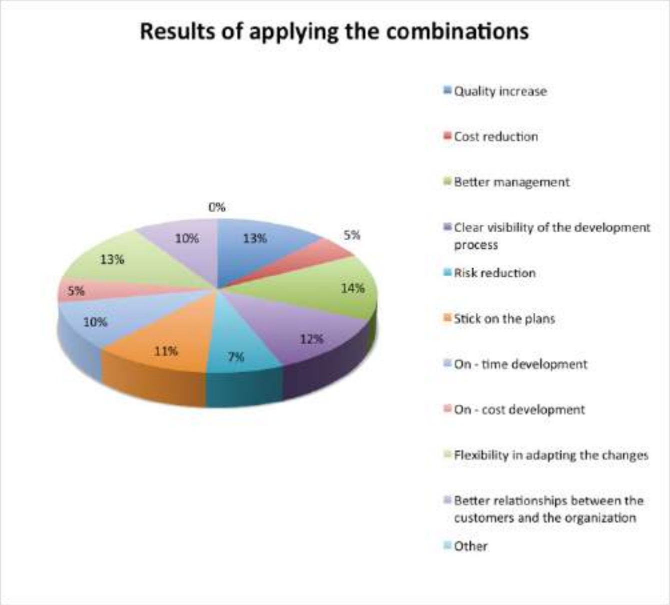

Figure IV. 9. Results of applying the combinations of Agile methodologies and SPI maturity models

**Questions 13 and 14:** These questions asked the practitioners if they have any other suggestions or proposal for improving the software development and if they have every used any other methodologies that the authors do not mention. The answers that they provided are the following:

- Firstly decide the methodology you are going to use, and the start developing the code.
- Establish good communication between the managers, the development team and the customers. In addition, the requirements of the product should be clear to all the involved parties.
- Focus early on the development process, and then provide tight control of the process.
- There is not only a correct methodology, since there are too many methods. Different combinations could bring different results.
- Agile is considered to be the best practice now a days, which makes the software development process fast, flexible regarding the change and on time. While one thing that is going down is quality, and quality can be improved by using CMMI or ISO standards, to control the quality of software in parallel.
- Provide training and motivation for learning. Change of responsibilities and roles, in order to help the members of the team to have an overall view for the different parts of the product.

### D. Interviews

The type of the interviews that were conducted was semi-structured, since the authors had already formulated the questions in advance. However, they wanted to offer freedom to the participants to express their ideas and be more sociable.

What is more, the duration of each interview was approximately 30 minutes. Nevertheless, depending to their free time and the answers they were providing the authors, this time limit could be adjusted. During the time of the interviews, the authors were keeping notes, in order not to miss the key parts of each response. They did not want to record the interviews, as the interviewees might not feel free to express themselves and provide wrong responses. After the interviews, the authors were creating a small document trying to answer all the questions asked, with details.

As far as the Agile methodologies are concerned, the authors found out that the participants are familiar with the most common Agile methodologies such as the Scrum, the XP, the Adaptive Software Development, the Feature Driven development and the Dynamic System Development Methodology. They have used the Scrum, the XP and the ASD for time between 3 and 5 years in various projects. In addition, they have used all the other methodologies for at least one time, however for less than 3 years.

To continue with the main Software Process Improvement maturity models, the interviewees know the CMM, the CMMI, the ISO / IEC 15504 and the PRINCE2.

Besides the Agile methodologies, the practitioners have used these maturity models mentioned above, in various projects for time less than three years; only a few participants have used the CMMI, the SPICE and the PRINCE2 for more than 3 years. However, the percentage of usage of the P3M3 and the OPM3 tends to be zero.

As far as the combination of the Agile methodologies and the SPI maturity models is concerned, the answers the authors received are significant lower. Only a small portion of the interviewees are familiar with the main combinations such as the CMM and XP, CMMI and XP, the CMM and Scrum, the CMMI and Scrum, the CMMI and Six Sigma, the PRINCE2 and DSDM and the PRINCE2 and XP.

According to the practitioners, the benefits that have been observed through the usage of these combinations is the quality increase and the on – cost and on – time development. In addition, there is better communication between the customers, the enterprise and the development team and therefore, the customer satisfaction is increased. Finally, there is clear visibility and better management of the development process.

It is worth to mention that the surveyed mentioned that for most of the combinations the information that exists in literature and in real examples on the industry is very limited. What is more, the combinations are again addressed for larger enterprises and as a result the project managers had to make adjustments as mentioned before, such as creating smaller artifacts and try to make the processes that the combinations demand simpler. Furthermore, training is required for the development team, the license of the SPI maturity models is not always for free and the enterprise had to change its culture.

Concluding, the interviewees mentioned that in order for an enterprise to be successful there should be established good communication between the managers and the development team. Moreover, they have noticed from their experience that firstly the have to decide which methodology or combination they are going to use and then start developing the code. They sum up that there is not a right or wrong methodology. Instead, it is advisable for an enterprise to follow the methodology or the combination that they are more familiar with, in order to achieve the maximum results.

### *E. Conclusion*

Agile methodologies such as the Scrum, the XP, the Adaptive Software Development and the Feature Driven Development are very familiar to the participants. Besides the Agile methodologies, the majority of the participants are familiar with the main SPI methodologies, such as the CMM, the CMMI, the SPICE and the PRINCE2. However, only one third of the participants are familiar with the combinations of Agile and SPI. They mention that they know the majority of the combinations; nevertheless, they have mostly used the combinations that include the CMM, the CMMI and the Scrum for more than three years.

What is more, the combinations provide a series of benefits such as better communication between customers, managers and clients, quality increase, better visibility and control of the development process and finally, development of the project within the time and budget constraints. Besides the benefits, the drawbacks that could emerge have to be taken into consideration. Most of the SPI models are not for free and their licenses are quite expensive. Finally, and skilled personnel is required as the processes that these combinations demand can become very complex.

## V. Threats to Validity

There are different ways to classify aspects of validity and threats to validity in the literature. In this report we decided to follow four aspects of validity and specifically the construct, the internal, the external and finally the conclusion validity according to Runeson et al. [117]. In addition, countermeasures against threats to validity were then taken.

As far as the construct validity is concerned, it refers to what extend the inferences actually represent the research questions [117]. In order to confront with the construct validity, both the researchers identified together the keywords, and formatted the search strings. Additionally, they designed together the questions for the questionnaire and the interviews. As each researcher performed different interviews, both of them performed rehearsals, so that they could interpret the interviews in the same way.

To continue with the internal validity, it refers to the determination of cause- and- effect relationships [117]. To deal this type of validity, both the researchers obtained good knowledge of the domain. They performed individual searches in literature and as well as individual interviews, in order to avoid the bias in the results. What is more, especially in the survey, the researchers provided the same information to all the participants, in order to become familiar with the topic of the survey and all of them should have a common understanding. Finally, the search was performed in a short period of time, as due to the evolution in the field of software engineering, new models could be created.

Besides the internal validity, external validity was also used. It refers to what extend the extent it is possible to generalize the findings, and to what extent the findings are of interest to other people outside the investigated case [117]. For this thesis, the researchers performed an SLR and a survey, including 71 articles and 46 participants from four European countries, Greece, Spain, France and the UK, as mentioned before. Furthermore, the majority of the participants are project managers that work in Small and Medium – sized firms and they have more than 10 years of experience in the field of Software Engineering.

The last type of validity that was used is the conclusion validity. It focuses on how sure we can be that the treatment the authors used in an experiment is actually related to the actual outcome they observed. The researchers, before conducting the survey, they had already performed a Systematic Literature Review regarding the same topic. The results that came up from this survey were compared with the results from the survey in order to see if there is an actual relationship between the literature and the "real" examples from the industry field.

Finally, it is worth to mention, that triangulation was achieved in different ways, both the researchers reviewed the results of the Literature Review and the survey. Additionally, both the researchers were working with the same material in parallel, in case they identified potential wrong answers or results. Concluding, as far as the survey is concerned, it was also seen as important that the majority of the participants were familiar with the researchers.

## VI. Conclusion

The overall goal of the thesis was to understand how the combination of Agile methodologies and SPI maturity models could be applied in SMEs, what are the possible benefits and drawbacks that could emerge and if they are really applied in enterprises, as presented in literature. The main aim of the combinations is that they try to extract the best parts of each Agile methodology and SPI maturity model, in order to create a new methodology containing all these key concepts. What is more, each methodology has to be transformed, in order to cover the need of each enterprise.

To understand all these mentioned above, the authors created 3 research questions. In the following section, a short summary of how each question was answered is presented.

**RQ1:** *"What Agile methodologies combined with Software Process Improvement maturity models crafted for Small and Medium enterprises exist?"*

In order to provide answers to this research question, an SRL has been performed. Through this study, the most common combinations that the authors came up are the following:

- CMMI and XP,
- CMM and XP,
- PRINCE2 and XP,
- CMMI and Scrum,
- CMM and Scrum,

- CMMI and Lean,
- CMMI and Six Sigma,
- PRINCE2 and DSDM.

**RQ2:** *"Under which situations and how could these methodologies be applied? What are the results, benefits and potential drawbacks that each methodology could present? When should each methodology be applied? What are the reasons for failure?"*

The authors in order to answer this question were based both on the SLR and survey. As analyzed in section III.3 each combination could be applied in a different ways and presents different benefits and drawbacks. The authors found that the enterprises usually follow firstly the Agile methodology and then they proceed to the adoption of the SPI maturity model.

The general benefits that could emerge almost in every combination the combinations are increase in the quality of the software project, better management and visibility during the development process.

Nevertheless, the SPI maturity models are not always for free and they are addresses to larger organizations. In addition qualified personnel is required. However, it has been proved that if SMEs perform small adjustments, such as such as creating smaller artifacts or making the processes that the methodologies demand, simpler, they will be able to adopt these combinations.

**RQ3:** *"Are these methodologies really applied in SMEs?"*

The answers for this research question came through the survey that the authors performed. The results of the survey show that although Small and Medium – sized firms face serious difficulties in adopting these methodologies, by performing small adjustments, they could be able to adopt the combinations mentioned above.

The combinations that are really applied are the ones who involve the CMM, the CMMI and the Scrum. Specifically, the CMMI and XP, the CMM and XP, the CMM and Scrum, the CMMI and Scrum and the CMMI and Six Sigma.

From all the studies that were used for this thesis it can be said that applying the combination between different methodologies would benefit the software development process. As Agile methodologies and SPI maturity models are applied differently, depending on the environment where they are applied, the obtained benefits from applying the combination will vary from one case to another. Still, it can be assured that there will be benefits, since the idea of the combination is to take the practices that can be "interesting" for giving an extra value to the software process creation.

Concluding, as it was mentioned before, all the papers selected for the study were peer reviewed. Apart of that, the research methodology used in the paper, how it was applied and the validity of the results was checked by both the authors. What is more, during this paper, the authors did not identify any studies referring to negative results by applying the combinations. This happens as a result of what it was mentioned before; getting bad results from the combination is not really possible, and if it happens, it will be because the combinations were not performed properly.

ACKNOWLEDGMENTS

The authors would like to express their highest appreciation to their supervisor Dr. Stefanie Betz for her guidance, encouragement and support during the development of their dissertation thesis. Her feedback and advices were more than valuable for the authors and were leading their efforts.

Special thanks to all the lectures from Blekinge Institute of Technology in Karlskrona, for their guidance throughout the Software Engineering Master's Degree modules.

Last but not least, the authors would like to thank, their families and friends for their excessive support and encouragement during the year.

## APPENDIX A – QUESTIONS FOR QUALITY ASSESSMENT

The questions that were used for the quality assessment of the selected studies are presented bellow.

1. Is there a clear statement of the aims of the research methodology used:
   a. SLR?
   b. Case Study?
      i. Exploratory study?
      ii. Action research?
      iii. Descriptive research?
   c. Comparative Research?
   d. Experimental Research?
   e. Survey?
   f. Other type of research?
2. Is there an adequate description of the context in which the
   a. SLR?
   b. Case Study?
      i. Exploratory study?
      ii. Action research?
      iii. Descriptive research?
   c. Comparative Research?
   d. Experimental Research?
   e. Survey?
   f. Other type of research?
3. Was the research design appropriate to address the aims of the
   a. SLR?
   b. Case Study?
      i. Exploratory study?
      ii. Action research?
      iii. Descriptive research?
   c. Comparative Research?
   d. Experimental Research?
   e. Survey?
   f. Other type of research?
4. Was there a control group with which to compare treatments?
5. Was the research methodology applied properly, in terms of type of the study, data analysis, etc.?
6. Was the data collected in a sufficiently rigorous way that addressed the research issue?
7. Is there a clear statement of findings?
8. Do the conclusions relate to the research aim?

9. Do they discuss about limitations in the research?
10. Does the paper / study discuss areas for future research?

## Appendix B – List of selected articles for the SLR

| # | Title |
|---|---|
| 1 | Cohen, David, Mikael Lindvall, and Patricia Costa. "Agile Software Development." *Data & Analysis Center for Software (DACS), New York* (2003). Doi=10.1.1.201.270 |
| 2 | Dybå, Tore, and Torgeir Dingsøyr. "Empirical Studies of Agile Software Development: A Systematic Review." *Information and Software Technology* 50, no. 9–10 (August 2008): 833–859. Doi:10.1016/j.infsof.2008.01.006. |
| 3 | Begel, A., and N. Nagappan. "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study." In *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*, 255–264, 2007. Doi:10.1109 |
| 4 | Sison, R., and T. Yang. "Use of Agile Methods and Practices in the Philippines." In *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific*, 462–469, 2007. Doi:10.1109 |
| 5 | Dyba, Tore, and T. Dingsoyr. "What Do We Know About Agile Software Development?" *IEEE Software* 26, no. 5 (2009): 6–9. Doi:10.1109/MS.2009.145. |
| 6 | De Souza Carvalho, W.C., P.F. Rosa, M. dos Santos Soares, M.A. Teixeira da Cunha Junior, and L.C. Buiatte. "A Comparative Analysis of the Agile and Traditional Software Development Processes Productivity." In *Computer Science Society (SCCC), 2011 30th International Conference of the Chilean*, 74–82, 2011. Doi:10.1109/SCCC.2011.11. |
| 7 | Jha, R., and A.K. Saini. "ERP Redefined: Optimizing Parameters with Lean Six Sigma for Small #x0026; Medium Enterprises." In *2011 International Conference on Communication Systems and Network Technologies (CSNT)*, 683–687, 2011. Doi:10.1109/CSNT.2011.147. |
| 8 | Flores, M., A. Cabello, L. Torredemer, M. Agrawal, J. Keast, S. Terzi, and A. Sopelana. "Do Enterprises Implement a Process Architecture Towards Lean in Product Development? A Comparative Study Among Large and Small Firms." In *2011 17th International Conference on Concurrent Enterprising (ICE)*, 1–9, 2011. |
| 9 | Nawrocki, J., B. Walter, and A. Wojciechowski. "Toward Maturity Model for Extreme Programming." In *Euromicro Conference, 2001. Proceedings. 27th*, 233–239, 2001. Doi:10.1109/EURMIC.2001.952459 |
| 10 | Buglione, Luigi. "Light Maturity Models (LMM): An Agile Application." In *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement*, 57–61. Profes '11. New York, NY, USA: ACM, 2011. Doi:10.1145/2181101.2181115. |
| 11 | Fitzgerald, Brian, Klaas-Jan Stol, Ryan O&#039;Sullivan, and Donal O&#039;Brien. "Scaling Agile Methods to Regulated Environments: An Industry Case Study." In *Proceedings of the 2013 International Conference on Software Engineering*, 863–872. ICSE '13. Piscataway, NJ, USA: IEEE Press, 2013. |
| 12 | Mahfouz, Amr, John Shea, and Amr Arisha. "Simulation Based Optimisation Model for the Lean Assessment in SME: a Case Study." In *Proceedings of the Winter Simulation Conference*, 2408–2418. WSC '11. Phoenix, Arizona: Winter Simulation Conference, 2011. http://dl.acm.org/citation.cfm?id=2431772.2431806. |
| 13 | Jha, Rashmi, and Anil Kumar Saini. "An Exploratory Factor Analysis on Pragmatic Lean ERP Implementation for SMEs." In *2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, PDGC 2012, December 6, 2012 – December 8, 2012*, 474–479. Proceedings of 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, PDGC 2012. IEEE Computer Society, 2012. Doi:10.1109/PDGC.2012.6449867. |
| 14 | Achanga, Pius, Essam Shehab, Rajkumar Roy, and Geoff Nelder. "A Fuzzy-logic Advisory System for Lean Manufacturing Within SMEs." *International Journal of Computer Integrated Manufacturing* 25, no. 9 (2012): 839–852. Doi:10.1080/0951192X.2012.665180. |
| 15 | Dombrowski, U., I. Crespo, and T. Zahn. "Adaptive Configuration of a Lean Production System in Small and Medium-sized Enterprises." *Production Engineering* 4, no. 4 (2010): 341–8. Doi:10.1007/s11740-010-0250-5. |
| 16 | Thomas, A., R. Barton, and C.C. Okafor. "Applying Lean Six Sigma in a Small Engineering Company : a Model for Change." *Journal of Manufacturing Technology Management* 20, no. 1 (2009): 113–29. Doi:10.1108/17410380910925433. |
| 17 | Ribeiro, F.L., and M.T. Fernandes. "Exploring Agile Methods in Construction Small and Medium Enterprises: a Case Study." *Journal of Enterprise Information Management* 23, no. 2 (2010): 161–80. Doi:10.1108/17410391011019750. |
| 18 | Abad, Zahra Shakeri Hossein, Mahsa Hasani Sadi, and Raman Ramsin. "Towards Tool Support for Situational Engineering of Agile Methodologies." In *17th Asia Pacific Software Engineering Conference: Software for Improving Quality of Life, APSEC 2010, November 30, 2010 – December 3, 2010*, 326–335. Proceedings – Asia-Pacific Software Engineering Conference, APSEC. IEEE Computer |

Society, 2010. Doi:10.1109/APSEC.2010.45.

19 Barafort, Béatrix, Bernard Renzo, and Olivier Merlan. “Benefits Resulting from the Combined Use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL).” In *Product Focused Software Process Improvement*, edited by Markku Oivo and Seija Komi-Sirviö, 2559:314–325. Berlin, Heidelberg: Springer Berlin Heidelberg. Accessed April 5, 2013. http://link.springer.com.miman.bib.bth.se/content/pdf/10.1007%2F3-540-36209-6_27.

20 Unterkalmsteiner, M., T. Gorschek, A.K.M.M. Islam, Chow Kian Cheng, R.B. Permadi, and R. Feldt. “Evaluation and Measurement of Software Process Improvement #x02014;A Systematic Literature Review.” *IEEE Transactions on Software Engineering* 38, no. 2 (2012): 398–424. Doi:10.1109/TSE.2011.26.

21 Tanovic, A., and F. Orucevic. “Integration of PRINCE2 Model into ITIL V3 Model.” In *Telecommunications Forum (TELFOR), 2011 19th*, 102–105, 2011. Doi:10.1109/TELFOR.2011.6143503.

22 Pino, Francisco J., Félix García, and Mario Piattini. “Software Process Improvement in Small and Medium Software Enterprises: a Systematic Review.” *Software Quality Journal* 16, no. 2 (June 1, 2008): 237–261. Doi:10.1007/s11219-007-9038-z.

23 Suwanya, S., and W. Kurutach. “An Analysis of Software Process Improvement for Sustainable Development in Thailand.” In *8th IEEE International Conference on Computer and Information Technology, 2008. CIT 2008*, 724–729, 2008. Doi:10.1109/CIT.2008.4594764.

24 Alexandre, S., A. Renault, and N. Habra. “OWPL: A Gradual Approach for Software Process Improvement In SMEs.” In *32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006. SEAA ’06*, 328–335, 2006. Doi:10.1109/EUROMICRO.2006.48.

25 Zhang, Lina, and Dan Shao. “Software Process Improvement for Small and Medium Organizations Based on CMMI.” In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2402–2405, 2011. Doi:10.1109/AIMSEC.2011.6011045.

26 Garcia-Mireles, Gabriel Alberto, Ma Angeles Moraga, and F. Garcia. “Development of Maturity Models: A Systematic Literature Review.” In *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, 279–283, 2012. Doi:10.1049/ic.2012.0036.

27 Khoshgoftar, M., and O. Osman. “Comparison of Maturity Models.” In *2nd IEEE International Conference on Computer Science and Information Technology, 2009. ICCSIT 2009*, 297–301, 2009. Doi:10.1109/ICCSIT.2009.5234402.

28 Ciolkowski, M., and M. Soto. “Towards a Process Maturity Model for Open Source Software.” In *Computer Software and Applications, 2008. COMPSAC ’08. 32nd Annual IEEE International*, 1213–1214, 2008. Doi:10.1109/COMPSAC.2008.47.

29 Cater-Steel, A.P. “Process Improvement in Four Small Software Companies.” In *Software Engineering Conference, 2001. Proceedings. 2001 Australian*, 262 –272, 2001. Doi:10.1109/ASWEC.2001.948520.

30 Boas, G.V., A.R.C. da Rocha, and M. Pecegueiro do Amaral. “An Approach to Implement Software Process Improvement in Small and Mid Sized Organizations.” In *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on The*, 447–452, 2010. Doi:10.1109/QUATIC.2010.77.

31 Habib, M., S. Ahmed, A. Rehmat, M.J. Khan, and S. Shamail. “Blending Six Sigma and CMMI – an Approach to Accelerate Process Improvement in SMEs.” In *Multitopic Conference, 2008. INMIC 2008. IEEE International*, 386–391, 2008. Doi:10.1109/INMIC.2008.4777768.

32 Scott, L., Ross Jeffery, L. Carvalho, J. D’Ambra, and P. Rutherford. “Practical Software Process Improvement – the IMPACT Project.” In *Software Engineering Conference, 2001. Proceedings. 2001 Australian*, 182–189, 2001. Doi:10.1109/ASWEC.2001.948512.

33 Jezreel, M., M. Mirna, N. Pablo, O. Edgar, G. Alejandro, and M. Sandra. “Identifying Findings for Software Process Improvement in SMEs: An Experience.” In *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2012 IEEE Ninth*, 141–146, 2012. Doi:10.1109/CERMA.2012.30.

34 Pino, F.J., F. Garcia, and M. Piattini. “A Support Tool for Rapid Software Process Assessment.” *Latin America Transactions, IEEE (Revista IEEE America Latina)* 5, no. 4 (2007): 218–223. Doi:10.1109/TLA.2007.4378509.

35 Sharma, B., N. Sharma, and N. Sharma. “Software Process Improvement: A Comparative Analysis of SPI Models.” In *2009 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET)*, 1019–1024, 2009. Doi:10.1109/ICETET.2009.206.

36 Pino, Francisco J., Félix Garcia, and Mario Piattini. “Key Processes to Start Software Process Improvement in Small Companies.” In *Proceedings of the 2009 ACM Symposium on Applied Computing*, 509–516. SAC ’09. New York, NY, USA: ACM, 2009. Doi:10.1145/1529282.1529389.

37 Espinosa-Curiel, Ismael Edrein, Josefina Rodríguez-Jacobo,

| | |
|---|---|
| | and José Alberto Fernández-Zepeda. "A Competency Framework for the Stakeholders of a Software Process Improvement Initiative." In *Proceedings of the 2011 International Conference on Software and Systems Process*, 139–148. ICSSP '11. New York, NY, USA: ACM, 2011. Doi:10.1145/1987875.1987898. |
| 38 | Staples, Mark, and Mahmood Niazi. "Two Case Studies on Small Enterprise Motivation and Readiness for CMMI." In *Proceedings of the 11th International Conference on Product Focused Software*, 63–66. PROFES '10. New York, NY, USA: ACM, 2010. Doi:10.1145/1961258.1961274. |
| 39 | Dyb\aa, Tore. "Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context." In *Proceedings of the 9th European Software Engineering Conference Held Jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 148–157. ESEC/FSE-11. New York, NY, USA: ACM, 2003. Doi:10.1145/940071.940092. |
| 40 | Mesquida, Antoni Lluís, Antonia Mas, Esperança Amengual, and Jose A. Calvo-Manzano. "IT Service Management Process Improvement Based on ISO/IEC 15504: A Systematic Review." *Information and Software Technology* 54, no. 3 (March 2012): 239–247. Doi:10.1016/j.infsof.2011.11.002. |
| 41 | Staples, Mark, and Mahmood Niazi. "Systematic Review of Organizational Motivations for Adopting CMM-based SPI." *Information and Software Technology* 50, no. 7–8 (June 2008): 605–620. Doi:10.1016/j.infsof.2007.07.003. |
| 42 | Pettersson, F., M. Ivarsson, T. Gorschek, and P. Öhman. "A Practitioner's Guide to Light Weight Software Process Assessment and Improvement Planning." *Journal of Systems and Software* 81, no. 6 (June 2008): 972–995. Doi:10.1016/j.jss.2007.08.032. |
| 43 | Habra, Naji, Simon Alexandre, Jean-Marc Desharnais, Claude Y. Laporte, and Alain Renault. "Initiating Software Process Improvement in Very Small Enterprises: Experience with a Light Assessment Tool." *Information and Software Technology* 50, no. 7–8 (June 2008): 763–771. Doi:10.1016/j.infsof.2007.08.004. |
| 44 | Pino, Francisco J., César Pardo, Félix García, and Mario Piattini. "Assessment Methodology for Software Process Improvement in Small Organizations." *Information and Software Technology* 52, no. 10 (October 2010): 1044–1061. Doi:10.1016/j.infsof.2010.04.004. |
| 45 | Lepmets, Marion, Tom McBride, and Eric Ras. "Goal Alignment in Process Improvement." *Journal of Systems and Software* 85, no. 6 (June 2012): 1440–1452. Doi:10.1016/j.jss.2012.01.038. |
| 46 | Villalón, Jose A. Calvo-Manzano, Gonzalo Cuevas Agustín, Tomás San Feliu Gilabert, Antonio De Amescua Seco, Luis García Sánchez, and Manuel Pérez Cota. "Experiences in the Application of Software Process Improvement in SMES." *Software Quality Journal* 10, no. 3 (November 1, 2002): 261–273. Doi:10.1023/A:1021638523413. |
| 47 | Monteiro, P., R.J. Machado, R. Kazman, and C. Henriques. "Dependency Analysis Between CMMI Process Areas." In *Product-Focused Software Process Improvement. 11th International Conference, PROFES, 21-23 June 2010*, 263–75. Product-Focused Software Process Improvement. Proceedings 11th International Conference, PROFES. Springer Verlag, 2010. Doi:10.1007/978-3-642-13792-1_21. |
| 48 | Mishra, Deepti, and Alok Mishra. "Simplified Software Inspection Process in Compliance with International Standards." *Computer Standards and Interfaces* 31, no. 4 (2009): 763–771. Doi:10.1016/j.csi.2008.09.018. |
| 49 | Homchuenchom, Disorn, Chayakorn Piyabunditkul, Horst Lichter, and Toni Anwar. "SPIALS: A Light-weight Software Process Improvement Self-assessment Tool." In *2011 5th Malaysian Conference in Software Engineering, MySEC 2011, December 13, 2011 – December 14, 2011*, 195–199. 2011 5th Malaysian Conference in Software Engineering, MySEC 2011. IEEE Computer Society, 2011. Doi:10.1109/MySEC.2011.6140668. |
| 50 | Mishra, D., and A. Mishra. "Software Process Improvement in SMEs: A Comparative View." *Computer Science and Information Systems* 6, no. 1 (June 2009): 111–40. Doi:10.2298/CSIS0901111M. |
| 51 | McCaffery, F., and G. Coleman. "Lightweight SPI Assessments: What Is the Real Cost?" *Software Process: Improvement and Practice* 14, no. 5 (September 2009): 271–8. Doi:10.1002/spip.430. |
| 52 | Sivashankar, M., A.M. Kalpana, and A.E. Jeyakumar. "A Framework Approach Using CMMI for SPI to Indian SME's." In *2010 International Conference on Innovative Computing Technologies (ICICT), 12-13 Feb. 2010*, 5 pp. 2010 International Conference on Innovative Computing Technologies (ICICT). IEEE, 2010. Doi:10.1109/ICINNOVCT.2010.5440086. |
| 53 | Mishra, Deepti, and Alok Mishra. "Software Process Improvement Methodologies for Small and Medium Enterprises." In *9th International Conference on Product-Focused Software Process Improvement, PROFES 2008, June 23, 2008 – June 25, 2008*, 5089 LNCS:273–288. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, 2008. Doi:10.1007/978-3-540-69566-0_23. |

| | |
|---|---|
| 54 | Omran, A. “AGILE CMMI from SMEs Perspective.” In *3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008*, 1 –8, 2008. Doi:10.1109/ICTTA.2008.4530352. |
| 55 | Al-Zoabi, Z. “Introducing Discipline to XP: Applying PRINCE2 on XP Projects.” In *3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008*, 1–7, 2008. Doi:10.1109/ICTTA.2008.4530347. |
| 56 | Sutherland, J., C.R. Jakobsen, and K. Johnson. “Scrum and CMMI Level 5: The Magic Potion for Code Warriors.” In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 466–466, 2008. Doi:10.1109/HICSS.2008.384. |
| 57 | Vriens, C. “Certifying for CMM Level 2 and IS09001 with XP@Scrum.” In *Agile Development Conference, 2003. ADC 2003. Proceedings of The*, 120–124, 2003. Doi:10.1109/ADC.2003.1231461. |
| 58 | Jakobsen, C.R., and T. Poppendieck. “Lean as a Scrum Troubleshooter.” In *Agile Conference (AGILE), 2011*, 168–174, 2011. Doi:10.1109/AGILE.2011.11. |
| 59 | Miller, J.R., and H.M. Haddad. “Challenges Faced While Simultaneously Implementing CMMI and Scrum: A Case Study in the Tax Preparation Software Industry.” In *2012 Ninth International Conference on Information Technology: New Generations (ITNG)*, 314–318, 2012. Doi:10.1109/ITNG.2012.53. |
| 60 | Lina, Zhang, and Shao Dan. “Research on Combining Scrum with CMMI in Small and Medium Organizations.” In *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 1:554–557, 2012. Doi:10.1109/ICCSEE.2012.477. |
| 61 | Marcal, A.S.C., B.C.C. de Freitas, F.S. Furtado Soares, and A.D. Belchior. “Mapping CMMI Project Management Process Areas to SCRUM Practices.” In *31st IEEE Software Engineering Workshop, 2007. SEW 2007*, 13–22, 2007. Doi:10.1109/SEW.2007.102. |
| 62 | Khan, M.I., M.A. Qureshi, and Q. Abbas. “Agile Methodology in Software Development (SMEs) of Pakistan Software Industry for Successful Software Projects (CMM Framework).” In *2010 International Conference on Educational and Network Technology (ICENT)*, 576–580, 2010. Doi:10.1109/ICENT.2010.5532104. |
| 63 | Osorio Martinez, Z.B., E. Irrazabal, and J. Garzas. “Toward Improving Agile Mantema: Measurement, Control and Evaluation of Maintenance Projects in SMEs.” In *2011 6th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6, 2011. |
| 64 | McCaffery, F., M. Pikkarainen, and I. Richardson. “Ahaa – agile, Hybrid Assessment Method for Automotive, Safety Critical Smes.” In *ACM/IEEE 30th International Conference on Software Engineering, 2008. ICSE ’08*, 551–560, 2008. Doi:10.1145/1368088.1368164. |
| 65 | Ruiz, J.C., Z.B. Osorio, J. Mejia, M. Munoz, A.M. Chavez, and 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 *2011 IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, 421–426, 2011. Doi:10.1109/CERMA.2011.74. |
| 66 | Al-Tarawneh, Mejhem Yousef, Mohd Syazwan Abdullah, and Abdul Bashah Mat Ali. “A Proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms.” *Procedia Computer Science* 3 (2011): 893–897. Doi:10.1016/j.procs.2010.12.146. |
| 67 | Pino, Francisco J., Oscar Pedreira, Félix García, Miguel Rodríguez Luaces, and Mario Piattini. “Using Scrum to Guide the Execution of Software Process Improvement in Small Organizations.” *Journal of Systems and Software* 83, no. 10 (October 2010): 1662–1677. Doi:10.1016/j.jss.2010.03.077. |
| 68 | Kirk, Diana, and Ewan Tempero. “A Lightweight Framework for Describing Software Practices.” *Journal of Systems and Software* 85, no. 3 (March 2012): 582–595. Doi:10.1016/j.jss.2011.09.024. |
| 69 | Coleman, Gerry, and Rory O’Connor. “Investigating Software Process in Practice: A Grounded Theory Perspective.” *Journal of Systems and Software* 81, no. 5 (May 2008): 772–784. Doi:10.1016/j.jss.2007.07.027. |
| 70 | Salo, O., and P. Abrahamsson. “Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study.” In *2005 International Symposium on Empirical Software Engineering, 2005*, 10 pp.–, 2005. Doi:10.1109/ISESE.2005.1541828. |
| 71 | Lin, Lie-Chien, Tzu-Su Li, and Judy P. Kiang. “A Continual Improvement Framework with Integration of CMMI and Six-sigma Model for Auto Industry.” *Quality and Reliability Engineering International* 25, no. 5 (2009): 551–569. Doi:10.1002/qre.988. |

## APPENDIX C – QUESTIONNAIRE QUESTIONS

### 1. Enterprise

1. What is the name of the enterprise? *

…………

2. What is the number of employees in your enterprise?
   a. 0-20
   b. 20-50
   c. 50-100
   d. 100-200
   e. Other: …..

3. How old is the company?
   a. 0-2 years
   b. 3-5 years
   c. 5-10 years
   d. 10-20 years
   e. 20+ years

4. What is your position on the enterprise?
   ………..

## 2. Agile Methodologies

1. Do you know any of the following methodologies?

| | |
|---|---|
| | Scrum |
| | XP |
| | Lean |
| | Feature Driven Development |
| | Crystal Methodologies |
| | Adaptive Software Development |
| | Dynamic System Development Method |
| | Other (please specify)<br>……… |

2. Have you ever applied any of the following methodologies? For how long?

| | Never | 0-3 years | 3-5 years | 5-10 years | 10 years or more |
|---|---|---|---|---|---|
| Scrum | | | | | |
| XP | | | | | |
| Lean | | | | | |
| Feature Driven Development | | | | | |
| Crystal Methodologies | | | | | |
| Adaptive Software Development | | | | | |
| Dynamic System Development Method | | | | | |
| Other | | | | | |

3. What were the results of applying these methodologies?

| | |
|---|---|
| | Easy adapt to changes |
| | Flexibility |
| | Rapid development |
| | Better visibility in the development process |
| | Better communication between the customers and the enterprise |
| | Continuous planning |
| | Reduce the risk in the development process |
| | Cost reduction |
| | Other (Please specify)<br>…… |

## 3. SPI Maturity Models

1. Do you know any of the following SPI Maturity Models?

| | |
|---|---|
| | CMMI + XP |
| | CMM + XP |
| | PRINCE2 + XP |
| | CMMI + Scrum |
| | CMM + Scrum |
| | CMMI + Lean |
| | CMMI + Six Sigma |
| | PRINCE2 + DSDM |
| | Other (Please Specify)<br>…….. |

2. Have you ever applied any of the following SPI maturity models? For how long?

| | Never | 0-3 years | 3-5 years | 5-10 years | More than 10 years |
|---|---|---|---|---|---|
| CMM | | | | | |
| CMMI | | | | | |
| ISO/IECE 15504 – SPICE | | | | | |
| P3M3 | | | | | |
| OPM3 | | | | | |
| PRINCE2 | | | | | |
| Other | | | | | |

3. What were the results of applying these maturity models?

| | |
|---|---|
| | Better control |
| | Better management |
| | Cost reduction |
| | On – cost and on – time development |
| | Quality increase |
| | Stick on the plans |
| | Risk reduction |

| | |
|---|---|
| | Other (please specify)<br>…….. |

## 4. Agile and SPI combination

1. Do you know any of the following combinations?

| | |
|---|---|
| | CMMI + XP |
| | CMM + XP |
| | PRINCE2 + XP |
| | CMMI + Scrum |
| | CMM + Scrum |
| | CMMI + Lean |
| | CMMI + Six Sigma |
| | PRINCE2 + DSDM |
| | Other (Please Specify)<br>…….. |

2. Have you ever applied any of the combinations? For how long?

| | Never | 0-3 years | 3-5 years | 5-10 years | More than 10 years |
|---|---|---|---|---|---|
| CMMI + XP | | | | | |
| CMM + XP | | | | | |
| PRINCE2 + XP | | | | | |
| CMMI + Scrum | | | | | |
| CMM + Scrum | | | | | |
| CMMI + Lean | | | | | |
| CMMI + Six Sigma | | | | | |
| PRINCE2 DSDM | | | | | |
| Other | | | | | |

3. What are the results of these methodologies?

| | |
|---|---|
| | Quality increase |
| | Cost reduction |
| | Better management |
| | Clear visibility in the development process |
| | Risk reduction |
| | Stick on the plans |
| | On – time development |
| | On – cost development |
| | Flexibility in adopting the changes |
| | Better relationship between the customers and the enterprise |
| | Other (please specify)<br>…….. |

## 5. Improvements

1. Do you have any suggestions – proposals of improving the software processes?
………

2. Do you know or have you ever applied or developed any other method(s) for improvement?
………

3. Would you agree if we ask for an interview, in order to receive more detailed information?
   a. Yes
   b. No

## REFERENCES

[1] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” *Guid. Perform. Syst. Lit. Rev. Softw. Eng.*, 2007.

[2] F. L. Ribeiro and M. T. Fernandes, “Exploring agile methods in construction small and medium enterprises: a case study,” *Journal of Enterprise Information Management*, vol. 23, no. 2, pp. 161–80, 2010.

[3] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.

[4]“Zotero | Home.” [Online]. Available: http://www.zotero.org/. [Accessed: 01-Dec-2013].

[5] B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, and S. Linkman, “Systematic literature reviews in software engineering – A tertiary study,” *Information and Software Technology*, vol. 52, no. 8, pp. 792–805, Aug. 2010.

[6] “Online survey software. Free web survey tool. eSurveysPro.com.” [Online]. Available: http://www.esurveyspro.com/. [Accessed: 01-Dec-2013].

[7] E. Kabaale and J. Nabukenya, “A Systematic Approach to Requirements Engineering Process Improvement in Small and Medium Enterprises: An Exploratory Study,” in *Product-Focused Software Process Improvement. 12th International Conference, PROFES 2011, 20-22 June 2011*, 2011, pp. 262–75.

[8] D. Mishra and A. Mishra, “Software process improvement in SMEs: A comparative view,” *Computer Science and Information Systems*, vol. 6, no. 1, pp. 111–140, 2009.

[9] J. A. Highsmith, *Agile Software Development Ecosystems*. Addison-Wesley Professional, 2002.

[10] E. Kamsties, K. Hörmann, and M. Schlich, “Requirements Engineering in Small and Medium Enterprises: State-of-the-practice, problems, solutions and technology transfer,” in *Proc. Conference on European Industrial Requirements Engineering (CEIRE’98), Hammersmith, UK*, 1998.

[11] A. Cockburn, *"Agile Software Development Joins the "Would-Be" Crowd,"* Cutter IT Journal, pp. 6-12, Jan. 2002.

[12] T. Merten, K. Lauenroth, and S. Bursner, “Towards a New Understanding of Small and Medium Sized Enterprises in Requirements Engineering Research,” in *Requirements Engineering: Foundation for Software Quality. 17th International Working Conference, REFSQ 2011, 28-30 March 2011*, 2011, pp. 60–5

[13] C. R. Klaus Pohl, *Requirements Engineering Fundamentals*. O’Reilly.

[14] A. Talbot and A. Connor, “Requirements Engineering Current Practice and Capability in Small and Medium Software Development Enterprises in New Zealand,” in *2011 9th International Conference*

*on Software Engineering Research, Management and Applications (SERA)*, 2011, pp. 17 –25.

[15] M. Kauppinen, M. Vartiainen, J. Kontio, S. Kujala, and R. Sulonen, “Implementing requirements engineering processes throughout organizations: success factors and challenges,” *Information and Software Technology*, vol. 46, no. 14, pp. 937–953, Nov. 2004.

[16] F. J. Pino, F. García, and M. Piattini, “Software process improvement in small and medium software enterprises: a systematic review,” *Software Qual J*, vol. 16, no. 2, pp. 237–261, Jun. 2008.

[17] M. Y. al-Tarawneh, M. S. Abdullah, and A. B. M. Ali, “A proposed methodology for establishing software process development improvement for small software development firms,” *Procedia Computer Science*, vol. 3, pp. 893–897, 2011.

[18] B. Sharma, N. Sharma, and N. Sharma, “Software Process Improvement: A Comparative Analysis of SPI models,” in *2009 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET)*, 2009, pp. 1019–1024.

[19] “Software process improvement in small and medium software enterprises: a systematic review - Springer.”

[20] H. Kaindl, S. Brinkkemper, J. A. Bubenko Jr, B. Farbey, S. J. Greenspan, C. L. Heitmeyer&#x0002A;, J. C. S. do P. Leite&#x02020;, N. R. Mead, J. Mylopoulos, and J. Siddiqi, “Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda,” *Requirements Engineering*, vol. 7, no. 3, pp. 113–123, Sep. 2002.

[21] M. Unterkalmsteiner, T. Gorschek, A. K. M. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, “Evaluation and Measurement of Software Process Improvement #x02014;A Systematic Literature Review,” *IEEE Transactions on Software Engineering*, vol. 38, no. 2, pp. 398–424, 2012.

[22] F. J. Pino, F. Garcia, and M. Piattini, “Key processes to start software process improvement in small companies,” in *Proceedings of the 2009 ACM symposium on Applied Computing*, New York, NY, USA, 2009, pp. 509–516.

[23] J. A. C.-M. Villalón, G. C. Agustín, T. S. F. Gilabert, A. D. A. Seco, L. G. Sánchez, and M. P. Cota, “Experiences in the Application of Software Process Improvement in SMES,” *Software Quality Journal*, vol. 10, no. 3, pp. 261–273, Nov. 2002.

[24] L. Scott, R. Jeffery, L. Carvalho, J. D’Ambra, and P. Rutherford, “Practical software process improvement - the IMPACT project,” in *Software Engineering Conference, 2001. Proceedings. 2001 Australian*, 2001, pp. 182–189.

[25] S. Jantunen, “Exploring software engineering practices in small and medium-sized organizations,” in *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, New York, NY, USA, 2010, pp. 96–101.

[26] A. Simon, R. Alain, H. Naji: *“OWPL: A Gradual Approach for Software Process Im- provement In SMEs”*. In: Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA 2006) (2006)

[27] D. Mishra and A. Mishra, “Software process improvement in SMEs: A comparative view,” *Computer Science and Information Systems*, vol. 6, no. 1, pp. 111–140, 2009.

[28] G. Cugola and C. Ghezzi, “Software Processes: A Retrospective and a Path to the Future,” Software Process: Improvement and Practice, vol. 4, no. 3, pp. 101-123, Sept. 1998.

[29] A. Fuggetta, “Software Process: A Roadmap,” Proc. Conf. Future of Software Eng., pp. 25-34, 2000.

[30] N. Ehsan, A. Perwaiz, J. Arif, E. Mirza, and A. Ishaque, “CMMI / SPICE based process improvement,” in *2010 IEEE International Conference on Management of Innovation and Technology (ICMIT)*, 2010, pp. 859–862.

[31] SEI, CMMI for I, Version 1.2. Technical Report CMU/SEI-2006-TR-008., 2006.http://www.sei.cmu.edu/cmmi/

[32] S. Garcia, S. Cepeda, J.M. Staley, G. Miluk, “Summary of CMMI For Small Business Pilot Study in Huntsville. Alabama, USA”, Presented at Info Seminar, June 2004, Carnegie Mellon Software Engineering Institute, Pittsburgh, USA

[33] PRINCE2_Maturity_Model_P2MM.pdf http://www.p3m3-officialsite.com/P3M3Model/Model_mhtry.aspx

[34] J. Dorr, S. Adam, M. Eisenbarth, and M. Ehresmann, “Implementing Requirements Engineering Processes: Using Cooperative Self-Assessment and Improvement,” *IEEE Software*, vol. 25, no. 3, pp. 71 –77, Jun. 2008.

[35] B. Kitchenham, S. Linkman, and D. Law, “DESMET: a methodology for evaluating software engineering methods and tools,” Computing Control Engineering Journal, vol. 8, no. 3, pp. 120–126, June.

[36] T. Dybå and T. Dingsøyr, “Empirical studies of agile software development: A systematic review,” *Information and Software Technology*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008.

[37] W. C. de Souza Carvalho, P. F. Rosa, M. dos Santos Soares, M. A. Teixeira da Cunha Junior, and L. C. Buiatte, “A Comparative Analysis of the Agile and Traditional Software Development Processes Productivity,” in *Computer Science Society (SCCC), 2011 30th International Conference of the Chilean*, 2011, pp. 74–82.

[38] Z. Al-Zoabi, “Introducing Discipline to XP: Applying PRINCE2 on XP Projects,” in *3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008*, 2008, pp. 1–7.

[39] A. Omran, “AGILE CMMI from SMEs perspective,” in *3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008*, 2008, pp. 1 –8.

[40] D. Cohen, M. Lindvall, and P. Costa, “Agile software development,” *Data & Analysis Center for Software (DACS), New York*, 2003.

[41] K. Beck, A. Cockburn, R. Jeffries, and J. Highsmith. Agile Manifesto. http://www.agilemanifesto.org . 2001. 12-4-2002.

[42] J. Highsmith, K. Orr, and A. Cockburn, "Extreme Programming," E-Business Application Delivery, pp. 4-17, Feb. 2000.

[43] S. Nerur and V. Balijepally, “Theoretical Reflections on Agile Development Methodologies,”Comm. ACM, vol. 50, no. 3, 2007, pp.79–83.

[44] S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, Communications of the ACM (May) (2005) 72–78.

[45] A. Qumer, and B. Henderson-Sellers (2006b), *“Measuring agility and adoptability of agile methods: a 4- dimensional analytical tool”*, in N. es Guimara˜, P. Isaias and A. Goikoetxea (Eds), Proceedings. IADIS International Conference Applied Computing, IADIS Press, pp. 503-7.

[46] J. Erickson, K. Lyytinen, and K. Siau, “Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research,” *Journal of Database Management*, vol. 16, no. 4, pp. 88–100, 34 2005.

[47] A. Thomas, R. Barton, and C. C. Okafor, “Applying lean six sigma in a small engineering company : a model for change,” *Journal of Manufacturing Technology Management*, vol. 20, no. 1, pp. 113–29, 2009.

[48] J. P. Womack, D. T. Jones (2003) *“Lean thinking: banish waste and create wealth in your corporation”*. Simon & Schuster, New York.

[49] U. Dombrowski, I. Crespo, and T. Zahn, “Adaptive Configuration of a Lean Production System in Small and Medium-sized Enterprises,” *Production Engineering*, vol. 4, no. 4, pp. 341–8, 2010.

[50] H. Takeuchi and I. Nonaka, *“The new product development game,”* Harvard Business Review (January- February), 1986.

[51] K. Schwaber and M. Beedle, *“Agile Software Development with*

*SCRUM"*, Prentice-Hall, 2002.

[52] B. Fitzgerald, K.-J. Stol, R. O&#039;Sullivan, and D. O&#039;Brien, "Scaling agile methods to regulated environments: an industry case study," in *Proceedings of the 2013 International Conference on Software Engineering*, Piscataway, NJ, USA, 2013, pp. 863–872.

[53] D. Cohen, M. Lindvall, and P. Costa, *"An introduction to agile methods"*, Advances in Computers, vol. 62, pp. 2–67, 2004.

[54] K. Beck, *"Extreme Programming Explained: Embrace Chang"*, Addison-Wesley, 2004, ISBN 978-0321278654

[55] A. Cockburn, *"Crystal Clear: A Human-Powered Methodology for Small Teams"*, Addison-Wesley, 2004, ISBN 0-201-69947-8.

[56] J. Stapleton, "DSDM: Dynamic Systems Development Method," in *Proceedings of Technology of Object-Oriented Languages and Systems, 1999*, 1999, pp. 406–406.

[57] S. R. Palmer and M. Felsing, *A Practical Guide to Feature-Driven Development*, 1st ed. Pearson Education, 2001.

[58] P. Coad et al. *"Java modeling in Color with UML"*, Upper Saddle River, NJ:Prentice Hall PTR, 1999.

[59] J. A. Highsmith, *"Adaptive Software Development: A Collaborative Approach to Managing Complex Systems"*, 2000 New York: Dorset House, 392pp, ISBN 0-932633-40-4.

[60] North American and European Enterprise Software and Services Survey, Business Technographics Ed., 2005.

[61] A. Begel and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study," in *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*, 2007, pp. 255–264.

[62] R. Sison and T. Yang, "Use of Agile Methods and Practices in the Philippines," in *Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific*, 2007, pp. 462–469.

[63] D. Turk, R. France, and B. Rumpe, *"Assumptions underlying agile software-development processes,"* J. Datab. Manage., vol. 16, 2005.

[64] P. Mcbreen, Questioning Extreme Programming, Pearson Education, Boston, MA, USA, 2003, ISBN 0-201- 84457-5.

[65] M. Cohn, and D. Ford, "Introducing an Agile Process to an Organization", http://www.mountaingoatsoftware.com/articles/IntroducingAnAgileProcess.pdf .2002. 8-2-2002.

[66] J. Stanworth and D. Purdy, 2003. *"SME fact and issues, a compilation of current data and issues on UK small and medium sized firms"*. London: University of Westminster.

[67] D. Leffingwell, *"Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise"*, 1st ed., ser. Agile Software Development Series. Addison-Wesley Professional, January 2011, no. ISBN- 13: 978-0321635846.

[68] T. Dyba and T. Dingsoyr, "What Do We Know about Agile Software Development?," *IEEE Software*, vol. 26, no. 5, pp. 6–9, 2009.

[69] B. J. Hicks, and J. Matthews, 2010. *"The barriers to realising sustainable process improvement: a root cause analysis of paradigms for manufacturing systems improvement"*, International Journal of Computer Integrated Manufacturing, 23 (7), 585–602.

[70] Y. Wang & G.King, *"Software Engineering Processes: Principles and Applications",* CRC Press LLC, Boca Raton, FL, USA,(2000).

[71] A. Cater-Steel, "An Evaluation Of Software Development Practice And Assessment-Based Process Improvement In Small Software Development Firms," Ph.D. Dissertation, Sch..Com. Info. Tech, Griffith Uni, Australia, (2004).

[72] K. Kautz, *"Software Process Improvement in Very Small Enterprises: Does it Pay Off ?"*, Software Process – Improvement and Practice, 4(4), 209-226. (1998).

[73] C. G. V. Wangenheim, S. Weber, JCR. Hauck, G. Trentin, *"Experiences on establishing software processes in small companies",* Information and Software Technology, 48(2006), 890-900. (2006).

[74] H. Oktaba, F. Garcia, M. Piattini, F. Ruiz, C. Alquicira, *"Software Process Improvement: The Competisoft Project"*, IEEE Computer, 40(10), 21-28. (2007).

[75] S. Trudel, J. M. Lavoie, M. C. Pare, W. Suryn, *"PEM: The small company- dedicated software process quality evaluation method combining CMMI and ISO/IEC 14598"*. Software Quality Journal, 14(1), 7-23. (2006).

[76] D. H. Bae, *"Panel: Software Process Improvement for Small organizations"*, 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), IEEE Computer Society. (2007).

[77] T. Dyba, "Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context," in *Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, New York, NY, USA, 2003, pp. 148–157.

[78] A. P. Cater-Steel, *"Low-rigour, rapid software assessments for small software development firms"* Proceedings of Australian Software Engineering Conference (ASWEC), Melbourne, 368-377. (2004).

[79] J. A. Calvo-Manzano, G. C. Agustin, T. S. F. Gilabert, A. D. A. Seco, L. Z. Sanchez, M. P. Cota, *"Experiences in the Application of Software Process Improvement in SMES",* Software Quality Journal, 10, 261-273. (2002).

[80] M. García, J. J. D. Cosín, *"Medición práctica de la coordinación utilizando GQ (I) M y CMMI"*, taller Apoyo a la decisión en ingeniería del software, septiembre 2007.

[81] M. García, E. Irrazábal, J. Garzás, *"Implantación de las normas ISO/IEC 15504 e ISO/IEC 12207:2008 con métodos ágiles y SCRUM,"* 2010.

[82] P. Team *"CMMI for development, version 1.2, CMMI-DEV v1. 2. Continuous Representation, CMU/SEI- 2006-TR-008"*, Technical Report. Software Engineering Institute, August, 2008.

[83] ISO. ISO/IEC 15504:2003, Information Technology Process Assessment - Part 2: Performing an assessment. Geneva, International Organization for Standardization. www.iso.org, 2004.

[84] J. C. Ruiz, Z. B. Osorio, J. Mejia, M. Munoz, A. M. Chavez, and 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 *2011 IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, 2011, pp. 421–426.

[85] M. Flores, A. Cabello, L. Torredemer, M. Agrawal, J. Keast, S. Terzi, and A. Sopelana, *"Do enterprises implement a process architecture towards Lean in product development? A comparative study among large and small firms,"* in *2011 17th International Conference on Concurrent Enterprising (ICE)*, 2011, pp. 1–9.

[86] O. Salo and P. Abrahamsson, "*Integrating agile software development and software process improvement: a longitudinal case study,"* in *2005 International Symposium on Empirical Software Engineering, 2005*, 2005, p. 10 pp.–.

[87] M. Pikkarainen and A. Mäntyniemi, *"An approach for using CMMI in agile software development assessments: experiences from three case studies,"* in SPICE 2006 conference, Luxemburg, 2006, pp. 4–5.

[88] K. C. Dangle, P. Larsen, M. Shaw, and M. V. Zelkowitz, *"Software process improvement in small organizations: a case study," IEEE Software*, vol. 22, no. 6, pp. 68 – 75, Dec. 2005.

[89] D. P. Kelly, and B. Culleton, *"Process improvement for small organizations"*, IEEE Computer, 32, 10(1999), 41- 47.

[90] M. I. Khan, M. A. Qureshi, and Q. Abbas, "Agile methodology in software development (SMEs) of Pakistan software industry for successful software projects (CMM framework)," in *2010*

*International Conference on Educational and Network Technology (ICENT)*, 2010, pp. 576–580.

[91] H. Glazer, J. Datton, D. Anderson, M. Kenradl, and S. Shrum, *"CMMI or Agile: Why Not Embrace Both!"*.

[92] D. J. Anderson, *"Stretching Agile to fit CMMI level 3"*, Agile conference, Denver July 2005.

[93] J. Newkirk, *"Introduction to Agile Processes and Extreme Programming"*, Addison-Wesley, Reading MA, 2002.

[94] Z. Lina and S. Dan, *"Research on Combining Scrum with CMMI in Small and Medium Organizations,"* in *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012, vol. 1, pp. 554–557.

[95] J. Sutherland, C. R. Jakobsen, and K. Johnson, *"Scrum and CMMI Level 5: The Magic Potion for Code Warriors,*" in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 466–466.

[96] C. R. Jakobsen and K. A. Johnson, *"Mature Agile with a Twist of CMMI,"* in Agile, 2008. AGILE '08. Conference, 2008, pp. 212–217.

[97] C. R. Jakobsen and J. Sutherland, *"Scrum and CMMI Going from Good to Great,"* in Agile Conference, 2009. AGILE '09., 2009, pp. 333–337.

[98] A. S. C. Marcal, B. C. C. de Freitas, F. S. Furtado Soares, and A. D. Belchior, *"Mapping CMMI Project Management Process Areas to SCRUM Practices,"* in *31st IEEE Software Engineering Workshop, 2007. SEW 2007*, 2007, pp. 13–22.

[99] C. Vriens, *"Certifying for CMM Level 2 and IS09001 with XP@Scrum,"* in *Agile Development Conference, 2003. ADC 2003. Proceedings of the*, 2003, pp. 120–124.

[100] C. R. Jakobsen and T. Poppendieck, *"Lean as a Scrum Troubleshooter,"* in *Agile Conference (AGILE), 2011*, 2011, pp. 168–174.

[101] R. Jha and A. K. Saini, *"ERP Redefined: Optimizing Parameters with Lean Six Sigma for Small #x0026; Medium Enterprises,"* in *2011 International Conference on Communication Systems and Network Technologies (CSNT)*, 2011, pp. 683–687.

[102] J. M. Edward D. Arnheiter, *"The integration of lean management and Six Sigma," The TQM Magazine*, vol. 17, no. 1, pp. 5–18, 2005.

[103] R. Shah, A. Chandrasekaran, and K. Linderman, *"In pursuit of implementation patterns: the context of Lean and Six Sigma,", International Journal of Production Research*, vol. 46, no. 23, pp. 6679–6699, 2008.

[104] N. Kamalakshi and H. Naganna, *"CMMI and Six Sigma-Relationship #x00026; Integration,"* in *International Conference on Signal Acquisition and Processing, 2009. ICSAP 2009*, 2009, pp. 174–176.

[105] J. Siviy, M. Penn, and E. Harper, *"Relationships Between CMMI and Six Sigma," Software Engineering Institute*, Dec. 2005.

[106] J. M. Siviy, M. L. Penn, and R. W. Stoddard, "*CMMI and Six Sigma: Partners in Process Improvement"*, Pearson Education, 2007.

[107] R. Hefner, *"Using Six Sigma to Accelerate CMMI Adoption (and Vice Versa)."* Presented at the Software Engineering Process Group Conference, Seattle, WA, March 7-10, 2005.

[108] J. Siviy, Jeannine, M. Penn, M. Lynn & R. W. Stoddard, (February 2008). *"Integrating the CMMI and Six Sigma: Strategies"*. Retrieved from http://www.informit.com/articles/article.aspx?p=1161616.

[109] Z. Xiaosong, H. Zhen, Z. Min, W. Jing, and Y. Dainuan, *"Process integration of Six Sigma and CMMI,"* in *6th IEEE International Conference on Industrial Informatics, 2008. INDIN 2008*, 2008, pp. 1650–1653.

[110] J. Bergey et al. "*Results of SEI Independent Research and Development Projects and Report on Emerging Technologies and Technology Trends,"* Chapter 5 (CMU/SEI-2004-TR-018). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.

[111] DSDM with PRINCE2 Task Group. *"Using DSDM with PRINCE2,"* DSDM consortium 2000.

[112] *"DSDM and PRINCE2 -Perfect Marriage or Strange Bedfellows?,"* D. Tudor – Technical Director TCC Ltd- www.tcc-net.com(Research Paper published on www.dsdm.org).

[113] *"Using DSDM with Prince2 - DSDM_Prince2_WP_10.pdf."* [Online]. Available: http://leadinganswers.typepad.com/leading_answers/files/DSDM_Prince2_WP_10.pdf. [Accessed: 31-Aug-2013].

[114] S. A. Bohner, "Software Change Impact Analysis," 1996.

[115] O. Pedreira, Et Al., "A Systematic Review Of Software Process Tailoring," Acm Sigsoft Software Engineering Notes, Vol. 32, No. 3, (2007) , Pp. 1-6.

[116] M. Habib, S. Ahmed, A. Rehmat, M. J. Khan, and S. Shamail, "Blending Six Sigma and CMMI - an approach to accelerate process improvement in SMEs," in *Multitopic Conference, 2008. INMIC 2008. IEEE International*, 2008, pp. 386–391.

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