Applying Continuous CMMI on Small and Medium Sized Companies with Agile Practices

Master Thesis

Ivica Stanimirovic

Madrid, October 2018
This thesis is submitted to the ETSI Informáticos at Universidad Politécnica de Madrid in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering.

Master Thesis
Master Universitario en Ingeniería del Software – European Master in Software Engineering
Thesis Title: Applying Continuous CMMI on Small and Medium Sized Companies with Agile Practices
Thesis no: EMSE-2018-9
October 2018

Author: Ivica Stanimirovic
Graduated Engineer for Electrical Engineering and Computer Science for Computer Science and Informatics
University of Nis

Supervisor:

Jose A. Calvo-Manzano Villalón
Doctor en Informática
Universidad de Vigo

Lenguajes y Sistemas Informáticos e Ingeniería de Software
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Acknowledgements

I would like to express my utmost gratitude to my supervisor Jose A. Calvo-Manzano, for his understanding, participation and support during this project.

I would like to thank my family and friends; without whom I would not have achieved any of this.
Abstract

The Software industry is one of the fastest growing branches of economy in the EU because of the highly educated workforce. Small and medium-sized enterprises (SMEs) represent 99% of all businesses in the EU, including the software industry.

Most young software development SMEs start their development without a well-defined development approach. As the SMEs mature their need for a well-structured development approach. The implementation of methodologies to structure and control the development activities in start-ups and SMEs is a major challenge for engineers. In general, the management of software development is achieved through the introduction of software processes. In the SME context, software engineering faces complex and multifaceted obstacles in understanding how to manage development processes.

SMEs usually have limited resources and typically wish to use them to support product development instead of establishing processes. This means that they turn to Agile methodology for a product and customer-oriented approach. Many new SMEs focus on team productivity, asserting more control to the employees instead of providing them with rigid guidelines. But that flexibility and freedom can negatively influence the quality of the product or on the vision and future direction of development.

Compared to the Agile methodology, there other approaches for software development with more structure. CMMI-DEV is used in this thesis. Agile and CMMI-DEV practices are often considered as opposites in the software development methodologies spectrum. Agile is seen as fluid and adaptable but lacking solid management and improvement strategies while CMMI is seen as rigid, costly to implement and maintain but with clearly defined goals and methods to reach them.

In this thesis an attempt is made, to create a process improvement approach for Agile environments based on the CMMI-DEV model. To that end a Systematic Literature Review (SLR) was conducted to determine the state of the art in relation with the possible cooperation of CMMI-DEV and Agile. During the SLR, it was decided that the direct combination of CMMI-DEV and Agile is too vague and hard to practically implement, so SCRUM was chosen as a representing Agile practice.

Using SCRUM as the new focus of the CMMI-DEV – Agile combination, a mapping was made, based on the SLR, between process areas of CMMI-DEV for Maturity Level 2 and SCRUM practices. Not all process areas could be covered and for some process areas additional actions had to be defined.

With the mapping of CMMI-DEV process areas to SCRUM defined the next step was creating a way to determine the current process area coverage in the SCRUM environment. This was done by tailoring the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) to the needs of an SME using SCRUM. The tailoring produced questionnaires, based on the previously defined mapping, and was created with having the limited resources available for process improvement in SMEs in mind.

The tailored SCAMPI covered the first two phases of the new process improvement approach. The two remaining phases of the improvement approach were defined by
using a combination of phases from AFIM (Action Focus Improvement Model) and IDEAL (Initiating, Diagnosing, Establishing, Acting, Leveraging) improvement lifecycle models and tailoring the combination to a SME environment.

After the new improvement approach was defined a case study was conducted in a Madrid based SME. All the problems that were encountered as well as the results of the case study can be seen in the case study chapter. The main conclusion of the case study is that the improvement approach was a good start, but it requires further refinement in order to reach its peak performance.

At the end of the thesis in the conclusion chapter an overview of the work done is given, with a focus on the contributions made by the thesis. In the end the positive and the negative sides of the new approach are highlighted and suggestions for future research are given.
Table of Contents

1. INTRODUCTION .................................................................................................................. 1
   1.1 CONTEXT .......................................................................................................................... 1
   1.2 PROBLEM DESCRIPTION ................................................................................................. 2
   1.3 IMPORTANCE OF RESOLUTION ...................................................................................... 3
   1.4 GOALS AND SUB GOALS OF THE THESIS ................................................................. 4
   1.5 APPROACH TO THE SOLUTION ...................................................................................... 4
   1.6 THESIS STRUCTURE ........................................................................................................ 5

2. STATE OF THE ART ................................................................................................................. 5
   2.1 CMMI-DEV ...................................................................................................................... 6
   2.2 AGILE SOFTWARE DEVELOPMENT .............................................................................. 10
   2.3 POSSIBILITY OF COMBINED USE ................................................................................ 11
   2.4 SYSTEMATIC LITERATURE REVIEW ............................................................................ 12
      2.4.1 SLR process ............................................................................................................. 12
      2.4.2 SLR Conclusion ...................................................................................................... 16
   2.5 ANALYSIS AND INTERPRETATION .............................................................................. 17
      2.5.1 Is CMMI applicable successfully on a SME using Agile ........................................ 17
      2.5.2 What practices have been so far on applying the CMMI-DEV based process 
          improvement on Agile oriented SDO? ................................................................. 18
   2.6 FINAL CONSIDERACIONS RELATED TO CMMI-DEV AND AGILE ....................... 19
   2.7 SCAMPI .......................................................................................................................... 19
      2.7.1 What is SCAMPI A? ............................................................................................... 19
      2.7.2 Core Concepts and Approach ............................................................................. 20
      2.7.3 SCAMPI A Methodology ..................................................................................... 20
   2.8 AFIM ............................................................................................................................... 22
   2.9 FINAL CONSIDERATIONS OF SCAMPI ................................................................... 23

3. RESOLUTION ............................................................................................................................ 25
   3.1 TAILORING SCAMPI A ................................................................................................. 25
      3.1.1 Mapping CMMI process areas to SCRUM ............................................................ 26
      3.1.2 Creating the questionnaires ................................................................................... 31
      3.1.3 Commitment to SPI ............................................................................................... 32
      3.1.4 Execution of appraisal ........................................................................................... 35
   3.2 INFRASTRUCTURE AND ACTIONS PLANS ............................................................... 40
      3.1.5 Infrastructure ......................................................................................................... 40
      3.1.6 Process Improvement Plan (PIP) ........................................................................... 42

4. CASE STUDY ........................................................................................................................... 47
   4.1 CASE STUDY METHODOLOGY IN SOFTWARE ENGINEERING ............................... 47
   4.2 CONTEXT ....................................................................................................................... 48
      4.2.1 SCRUM in Glownet ............................................................................................... 49
   4.3 CASE STUDY IMPLEMENTATION STRATEGY ............................................................. 50
   4.4 IMPLEMENTING THE APPROACH ............................................................................... 50
      4.4.1 Obtaining commitment ......................................................................................... 50
      4.4.2 Appraisal plan ....................................................................................................... 51
      4.4.3 Training the appraisal team .................................................................................. 52
      4.4.4 Risk assessment ................................................................................................... 52
      4.4.5 Executing the assessment .................................................................................... 53
      4.4.6 Reporting assessment results ............................................................................. 62
      4.4.7 Process improvement plan and infrastructure ..................................................... 62
      4.4.8 Discussing the new processes .......................................................................... 64
   4.5 CONCLUSIONS ................................................................................................................ 67
5. CONCLUSION......................................................................................................................... 69
  5.1 SUMMARY .......................................................................................................................... 69
  5.2 RESEARCH QUESTION REVISITED .................................................................................. 69
  5.3 CONTRIBUTIONS AND OUTCOMES OF THE RESEARCH ............................................. 70
  5.4 LIMITATIONS .................................................................................................................... 71
  5.5 RECOMMENDATIONS FOR FUTURE WORK ................................................................. 71
6. REFERENCES .......................................................................................................................... 73
  6.1 SLR PAPERS ....................................................................................................................... 74
7. APPENDIX ................................................................................................................................. 77
  7.1 CONFIGURATION MANAGEMENT (CM) ......................................................................... 77
  7.1.1 Questions tailored for SCAMPI CM area ................................................................. 77
  7.1.2 Answers for SCAMPI CM area .................................................................................. 79
  7.2 PROJECT MANAGEMENT AND CONTROL (PMC) ....................................................... 80
  7.2.1 Questions tailored for SCAMPI PMC area ............................................................... 80
  7.2.2 Answers tailored for SCAMPI PMC area ................................................................. 83
  7.3 PROJECT PLANNING (PP) ............................................................................................... 85
  7.3.1 Questions tailored for SCAMPI PP area .................................................................... 85
  7.3.2 Answers tailored for SCAMPI PP area ....................................................................... 88
  7.4 REQUIREMENT MANAGEMENT (REQM) .................................................................... 90
  7.4.1 Questions tailored for SCAMPI REQM area ............................................................ 90
  7.4.2 Answers tailored for SCAMPI REQM area ............................................................... 92
List of tables

Table 2.1 CMMI levels ............................................................................................................. 7
Table 2.2 SLR papers processed ............................................................................................ 16
Table 3.1 Project Monitoring and Control mapping to SCRUM .............................................. 27
Table 3.2 Project Planning mapping to SCRUM ................................................................. 28
Table 3.3 Requirements Management mapping to SCRUM ................................................ 29
Table 3.4 Configuration management mapping to SCRUM ................................................ 30
Table 3.5 Example of questions derived from the mapping .................................................. 32
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Eurostat data on SMEs in Europe</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Template used</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>ISO 15504 standard discreet answers</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Flow diagram of actions to be taken during the assessment</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Improvement infrastructure from AFIM</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>Development teams structure</td>
<td>48</td>
</tr>
<tr>
<td>4.2</td>
<td>Glownet system architecture</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>SCRUM board example</td>
<td>50</td>
</tr>
<tr>
<td>4.4</td>
<td>Example of google calendar scheduling tool</td>
<td>52</td>
</tr>
<tr>
<td>4.5</td>
<td>Project planning specific practice coverage</td>
<td>53</td>
</tr>
<tr>
<td>4.6</td>
<td>Project Monitoring Control specific practice coverage</td>
<td>54</td>
</tr>
<tr>
<td>4.7</td>
<td>Configuration Management specific practice coverage</td>
<td>54</td>
</tr>
<tr>
<td>4.8</td>
<td>Requirements management specific practices coverage</td>
<td>55</td>
</tr>
<tr>
<td>4.9</td>
<td>Overall coverage of the process areas</td>
<td>55</td>
</tr>
<tr>
<td>4.10</td>
<td>Project Planning Answers analysis</td>
<td>57</td>
</tr>
<tr>
<td>4.11</td>
<td>Project Monitoring and Control Answers analysis</td>
<td>58</td>
</tr>
<tr>
<td>4.12</td>
<td>Configuration Management Answers analysis</td>
<td>59</td>
</tr>
<tr>
<td>4.13</td>
<td>Requirements Management Answer analysis</td>
<td>60</td>
</tr>
<tr>
<td>4.14</td>
<td>Project planning specific practice coverage after resolving inconsistencies</td>
<td>61</td>
</tr>
<tr>
<td>4.15</td>
<td>Project Management Control specific practice coverage after resolving inconsistencies</td>
<td>61</td>
</tr>
<tr>
<td>4.16</td>
<td>Requirements management specific practice coverage after resolving inconsistencies</td>
<td>62</td>
</tr>
<tr>
<td>4.17</td>
<td>Project Planning specific sub practice coverage after solving inconsistencies</td>
<td>63</td>
</tr>
<tr>
<td>4.18</td>
<td>Improved SCRUM process with project planning</td>
<td>66</td>
</tr>
</tbody>
</table>
1. Introduction

In this chapter the current context for the paper is described, giving an explanation of the current status of agile development in software development organizations and how CMMI is being used. Then the problem which will be addressed in the thesis is depicted. Followed by a subsection about importance of resolving the given problem. Next a subsection about goals of the whole thesis will be presented. After the goals of the thesis have been explained the approach to solving the depicted problems is described. In the end of this chapter the Structure of the whole thesis is briefly explained.

1.1 Context

The methodologies and process models that have been used on the Software Development Organizations (SDO) have evolved over time. Agile methodologies are being embraced by more and more SDOs in any of its many forms (D. K. Rigby, J. Sutherland, H. Takeuchi, 2016).

Agile approaches utilize technical and managerial processes that continuously adapt and adjust to (1) changes derived from experiences gained during development, (2) changes in software requirements, and (3) changes in the development environment.

Agile processes are intended to support early and quick production of working code. This is accomplished by structuring the development process into iterations, where an iteration focuses on delivering working code and other artefacts that provide value to the customer and, secondarily, to the project. Agile process proponents and critics often emphasize the code focus of these processes. Proponents often argue that code is the only deliverable that matters, and marginalize the role of analysis and design models and documentation in software creation and evolution. Agile process critics point out that the emphasis on code can lead to corporate memory loss because there is little emphasis on producing good documentation and models to support software creation and evolution of large, complex systems.

The claims made by agile process proponents and critics lead to questions about what practices, techniques, and infrastructures are suitable for software development in today’s rapidly changing development environments. In particular, answers to questions related to the suitability of agile processes to particular application domains and development environments are often based on anecdotal accounts of experiences.

Following the agile development methodology and the agile manifesto (agilemanifesto.org) have its benefits and its drawbacks, but empirical data suggest that for most companies the benefits outweigh the drawbacks. Agile development is being used in small and medium enterprises (SMEs) and start-ups because among others setting up well defined software process improvement (SPI) requires a lot of time and effort, most of the SMEs are not ready to invest so much time and effort and risk being overwhelmed by the resources need to adhere to it. In most countries with a developed software industry it is made up mainly of SMEs. This in turn means that SMEs are essential for further advancements in that industry. Improving the existing developments processes in the SMEs would benefit the companies, the industry and also the discipline of Software Engineering.
Chapter 1: Introduction

Opposed to light software development processes which Agile is subset of, CMMI has been adopted advantageously in large companies enabling large scale improvements quality control budget fulfilling, and customer satisfaction.

The core of CMMI is a set of global best practices organized by critical business capabilities which improve business performance. These critical capabilities address the biggest challenges common to any organization, including:

- Engineering and Developing Products.
- Improving Performance.
- Building and Sustaining Capability.
- Managing Business Resilience.
- Planning and Managing Work.
- Selecting and Managing Suppliers.
- Ensuring Quality.
- Managing the Workforce.
- Supporting Implementation.

The aim of this thesis is to apply some of the CMMI improvement practices on software developing SMEs that have already embraced the agile methodology.

1.2 Problem Description

The global software industry keeps growing. This in term means expansion of the existing companies as well as creation of new companies (start-ups). There are many similarities and connections between SMEs and start-ups. Start-ups begin as small companies with few employees and big potential but the challenges faced both financial but also structural.

The implementation of methodologies to structure and control the development activities in start-ups and SMEs is a major challenge for engineers. In general, the management of software development is achieved through the introduction of software processes, which can be defined as “the coherent set of policies, organizational structures, technologies, procedures, and artefacts that are needed to conceive, develop, deploy and maintain a software product” (A. Fuggettab, 2000)

In the SME context, software engineering (SE) faces complex and multifaceted obstacles in understanding how to manage development processes. Start-ups and small companies are creative and flexible in nature and reluctant to introduce process or bureaucratic measures which may hinder their natural attributes. Furthermore, start-ups have very limited resources and typically wish to use them to support product development instead of establishing processes. This means that they turn to Agile methodology as a product and customer oriented approach.

Product-oriented practices help small companies and start-ups in having a flexible team, with workflows that leave them the ability to quickly change the direction according to the targeted market. Therefore, many start-ups focus on team productivity, asserting more control to the employees instead of providing them rigid guidelines. But flexibility and freedom can often negatively influence on the quality of the product or on the vision and future direction of development (T. Dybå, T. Dingsøyr, 2008; C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, & P Abrahamsson, 2016).
1.3 Importance of resolution

Small and medium-sized enterprises (SMEs) represent 99% of all businesses in the EU. They account for around two-thirds of total employment in the EU (Figure 1.1), ranging from 47% in the United Kingdom to 85% in Malta. Enterprises with fewer than 250 persons employed contribute about 56% of the total turnover in the EU (European commission on SME). The year 2016 marked the third consecutive year of steady increases in EU-28 SME employment and EU-28 SME value added (ANNUAL REPORT ON EUROPEAN SMEs 2016/2017).

![Number of enterprises, persons employed and turnover, independent enterprises share of all enterprises with fewer than 250 persons employed](image)

### Countries participating in the 2016 Microdata linking project.

**Figure 1.1 Eurostat data on SMEs in Europe**

Small and medium-sized enterprises are also a significant segment of the Serbian economy:

- they make up 99.8% of total active enterprises,
- employ almost 2/3 employees in the non-financial sector, and
- participate with 32% in the formation of the GDP (Gross Domestic Product) of Serbia

The Software industry is one of the fastest growing branches of economy in the whole world, even more so in the EU because of the highly educated workforce. A global trend of increasing the use of software in every branch of economy and everyday life means that the software industry still has many good years ahead. In 2014, the software industry in the EU directly contributed €249 billion to EU GDP, employed more than 3 million people, and paid over €139.2 billion in wages. When indirect impacts are considered, the software industry supported an additional estimated €228 billion to EU GDP and supported almost 2.5 million additional jobs.

The software industry helped to drive an additional €432 billion to European GDP and over 6 million jobs. Combining all impacts — direct, indirect, and induced — the
software industry added €910 billion to total EU GDP in 2014 (ANNUAL REPORT ON EUROPEAN SMEs 2016/2017).

The software industry like the rest of the economy is also made up of mostly of SMEs. A lot of the SMEs have stated using Agile in some form, tailored to their development process, as a starting point. There are many challenges in reaching stability and maturity from zero, using Agile.

It was shown by multiple studies into the benefits of SPI in software SMEs, that there exists a general relationship between relatively higher levels of SPI activity and business success (Paul Clarke, Rory O’Connor, 2012).

As it is shown in the Figure 1.1 and the numbers presented the SMEs represent a big part of all the businesses and are a “spine of the economy” meaning that an improvement in the software development part of those businesses could have significant positive effect.

1.4 Goals and sub goals of the thesis

As the title of the thesis hints the main goal of the thesis is to

Help improving software development processes in small and medium sized software development enterprises that have already embraced the Agile paradigm using CMMI-DEV as their process model.

The main goal of the thesis can be decomposed into 3 sub goals:

I. Determining the current state of the art for combining CMMI-DEV together with Agile for process improvement.

II. Creating or improving the existing approaches for using CMMI-DEV with Agile for process improvement.

III. Defining an assessment approach of the current state of a small or medium sized enterprise and helping developing a process improvement plan with the corresponding logistic structure.

IV. Definition of the improvements (means of improvement to an enterprise).

1.5 Approach to the Solution

Having the goal and the sub goals as a guiding essence of the thesis and the scope being confined to the small and medium sized enterprises, the particular set of steps and activities to be taken can be defined.

Since the thesis contains a practical part that will be implemented in a real company, one of the practical goals of the thesis is to provide means of improvement to the particular enterprise.

First the context of the software development SMEs in the business world needs to be explored. Next, a systematic literature review needs to be executed in order to determine the state of the art of the CMMI-DEV and Agile fusion for process improvement. Based on the state of the art gained from the systematic literature review, an approach for using CMMI-DEV for improving Agile oriented SMEs is generated.
Chapter 2: State of the art

The development of a new process improvement approach is divided into multiple phases. First phase is creating a new mapping between the CMMI-DEV and a particular Agile based practice. Next step is tailoring the standard CMMI-DEV appraisal method for process improvement using the created mapping and phases from AFIM (Action Focus Improvement Model) and IDEAL (Initiating, Diagnosing, Establishing, Acting, Leveraging) improvement lifecycles. Next phase is creating a new set of actions for the planning and implementation of the processes to be improved.

The approach is further developed and explained and then applied to a real SME in a case study. In the end the results of the case study are gathered and combined with the rest of the thesis products to create a summary of the thesis.

1.6 Thesis Structure

This chapter, titled INTRODUCTION, is the first chapter. It starts with presenting the context of the thesis development. The problems that it strives to solve or remedy is described in order to define the short comings that should be solved within the context. Next the importance of resolution is described to explain the reasons why the particular problem has been chosen as significant for being solved. At the end the goals of the thesis are described, decomposed into sub goals and pragmatically instantiated through objectives.

Chapter 2, titled STATE OF THE ART, is directed at describing the current state of the art related to the goals and sub goals of the thesis. First a systematic literature review is defined and executed with the goal of exploring the existing published approaches to solving the problems defined in the thesis and the shortcoming of the existing approaches as well as generating groundwork for creating a new or improved approach.

Chapter 3, titled SOLUTION, starts from the results of chapter 2 and uses them as a basis for defining an improved process improvement lifecycle. Starting with creating a mapping between CMMI and a specific Agile practice and continuing with all the steps to be taken to determine the current state of the processes and improve them.

Chapter 4, title CASE STUDY, describes the application of the improvement approach defined in Chapter 3 in a Madrid based SME, with all the plans, personnel and infrastructure involved and presentation of the final results of the case study.

The final chapter of the thesis is the CONCLUSION and it summarizes the contributions made by the thesis research. Next the research questions and sub questions are revisited and reflected upon. Consequently, the expected impact of the thesis as well as its limitations are explored. In the end recommendations for future research based on the research done in the thesis are given.

2. State of the art

In this chapter basic attributes of CMMI-DEV and Agile methodology shall be explained and the existing and possible combinations between the two explored. The possibly of cooperation between the two approaches is explored through a systematic literature review and some general conclusions about the combination of Agile and CMMI-DEV as well as some more practical results that are used in the case study are given.
2.1 CMMI-DEV

Capability Maturity Model Integration for Development (CMMI-DEV) (CMMI for Development, Version 1.3) was created by the Software Engineering Institute (SEI) of Carnegie Mellon University and its purpose is the integration of maturity models in software production, and it is often used by organizations and government entities to assess the capability and maturity of a Software Development Organisation (SDO) for the purpose of assigning a project to the SDO.

When this paper was being written, CMMI for Development (CMMI-DEV), the model considered here, was in version 1.3, although version 2.0, was released in April 2018 there is not enough empirical data on the model to be used this early on.

The main goal of CMMI is to be used as a guideline in improving processes by using already established and proven methods of process improvement in order to remove inconsistencies and reduce the overall effort required for improvement.

CMMI for Development contains practices that cover project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes used in development and maintenance.

The process areas in this model represent the best practices for most users. Process areas and practices as per recommendation of the creators should be interpreted using an in-depth knowledge of CMMI-DEV, organizational constraints, and the business environment of the organization.

There are two improvement paths that are offered by the CMMI.

1. One path enables organizations to incrementally improve processes corresponding to an individual process area selected by the organization (capability levels).
2. The other path enables organizations to improve a set of related processes by incrementally addressing successive sets of process areas (maturity levels).

Based on these paths, there are two approaches to improvement which are referred to as “representations”. Those two representations are called continuous and staged respectively. A particular level can be reached only if all of the goals of the process area or set of process areas that are targeted for improvement are satisfied. In both representations means and tools are provided so that the wanted level could be reached.

Capability levels are oriented towards improvement in particular process areas of the organization. The main purpose of these levels is to incrementally improve the processes corresponding to a given process area. There are 4 capability levels numbered 0 through 3 (see Table 2.1).

<table>
<thead>
<tr>
<th>Level</th>
<th>Continuous Representation</th>
<th>Staged Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Incomplete</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Performed</td>
<td>Initial</td>
</tr>
<tr>
<td>2</td>
<td>Managed</td>
<td>Managed</td>
</tr>
<tr>
<td></td>
<td>Defined</td>
<td>Defined</td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>4</td>
<td>Quantitatively managed</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Optimizing</td>
<td></td>
</tr>
</tbody>
</table>

*Table 2.1 CMMI levels*

Maturity levels (see Table 2.1) apply to an organization’s process improvement achievement across multiple process areas. These levels are a means of improving the processes corresponding to a given set of process areas (i.e., maturity level).

**Capability Levels**

CMMI models are used in both continuous and staged representations, so in order to support the continuous representation, all CMMI models reflect capability levels in their design and content.

The four capability levels, each a layer in the foundation for ongoing process improvement, are designated by the numbers 0 through 3:

0. Incomplete. This is the lowest possible state a process can have. Being at this capability level means that the process is either not performed at all or it is performed only partially but in a way that one or more goals of the process are not achieved.

1. Performed. A process of this capability level is being performed and the set goals are being met. Even though reaching this level is a significant improvement compared to level 0, the risk of reverting to the previous level remains because of the lack of explicitly defined guidelines for execution. In order to eliminate the risk of reverting to the incomplete state the process should be improved towards higher levels.

2. Managed. When a process has reached the capability level 2, it means that it is being managed. A managed process can be properly monitored, controlled, and reviewed and it contains detail about required resources (both human and other) for execution according to its description. This level also reinforces the process and enables that the process is performed in accordance with its purpose and definition even at a more intense pace.

3. Defined. This is a highest capability level a process can reach. Being at this level means that the process has been derived from the general guidelines defined by the organisation and tailored for a specific need. It also means that while retaining all the benefits of level 2 it also contributes to the overall process knowledge repository in the organisation and helps to improve other processes.

The main difference between capability levels 2 and 3 is the approach of creating the process. At level 2 the process is defined out of its level 1 predecessor with clearly defined guidelines (bottom-up) while at level 3 it is derived out more general organization process management guidelines (top-down). Level 3 ensures a more consistent set of processes that all originated from the companies process meta-data while also enabling them to contribute to that meta-data.

Another important difference between capability levels 2 and 3 is that level 2 processes are usually more specifically defined since they were derived from previous experience
while at level 3, processes in this area can rely on general metrics which can be applied to any of the processes in the area since they all have the common general structure.

**Advancing Through Capability Levels**

The capability levels of a process area are achieved through the application of generic practices or suitable alternatives to the processes associated with that process area.

When one area is of a certain capability levels, it implies that all processes associated with it all have some underlying qualities:

- Level 1: Processes are performed and process goals are reached.
- Level 2: All the important aspects of the processes are addressed, the processes are planned, resources are provided, execution and products are controlled.
- Level 3: Consistency in all the processes is kept, based on organization standards for processes.

After raising the capability levels of the process areas, the organization has deemed to be most pressing. It is recommended in (CMMI for Development, Version 1.3) that the next target for improvement should be “high maturity process areas”:

- Organizational Process Performance (OPP).
- Quantitative Project Management (QPM).
- Causal Analysis and Resolution (CAR).
- Organizational Performance Management (OPM).

It is argued there that most benefit can be derived by first selecting the OPP and QPM process areas, and bringing those process areas to capability levels 1, 2, and 3. Because in doing so, there is a better coordination of projects and organizations the selection and analyses of processes and business objectives are better coordinated.

After improving the capability levels of the first two areas, it is recommended to focus on the CAR and OPM process areas. In doing so, the organization analyses the business performance using statistical and other quantitative techniques to determine performance shortfalls, and it identifies and deploys process and technology improvements that contribute to meeting quality and process performance objectives.

**Maturity Levels**

A maturity level is reached when the specific set of process areas defined for that level is at an acceptable level. A maturity level of a company is used as metric of its overall performance.

With each higher maturity level, additional process areas are made more mature which is also preparation for the next maturity level.

A maturity level is a defined evolutionary plateau for organizational process improvement. Each maturity level matures an important subset of the organization’s processes, preparing it to move to the next maturity level. The maturity levels are measured by the achievement of the specific and generic goals associated with each predefined set of process areas (CMMI for Development, Version 1.3).

The five maturity levels, each a layer in the foundation for ongoing process improvement, are designated by the numbers 1 through 5:
Chapter 2: State of the art

1. **Initial.** An organization that is at maturity level 1 usually creates a product that is required but with redundant approach and a lot of wasted effort. The processes are dependent on personal commitment of the people executing it and not on defined practices. Some of the recurring problems of level 1 maturity organizations are over commitment and tendency to not have predictable behaviour in crisis situations.

2. **Managed.** When an organization has reached maturity level 2, that means that the projects are well planned, executed according to the plan and description, monitored during execution, controlled, resources and personal with required skills are included in the process together with relevant stakeholder. At this level the processes also contain monitoring points used by the management to monitor the current stated of process execution. The work products and services satisfy their specified process descriptions, standards, and procedures.

3. **Defined.** At this level the organization has a set of standard processes which provides a basis for maturity. All the processes are adhering to these standards which creates consistency in the organization. Being at this level means that the process has been derived from the general guidelines defined by the organization and tailored for a specific need. At maturity level 3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods.

Like in the capability levels the crucial difference between levels 2 and 3 for maturity levels is the approach in the scope of standards, process descriptions, and procedures. At level 2 standards, process descriptions, and procedures are adapted to each specific area in order to maximize efficiency and effective in that particular case while at level 3 standards, process descriptions, and procedures are derived from more general organization guidelines. This means that the level 3 process artefacts are consistent across all the process areas which enables improvement, monitoring and control to be generalized all areas.

Processes are typically described more rigorously than at maturity level 2. A defined process clearly states the purpose, inputs, entry criteria, activities, roles, measures, verification steps, outputs, and exit criteria. At maturity level 3, processes are managed more proactively using an understanding of the interrelationships of process activities and detailed measures of the process, its work products, and its services.

At maturity level 3, the organization further improves its processes that are related to the maturity level 2 process areas. Generic practices associated with generic goal 3 that were not addressed at maturity level 2 are applied to achieve maturity level 3 (CMMI for Development, Version 1.3).

4. **Quantitatively Managed.** After reaching maturity level 4, the organization and projects establish quantitative objectives for quality and process performance and use them as criteria in managing projects. Quantitative and Qualitative objectives are measure using statistics and are based on the needs of the customer, end users, organization, and process implementers.

A form of data-mining on sub processes is executed. Statistical measures of process performance are made and then the data is interpreted through statistical tools as well as relationships between different sub processes. For
statistical measure of quality and objectives usually baselines and models are used.
The main difference between maturity level 3 and 4 is that at level 4 the processes are more predictable because of the statistical approach of monitoring and predicting based on previous experience.
5. Optimizing. The main addition at level 5 to the already stable and predictable processes is the constant improvement based on quantitative understanding of its business objectives and performance needs. Iterative improvement is applied and adapted for changes in the business environment, objectives and requirements. The project’s defined processes, the organization’s set of standard processes, and supporting technology are targets of measurable improvement activities. (CMMI for Development, Version 1.3).
The processes in the organization at maturity level 5 are ever evolving and adopting, knowledge is gathered from past projects and constant improvements are made.

As it can be seen in Table 2.1, maturity levels 2 and 3 use the same terms as capability levels 2 and 3. The purpose of this intentional equivalence is to highlight the fact that the capability and maturity levels are complementary. Maturity levels are higher in the improvement hierarchy and are used to characterize organizational improvement relative to a set of process areas, while capability levels characterize organizational improvement relative to an individual process area.

**Advancing Through Maturity Levels**

Advancing through maturity levels is done by continuous improvement of all the process related resources and management using qualitative and quantitative metrics to help the advancement. Maturity of an organization is used as a predictor for future stability and future projects outcome. As the organization progresses through the levels it reaps the benefits of previous improvements which positively affect all the process areas. The order in which organizations can perform improvements is not strictly defined. An improvement can be performed even if it is not required for the next maturity level, but a problem can occur when improvement in a certain process area requires other improvements as a base. Without a proper base the whole thing could collapse under pressure.

### 2.2 Agile software development

The basis of the agile development is the agile manifesto ([http://agilemanifesto.org](http://agilemanifesto.org)). The focus of agile manifesto and in turn of the agile software development is on the product, the customer and the developers. The developers are motivated and empowered and their main goal according to the agile manifesto should be creating business value by delivering quality working product in regular, short intervals. This should be achieved by relying on technical excellence and simple designs. Out of these basic principles, many new practices have been derived. The first of the common core principles, which connects all of the newly spawned practices, is the idea of self-organized teams that work in such a way both productivity and creativity, but also the sustainability of work pace, are kept at a high level. The second core principle is welcoming change, at any time in phase of the lifecycle of software development. This
principle is harder to follow and usually requires a lot of commitment in the organization using agile practices. The customers or their representatives are actively involved in all stages of the development process, their task being feedback and reflection for the purpose of getting the most complete and satisfying product.

The principles of agile are not a formal definition of agility, but are rather guidelines for delivering high-quality software in an agile manner.

Ever since the manifesto was articulated, practitioners and researchers have been trying to explicate agility and its different facets. At its core, agility entails ability to rapidly and flexibly create and respond to change in the business and technical domains (Henderson-Sellers and Serour, 2005; Highsmith and Cockburn, 2001). Other aspects of agility explored include lightness or leanness (i.e., having minimal formal processes) (Cockburn, 2007) and related concepts such as nimbleness, quickness, dexterity, suppleness or alertness (Erickson et al., 2005). In essence, these ideas suggest a “light’ methodology that promotes manoeuvrability and speed of response” (Cockburn, 2007).

2.3 Possibility of combined use

Agile and CMMI-DEV practices are often considered as opposites in the software development methodologies spectrum. Agile is seen as fluid and adaptable but lacking solid management and improvement strategies while CMMI is seen as rigid, costly to implement and maintain but with clearly defined goals and methods to reach them (H. Glazer, J. Dalton, D. Anderson, M. Konrad, S. Shrum, 2008).

The SEI (Software Engineering Institute) which created CMMI states that Agile development arose, in part, as a backlash to CMM and CMMI misuse and that Agile applies sounds Software Engineering principles in certain project contexts (CMMI & Agile, Michael D. Konrad, 2008). In early adoption of both approaches, there are extremes where CMMI is used for large-scale, failure resistant, mission critical systems while Agile extreme entails volatile requirements, dynamic conditions with lot of customer communication.

Since Agile did arise from a real need for change in some aspects, there are some problems which Agile addresses better or in more detail than CMMI. Lightweight approaches for small teams working tightly together with embedded customer are much closer to the Agile approach. Agile is also more “down to earth” and provides practitioners more voice in the processes they use and the customers more influence during development

Even the creators of CMMI acknowledge there are some shortcomings in the implementation of CMMI like “level mania” and imposing processes from outside (or exclusively top-down) as misuse (M. Konrad, S. McGraw, 2008). CMMI has been successfully and effectively implemented in big and mission critical systems, and when introduced correctly it provides many benefits like transparency, learning and reuse to the projects, processes and management.

The main advantages that CMMI has over Agile, according to (M. D. Konrad, 2008) are:

- Systems engineering (including risk management) practices that extend Agile to work in more complex situations.
Chapter 2: State of the art

- Align and coordinate across teams on: larger strategy, objectives, architecture choices, interfaces and changes.
- Maintain visibility into status, predicted completion, and risk.
- An infrastructure for organizational learning and improvement:
  - Benefits projects even before they start.
  - Supports use of processes, measurement, training, and improvement.
  - Reduces waste.
- A “safety net” that helps identify gaps and lapses in attention.
- Helps address lack of management support and resistance to change.

Agile methodology is an efficient way to develop software which gives liberty to change the plan according to the current conditions and situations. This type of strategy keeps the organization alive in the market but CMMI which is defined in late 1980s give the organization strength to bear and react in every type of situations. It is a way to make the high level management assure that the working will be done smoothly and in the end working product will be delivered.

When Agile methodology and CMMI are planned to work at the same floor, then there are some conflicts between the nature of Agile methodology and CMMI. Where Agile methodology emphasize on short iteration planning, CMMI focuses on process planning in a strategic manner. Agile methodology is a quick development method where communication and collaboration is the only way to overcome the problem, documentation is not considered as a part of development and just made before starting the development. On the other side, CMMI is a process to make the organization capable of doing things with stability at each step of maturity where documentation is an essential part to carry on the work. CMMI works to measure the performance and analyse the work, it also controls the whole process to deliver the product on time.

So both agile methodology and CMMI ultimately want to provide the product on time. At one side where Agile provides quickness in development, CMMI works to make the organization stable at the other side. Agile methodology and CMMI work combination could make the development quicker and assured. (S. K. Aggarwal, 2014).

2.4 Systematic Literature Review

The purpose of this Systematic Literature Review (SLR) is to summarize the existing evidence theoretical approaches and executed studies in the field of software process improvement with combination of CMMI-DEV and Agile approaches.

One of the key parts required by the SLR are explicit inclusion and exclusion criteria. As the name suggests the criteria are used to include and exclude previous works respectively from the final result of the SLR and are also essential in determining the value and the credibility of the review.

2.4.1 SLR process

The stages associated with conducting the SLR in software engineering, based on (Biolochni, Mian, Natali and Travassos, 2005), are:

1. Problem Formulation. In the first stage, the problem is formulated by creating definitions that allow the researcher to decide which studies are relevant and which are irrelevant in the context of the problem and which
evidence should be included in the review. During this stage there are threats to conclusion validity which should be kept in mind. One of them is using narrow concepts, which are hard to make robust, for conclusion. Another one is using excessive operations detail during definition construction, which could hide interacting variables.

2. Data Collection. During the second stage data collection is the focus of the review. Important part of this stage is searching for and choosing the sources from which potentially important studies shall be drawn as well as how this sources, if in different forms, should fit together for best outcome. Threat to validity at this stage could occur if studies which are not within the scope defined in the previous stage or do not have the focused research subjects as a target are used.

3. Data Evaluation and Studies Selection. In the third stage, collected data is evaluated in order to decide which evidence is considered relevant and should be included in the review. After that, quality criteria are applied to distinct which studies are considered valid and which are not. During this stage it is also decided which kind of information should be extracted from the filtered papers. Threats to validity in this stage are faulty quality criteria which could lead to dismissing valid studies and oversights in study reports that might lead to inaccurate conclusions.

4. Analysis and Interpretation. In the fourth stage, the collected data is processed. The key part of this stage is defining procedures to infer conclusions about the collected data as a whole from selected valid studies. Threats to validity at this stage come from procedures which could infer connections and causality in the data which are not objectively there.

5. Conclusion and Presentation. The fifth stage of the review finalizes the process. In this stage the report is formed using the relevant information that was chosen. The relevant information needs to be clearly separated from the rest via precisely defined criteria. A threat to validity here is not adhering to the previously defined procedure which would impede the reproducibility of the review and in turn nullify its conclusion.

The process shall be executed through the use of a template similar to the one defined in (J. Biolchini, 2005) (see Figure 2.1).

Next, the SLR is executed following the stages indicated previously.

2.4.1.1 Problem Formulation

1.1. Question Focus: the focus of the question is to find empirical or theoretical studies on applying CMMI for improving processes in a software development organization (SDO) already using agile practices.

1.2. Question Quality and Amplitude:

Problem: the SLR is done with the purpose of defining the current state of art and finding best practices for implementing CMMI-DEV improvement on agile oriented SDO.

• Questions:
  • Is CMMI-DEV applicable successfully on a SME using Agile?
  • What practices have been used so far on applying the CMMI-DEV based process improvement on Agile oriented SDO?
Keywords and Synonyms: CMMI-DEV, Agile, Agile practice, SCRUM, Kanban, SCRUMban.

Intervention: analyse agile practices based on the processes of Maturity Level 2 (ML2) related to CMMI-DEV v1.3.

Control: the baseline consists of similar SLRs that have been done in the past and some of them were found by consulting peers.

Effect: identify all approaches related to agile practices and their correspondence to CMMI-DEV.

Outcome Measure: number of approaches found for applying CMMI-DEV maturity level 2 on agile practices.

Population: the set of research proposals related to Agile and CMMI ML2 which have been published in the list of sources selected for conducting the systematic review.

Application: this SLR will be used as a starting point for deciding and developing an approach for the thesis.

2.4.1.2 Data Collection

2.1. Sources Selection Criteria Definition: the sources are chosen based on primary previous literature reviews into similar topics and with regards to database size, quality of search engine.

2.2. Studies Languages The only language that will be used is for the sources is English.

2.3. Sources Identification:

The sources used are based on the previously executed reviews (M. Palomino, A. Dávila, K. Melendez, M. Pessoa, 2017) as well as limitations on the available material through student account.

- Sources Search Methods: the search is conducted through web search engines provided by the sources’ websites using the search string.

- Search Strings:
  1. (Agile OR SCRUM OR kanban OR Scrumban) AND (CMMI-DEV) AND (small OR medium AND (enterprises OR organisations OR companies OR team OR firms OR settings)))
  2. ((((Agile) OR Scrum) OR Kanban) OR Scrumban) AND CMMI-DEV)

- Sources List:
  
  https://www.scopus.com/
  https://www.sciencedirect.com/
  http://ieeexplore.ieee.org/
  https://rd.springer.com
  https://dl.acm.org/

2.4 Sources Selection after Evaluation: sources are selected based on relevance and accessibility, as a student there are some limitations to source availability (not all papers are accessible through the university network without payment).

The protocol depicting the process of evaluation is shown in Figure 2.1
Chapter 2: State of the art

![Diagram](image)

**Figure 2.1 Template used**

2.5 References Checking: references checking shall be executed manually on the relevant papers.

2.4.1.3 Data Evaluation and Studies Selection

3.1 Studies Definition: relevant studies shall be extracted based on the inclusion and exclusion criteria:

- Inclusion criteria: academic articles with methodological basis or proposition. Articles from sources mentioned in the research were mainly focused papers from other sources only obtained through references processing. Also, articles in English language were considered. Finally, only articles that show or imply the combined use of agile and CMMI-DEV v1.3 approaches were considered and priority is taken by the articles describing the use of CMMI-DEV v1.3 in agile environment.

- Exclusion criteria: duplicated articles were excluded and the search scope was limited to the following publication types: journals, conferences, magazines, technical reports and books. In addition, we excluded the articles that only show the results of adopting agile practices without considering CMMI contexts or vice versa.

3.2 Procedure for Studies Selection: the procedure by which the studies were obtained and evaluated according to exclusion and inclusion criteria. Each initial paper is
processed by reading the Abstract and the Summary, if it is available, and looking for key words inside the paper. The initial number of papers and the number after applying the procedure is shown in Table 2.2

<table>
<thead>
<tr>
<th>Sources</th>
<th>Initial number of papers</th>
<th>Number after first processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopus</td>
<td>552</td>
<td>45</td>
</tr>
<tr>
<td>Science Direct</td>
<td>134</td>
<td>16</td>
</tr>
<tr>
<td>ACM</td>
<td>164</td>
<td>7</td>
</tr>
<tr>
<td>IEEE</td>
<td>55</td>
<td>18</td>
</tr>
<tr>
<td>Springer</td>
<td>670</td>
<td>24</td>
</tr>
</tbody>
</table>

*Table 2.2 SLR papers processed*

4. Quality Assessment


The following are the criteria used in the Quality Assessment:

- Is this study based on research?
- Is there a clear statement of the aims of the research?
- Is there an adequate description of the context?
- Was the study design appropriate to address the aims of the research?
- Was the selection strategy appropriate to the aims of the research?
- Was there a control group for comparing treatments?
- Was the data collected in a way that addressed the research aims?
- Was the data analysis rigorous enough?
- Has the relationship between researcher and participants been considered as an adequate degree?
- Is there a clear statement of results?
- Is the study relevant for practice or research?

According to (B. J. Shea, J. M. Grimshaw, G. A. Wells, M. Boers, N. Andersson and C. Hamel, 2008), these mentioned criteria include three important issues related to quality, which were considered in the Quality Assessment:

- Rigor: a complete and adequate approach was applied to key research methods in the study?
- Credibility: are the results in a meaningful and well-presented way?
- Relevance: how useful are the results to the software industry and the scientific community?

2.4.2 SLR Conclusion

The SLR was successful and managed to find the papers that answer the research questions and give a basis for going towards the goals of the thesis. In the following subchapter the exact
results of the SLR are shown and relevant papers that were directly used to answer the research questions are referenced.

2.5 Analysis and Interpretation

The results of the SLR are divided by research question they give an answer to. The first questions had a definitive positive answer, with only different levels of success in application and implementation. For the second question a big specter of different techniques was used but a dominating presence was the application of CMMI-DEV to SCRUM practices, so another additional result was generated, a CMMI-DEV process area to SCRUM mapping.

2.5.1 Is CMMI applicable successfully on a SME using Agile

SPI initiatives based on CMMI-DEV focus on improving software productivity rather than on delivering higher value to stakeholders (Boehm and Turner, 2003), which highlights differences between the principles behind SPI initiatives and those behind agile methods (R. M. Fontana, I. M. Fontana, P. Garubio, S. Reinehr, A. Malucelli, 2014)

Still, CMMI-DEV and the agile methods are compatible. At the project level, CMMI-DEV focuses at a high level of abstraction on what projects do, not on what development methodology is used, while SCRUM focus on how projects develop products. Therefore, CMMI-DEV and SCRUM can co-exist. There can be much value gained from SCRUM and CMMI-DEV synergies.

Conversely, CMMI-DEV can be effectively introduced in an Agile setting where an iterative, time-boxed approach is used, which is perfectly compatible with CMMI-DEV. After going through the papers gained through SLR, SCRUM was chosen to represent the combination of Agile and CMMI-DEV as the one most widely spread in cooperation with CMMI-DEV.

One of the conclusions from the SLR is that the most analysed Agile development methodology is SCRUM and since this coincides with the methodology used in the company in which the case study shall be performed, SCRUM will be used as a representative of the Agile development in the rest of the paper.

CMMI-DEV and SCRUM can complement each other by creating synergies that benefit the organization using them. SCRUM provides software development guidelines that are missing from CMMI-DEV best practices that work well—especially with small, co-located project teams. CMMI-DEV provides the systems engineering practices that help enable SCRUM on large projects. CMMI-DEV also provides the process management and support practices that help deploy, sustain, and continuously improve the deployment of SCRUM in small and medium organization. (Z. Lina, S. Dan, 2012)

An approach of combining SCRUM and CMMI-DEV in small and medium software companies is built up. The main idea is as follows:

A. Organizational Level Process

The focus of CMMI-DEV is on the structure and processes in an organization or enterprise. CMMI-DEV is most beneficial when it is implemented at organizational level, so that all functions and capabilities contributing to the development of products and services are addressed by the process improvement effort. As other agile methods, SCRUM does apply to the level of organization. So the Processes of organizational level should follow CMMI-DEV.
B. Project Management Activities

As can be seen in the next subchapter SCRUM does not cover all the specific practices of the project management process area, but it could be tailored to be more compliant with CMMI-DEV. On the other hand, processes based on CMMI-DEV model could be made more accessible by adding some SCRUM agile practices on their activities.

C. Risk Management

The purpose of Risk Management is to identify potential problems before they occur so that activities to handle the risks can be planned and used when needed. In SCRUM, risks are identified, but it does not have defined practices to define sources, parameters or categories to analyse and control the risk management effort. In SCRUM, there are no strategies for risk response or a mitigation plan for the critical risks based on historical data. This means that the assessment, categorization and priority of these risks occur in an informal manner. So risk management should be based on the CMMI-DEV.

D. Support Processes of Configuration Management and Product Quality Assurance

In SCRUM, there is no practice directly addressing these process areas. According to CMMI-DEV, Configuration management assures that the deliverable is reproducible, traceable, and approved for release. None of these attributes are guaranteed in SCRUM Product Quality Assurance in CMMI-DEV should support all process areas by providing specific practices for objectively evaluating performed processes, work products, and services against the applicable process descriptions, standards, and procedures and ensuring that any issues arising from these reviews are addressed. SCRUM has no such directly defined Quality Assurance and its more oriented to solving problems when they occur. So the support processes should be based on CMMI-DEV.

2.5.2 What practices have been used so far on applying the CMMI-DEV based process improvement on Agile oriented SDO?

There are 2 major approaches to combining Agile and CMMI-DEV (V. Henriques, M. Tanner, 2017):

Major Theme 1 - Agile/CMMI-DEV combination focusing on the Agile/CMMI-DEV theme where CMMI-DEV is used to formalize the agile practice or, agile is used to make the mature company more flexible.

Major Theme 2 – Agile Maturity where a new model is derived from by combing CMMI-DEV model with some of the agile principles and is used to assess maturity of agile SDO (V. Henriques, M. Tanner, 2017).

It is important to realize that is possible to reach maturity level 2 using agile methods if the following issues are addressed:

- Each process area must be explicitly defined within the organization. Agile methods provide almost all the required elements for this purpose.
- Process areas such as requirement management, measurement and analysis, quality assurance and configuration management should be complemented with elements obtained from other sources.

The weakest process areas in agile methods are requirement management, measurement and analysis, and product and process quality assurance. With respect to
configuration management, the effort should be even harder because agile methods cover only a little part of this area.

All processes must have the following basic elements:

- a clear policy, obtained from values and principles of the agile methods;
- workflows implementing in detail each part of the life cycle: artefacts, activities and roles, obtained from artefacts, practices and roles defined by agile methods.

Each process is particular and unique because it depends on the goals and the structure of the organization where it is applied. So, it is important to define a catalogue of process assets in a structured and organized way, and providing a series of application guidelines, so that a small organization can adopt and adapt them to be able to reach a CMMI-DEV certification (J. A. Hurtado, M. C. Bastarrica, 2006).

2.6 FINAL CONSIDERACIONES RELATED TO CMMI-DEV AND AGILE

During the conduction of the systematic literature review SCRUM has been chosen as a practices representing all Agile based practices. This choice was made due to an overwhelming presence of SCRUM as an Agile practice in the reviewed papers. Also it is the authors opinion that most of the Agile share their shortcomings in relation to CMMI-DEV and that a more specialized approach would be more beneficial during the development of process improvement approach.

Since SCRUM was chosen as a representing framework for all Agile approaches, a mapping between SCRUM practices and CMMI-DEV specific practices will be made using the information gained from the systematic literature review. The mapping will serve as a foundation for developing process improvement approach.

2.7 SCAMPI

The initial phase of the process improvement endeavour to be done will be based on the CMMI continuous integration and improvement. In this way, the Standard CMMI Appraisal Method for Process Improvement(SCAMPI) for CMMI 1.3 that has been tailored to the SCRUM approach will be used. First SCAMPI needs to be explored, the explanation of SCAMPI in this chapter is based on the official SCAMPI Manual (SCAMPI A, Version 1.3, 2011).

2.7.1 What Is SCAMPI A?

The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) is designed to provide benchmark-quality ratings relative to (CMMI) models. SCAMPI A satisfies all of the Appraisal Requirements for CMMI (ARC) and is the most rigorous of the Software Engineering Institute (SEI) process appraisal methods.

SCAMPI A enables a sponsor to:

- Gain insight into an organization’s capability by identifying the strengths and weaknesses of its current processes relative to appraisal reference model(s).
- Prioritize improvement plans.
- Focus on improvements (correct weaknesses that generate risks) that are most beneficial to the organization given its current level of organizational maturity or process capabilities.
Chapter 2: State of the art

- Derive capability level ratings as well as a maturity level rating.
- Identify risks relative to capability/maturity determinations.

Sponsors who want to compare an organization’s process improvement achievements with other organizations in the industry may have a maturity level determined as part of the appraisal process. This formal method enables determining the maturity level by objective measurements, in term decisions made on the basis of maturity level ratings are only valid if the ratings are based on known criteria.

The contextual information—organizational unit, organizational scope, reference model scope, appraisal method type, the identity of the appraisal team leader and the team—are items for which criteria and guidance are provided within the method to ensure a consistent interpretation within the community. With this approach the maturity of each process area is precisely and objectively determined.

2.7.2 Core Concepts and Approach

SCAMPI A relies on combining the information that is collected via defined approaches for gathering of objective evidence. The objective evidence is processed thought a series of transformations to get the end result. The appraisal team gathers the information that is transformed into notes, and then into characterizations of practice implementation gaps or compliance, which is sequentially transformed into preliminary findings. These findings are validated by the organizational unit before they become final findings.

2.7.3 SCAMPI A Methodology

SCAMPI A consists of three phases which are further extended into more detail by processes contained within them.

**Phase 1: Plan and Prepare for Appraisal**

Appraisal planning starts with understanding the sponsor’s objectives, requirements, and constraints. All other activities proceed from this initial activity. Significant investment and logistical planning involved are needed which also means considerable iteration and refinement of planning activities should be expected in phase 1. For smaller enterprises the planning is more straightforward and in term less time and resource consuming.

A team of trained personnel performs a SCAMPI A over a period of time negotiated by the sponsor and the appraisal team leader. The scope of the organization to be appraised, as well as the scope of process areas should be defined and agreed upon. The scope of the organization and model provides the basis on which to estimate personnel time commitments and overall costs to the appraised organization and to the sponsoring organization. For SMEs this cost is significantly smaller and the time needed for planning is greatly reduced compared to the large enterprises.

During the appraisal, the appraisal team verifies and validates the objective evidence provided by the appraised organization to identify strengths and weaknesses relative to the appraisal reference model. Before the Appraisal phase begins, members of the appraised organization typically collect and organize documented objective evidence,
using defined data collection strategies based on the extent of artefacts available within
the organization and aligned with the appraisal reference model.

The key to efficient execution of the appraisal is the advance preparation by both the
appraisal team and the appraised organization. Analysis of preliminary documented
objective evidence provided by the appraised organization plays an important role in
preparing for appraisal execution. If there are data inconsistencies or if substantial data
are missing at this point, subsequent appraisal activities can be delayed or even
cancelled if the judgment is made that continuing appraisal activities will not be
sufficient to make up for the deficiency given the resources available.

The collection of documented objective evidence, to some degree, by the appraised
organization in advance of the appraisal can help to improve appraisal team efficiency,
but it can also offer several other benefits to the organization:

• improved accuracy in appraisal results delivered by external appraisal teams (i.e.,
clear understanding of implemented processes, strengths, and weaknesses)
• obtaining a detailed understanding of how each part of the organization
participating in the appraisal has implemented model practices, and the degree
of compliance and tailoring of organizational standard processes
• establishing residual appraisal assets that can be reused on subsequent
appraisals, minimizing the effort necessary for preparation

A data collection plan, developed by the appraisal team leader in conjunction with the
appraised organization, can help make explicit the choices on how much data collection
effort to distribute between the organization and the appraisal team.

Phase 2: Conduct Appraisal

In phase 2, the appraisal team focuses on collecting data from the appraised
organization to judge the extent to which the model is implemented. The essential part
of this approach is the concept of coverage, which implies two things:

• the collection of sufficient data for each model component within the model
  scope selected by the sponsor,
• obtaining a representative sample of ongoing processes.

The samples taken out of the process areas being appraised are taken with 2 main goals:

• ensuring a representative sample of the organizational unit, and
• optimizing the effort for collection and analysis of objective evidence.

This means collecting data and information on all the appraisal reference model
practices in the appraisal scope, across sampled process instantiations within the
organizational unit being appraised. The data collection plan developed in phase 1
undergoes continuous iteration and refinement until sufficient coverage is achieved.

After determining that sufficient coverage of the process areas and organizational unit
has been obtained, appraisal findings and ratings can be generated. Goal ratings are
determined within each process area, which collectively can be used to determine
aggregate ratings for the individual process areas or for the organizational unit, as
appropriate.

Phase 3: Report Results
Chapter 2: State of the art

In phase 3, the appraisal team provides the findings and ratings to the appraisal sponsor and the organization. These artefacts become part of the appraisal record, which becomes protected data in accordance with the appraisal disclosure statement. The level of protection and the plan for the disposition of appraisal materials and data are determined in phase 1 in collaboration with the sponsor.

2.8 AFIM

Action Focus Improvement Model (AFIM) is a process improvement lifecycle that contains the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) for the first two phases of the lifecycle. SCAMPI is used for getting the commitment and appraisal needed for the improvement lifecycle.

The AFIM has well defined steps for each part of the lifecycle with precise calculations of cost and effort needed.

The Figure 2.2 depicts the phases of the AFIM lifecycle. As it can be seen in the figure, the first 2 phases are covered by SCAMPI.

The third phase is Infrastructure and Plans for Software process improvement (SPI). During this phase the personnel infrastructure that will create and implement SPI is created as well as plans for the process improvement.

The fourth phase is the Implementation of the SPI. In this phase the plans for improvement are executed and new or improved processes are implemented.
2.9 FINAL CONSIDERATIONS OF SCAMPI

Successful application of SCAMPI relies on adjusting the parameters of the method to the needs of the organization and to the objectives and constraints of the sponsor’s organization. An additional challenge which is addressed in this paper is tailoring SCAMPI to the needs of an Agile organization.

The reference model scope and representation (staged or continuous), the size of the organizational unit, the parts of the organization sampled, the size of the appraisal team, and the number of interviews greatly influence things such as preparation time, time on site, and monetary costs. All of the motioned included things are major factors when choosing tailoring options.

Tailoring provides flexibility to efficiently adapt the appraisal to the needs of the appraisal sponsor, within acceptable limits allowed by the appraisal method. The appraisal team leader is responsible for ensuring that the requirements of the method are satisfied. Tailoring the method too severely could result in failure to satisfy method requirements, the inability to obtain sufficient data for generation of appraisal findings or ratings, or failure to meet the criteria necessary for recognition as a valid SCAMPI A appraisal. The detailed tailoring steps for SCAMPI will be described in the next chapter.
Chapter 2: State of the art
Chapter 3: Resolution

3. Resolution

In this chapter the planning and execution of process improvement with CMMI adapted for SCRUM as an underlying model is explained in detail.

The resolution of the suggested problems is made by combining multiple process improvement approaches and tailoring them to the needs of a specific target group, in this case SMEs that have embraced Agile development. The first part of the improvement approach is focused on getting the commitment for the improvement approach and executing the appraisal of the current state of processes in the targeted company. This part of the improvement approach is based on the standard appraisal method for CMMI-DEV process improvement (SCAMPI) which also gives good guidelines for obtaining commitment for the improvement approach. SCAMPI is tailored to the needs of a SME with Agile environment.

The second part of the improvement approach contains planning, implementing and measuring the implemented improvements. This part of the improvement approach was created by combining the two process improvement lifecycles: AFIM and IDEAL with the information and recommendations gained from the SLR that was described in the previous chapter. The CMMI-DEV is used in this part of the improvement approach as a starting point for categorizing and suggesting improvements.

In the next subchapter SCAMPI is tailored to the needs of a SME that has embraced the Agile paradigm for software development.

3.1 Tailoring SCAMPI A

In this sub chapter the methodology used for appraisal will be explained together with particular activities that will be taken to execute the appraisal. All the necessary limitations and guidelines needed for a correct appraisal will be listed.

Implementing and maintaining any kind of process improvement in a SME is by itself a challenge that can result in waste of resources if not executed properly (N. Baddoo, T Hall, 2003). When other companies have had previous bad experiences with process improvement, it may sometimes result in some software companies choosing to implement SPI in response to negative business events alone (G Coleman, R O'Connor, 2008).

As mentioned in the previous chapter SCAMPI A satisfies all of the Appraisal Requirements for CMMI (ARC) and it is the most rigorous of the Software Engineering Institute (SEI) process appraisal methods. This also means it is formalized and has many phases, actions, and steps to execute. Executing the SCAMPI as it is defined clashes with the practices of a SME relying on Agile environment. It would be nearly impossible to maintain the improvement devotion of the members as well as the management throughout all the of SCAMPI. That is why SCAMPI needs to be tailored to an Agile environment.

Tailoring in this case means taking the essential actions of SCAMPI and modifying them to be more applicable to the needs of a SME with Agile environment. The tailoring was done by combining SCAMPI with phases from AFIM and IDEAL, while also using additional papers about SPI that will be referenced in the chapter.
### 3.1.1 Mapping CMMI process areas to SCRUM

Using the data gained from the SLR papers, mostly relying on (C. R. Jakobsen, & J. Sutherland, 2009; J. Diaz, J. Garbajosa, & J. A. Calvo-Manzano, 2009; A. S. C. Marcal, F. S. F. Soares, & A. D. Belchior, 2007), the following tables were created, to serve as a guideline for possible improvements in the companies already using SCRUM. CMMI process areas were used as a starting point since CMMI has more structure and formality and then the SCRUM practices were mapped onto them. Paper (A. S. C. Marcal, F. S. F. Soares, & A. D. Belchior, 2007) contains an estimation of CMMI practice coverage with pure SCRUM practices, the different process areas were differently covered ranging from completely covered, partly covered to not covered at all. In the Tables 3.1 – 3.4 this is represented with different underlying colours. The completely uncovered sub process areas need a completely new process to fill the gap, the partly covered areas need at least an upgrade of existing processes and the completely covered areas just entail upholding the practices already defined by SCRUM.

The part of the tables that was not colour coded was based of the two remaining papers in combination with authors personal experience from being part of SCRUM based environments. Some of the mappings are because of minor intersections between the CMMI process areas and the SCRUM approach practices.

<table>
<thead>
<tr>
<th>CMMI PROCESS AREA</th>
<th>Specific Goal</th>
<th>PRACTICES</th>
<th>SCRUM PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Monitoring and Control (PMC) (CMMI – DEV)</td>
<td></td>
<td>1.1 Monitor project planning parameters</td>
<td>Burn down graphs and project meetings</td>
</tr>
<tr>
<td>1. Project management</td>
<td></td>
<td>1.2 Monitor commitments</td>
<td>Sprint planning meeting and monitored through sprint burn down and daily meetings and, finally, reviewed in the sprint retrospective meeting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.3 Monitor project risks</td>
<td>Daily meeting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Risks are registered on white boards, flip charts or impediment lists</td>
<td></td>
</tr>
<tr>
<td>1.6 Monitor project risks</td>
<td>1.4 Monitor data management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5 Monitor stakeholder involvement</td>
<td></td>
<td>Project meetings by the SCRUM master</td>
</tr>
<tr>
<td></td>
<td>1.6 Conduct progress reviews</td>
<td></td>
<td>Frequent inspections and progress review meetings (the</td>
</tr>
<tr>
<td>CMMI PROCESS AREA</td>
<td>SPECIFIC GOAL</td>
<td>PRACTICES</td>
<td>SCRUM PRACTICE</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Project Planning (PP) (CMMI - DEV)</td>
<td>1. Establish</td>
<td>1.1 Estimate the scope of the project</td>
<td>Product backlog and the set of all pre-defined sprints, beginning of sprint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Manage corrective action to closure</th>
<th>1.7 Conduct milestone reviews</th>
<th>Sprint review meetings, the project progress is inspected, providing visibility of the accomplishment of commitments.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.1 Analyse issues</td>
<td>During the daily SCRUM meetings, the team reports all impediments against expected quality or performance levels. Impediments are registered on a white board, flip chart or impediment list and they are erased when overcome.</td>
</tr>
<tr>
<td></td>
<td>2.2 Take corrective action</td>
<td>Corrective actions are taken for the impediments found. However, there is not any register of how these actions are planned and monitored</td>
</tr>
<tr>
<td></td>
<td>2.3 Manage corrective action</td>
<td>All corrective actions are monitored to closure. However, the results of these actions are not analysed to determine its effectiveness</td>
</tr>
</tbody>
</table>

*Table 3.1 Project Monitoring and Control mapping to SCRUM*
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>SCRUM Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>Establish estimates of work product and task attributes</td>
<td></td>
</tr>
</tbody>
</table>
| 1.3  | Define project lifecycle | • Planning  
• Staging  
• Development  
• Release |
| 1.4  | Determine estimates of effort and cost | Product backlog and sprint backlog estimates |
| 2.1  | Establish the budget and schedule | Product backlog and directly derived from the estimated effort |
| 2.2  | Identify project risks | Daily meetings and registered on white-boards, flip charts or impediments |
| 2.3  | Plan for data management | Meetings or documents |
| 2.4  | Plan for project resources | Staging phase and backlog |
| 2.5  | Plan for needed knowledge and skills | Teams are multi-functional groups, self-managed, and made up of seven skilled people implementing the sprint backlog items |
| 2.6  | Plan stakeholder involvement |  |
| 2.7  | Establish the project plan | Vision and a product backlog |

*Table 3.2 Project Planning mapping to SCRUM*
<table>
<thead>
<tr>
<th>CMMI PROCESS AREA</th>
<th>SPECIFIC GOAL</th>
<th>PRACTICES</th>
<th>SCRUM PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Management (REQM) (CMMI – DEV)</td>
<td>Manage Requirements</td>
<td>1.1 Understand Requirements</td>
<td>User Stories in an iterative way (sprints)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2 Obtain Commitment to Requirements</td>
<td>Planning meetings. Backlogs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.3 Manage Requirements Changes</td>
<td>Planning and Review meetings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.4 Maintain Bidirectional Traceability of Requirements</td>
<td>User Stories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.5 Ensure Alignment Between Project Work and Requirements</td>
<td>Pre-game and Planning meetings</td>
</tr>
</tbody>
</table>

Table 3.3 Requirements Management mapping to SCRUM
<table>
<thead>
<tr>
<th>CMMI AREA</th>
<th>SPECIFIC GOAL</th>
<th>PRACTICES</th>
<th>SCRUM PRACTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management(CM)</td>
<td>1. Establish Baselines</td>
<td>1.1 Identify Configuration Items</td>
<td>When working on a project configuration items are identified in the product backlog</td>
</tr>
<tr>
<td>Configuration Management (CM)</td>
<td></td>
<td>1.2 Establish a Configuration Management System</td>
<td>A tool or group of tools is used to establish a CM system</td>
</tr>
<tr>
<td>Configuration Management (CM)</td>
<td></td>
<td>1.3 Create or Release Baselines</td>
<td>A burn down chart used as a baseline for</td>
</tr>
<tr>
<td>(CMMI-DEV)</td>
<td>2. Track Changes</td>
<td>2.1 Track Change Requests</td>
<td>Change requests are discussed during sprint planning and sprint review</td>
</tr>
<tr>
<td>Configuration Management (CM)</td>
<td></td>
<td>2.2 Control Configuration Items</td>
<td>Configuration items are monitored using external tools</td>
</tr>
<tr>
<td>(CMMI-DEV)</td>
<td>3. Establish Integrity</td>
<td>3.1 Establish Configuration Management Records</td>
<td>Tools are used to establish configuration management records</td>
</tr>
<tr>
<td>Configuration Management (CM)</td>
<td></td>
<td>3.2 Perform Configuration Audits</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4 Configuration management mapping to SCRUM
3.1.2 Creating the questionnaires

The existing mapping created in the Tables 3.1 – 3.4 was used as a foundation for creating questionnaires. The questionnaires’ were created in accordance with CMMI-DEV process areas with each question being mapped onto one sub practice of the process area. Each question was created by using the mapping from the table as well as authors own theoretical and practical knowledge about SCRUM and CMMI, and the guidelines from the SCAMPI. In the next chapter it will be described how well it was understood and received by people who never encountered CMMI before and are used to the SCRUM practices. Examples of the questions created for different process areas are shown in Table 3.5

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Sub practice</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management</td>
<td>3.1.2 Ensure that relevant stakeholders have access to and knowledge of the configuration status of configuration items.</td>
<td>Product owner ensures that relevant stakeholders have access to and knowledge of the configuration status of configuration items.</td>
</tr>
<tr>
<td>Project Monitoring and Control (PMC)</td>
<td>1.2.2 Identify commitments that have not been satisfied or are at significant risk of not being satisfied.</td>
<td>Commitments that have not been satisfied or are at significant risk of not being satisfied are identified during each sprint.</td>
</tr>
<tr>
<td>Project Planning (PP)</td>
<td>1.4.1 Collect models or historical data to be used to transform the attributes of work products and tasks into estimates of labour hours and costs.</td>
<td>During sprint reviews and sprint retrospective historical data is used to improve estimates of labour hours and costs.</td>
</tr>
</tbody>
</table>
3.1.3 Commitment to SPI

In the developed process improvement approach the commitment to the execution of the improvement and the appraisal of the current state of processes in the SME are based on the SCAMPI method. Since SCAMPI was created to work with CMMI it fits perfectly with the final goal of the actions taken.

The discussion about performing process improvement in a company begins with two people: the appraisal team leader who will oversee the whole appraisal and report to all involved stakeholders; and the sponsor, a member of senior management in the enterprise wanting to perform the improvement who champions the executions to the rest of the senior management.

The first phase of executing the appraisal requires obtaining commitments from all the stakeholders involved. Appraisal planning starts with understanding the sponsor’s objectives, requirements, and constraints. All other activities proceed from this initial activity. The entity in charge of the appraisal must understand business needs of the organization for which the appraisal is being requested.

The activities to be executed in this phase are:

- Analyse the Requirements based on business objectives.
- Develop the Appraisal Plan.
- Train the Assessment Team.
- Analyse the risks.

3.1.1.1 Analyse the requirements based on business objectives

The starting point of this step is to identify the sponsor of the whole endeavour form the senior management. They will need to advocate for the appraisal and support it during execution. After getting the support of a senior management sponsor, the objectives of the appraisal need to be determined. The base for appraisal objectives are the business objectives (reduce costs, improve quality, reduce time to market).

When definition of the appraisal goals is complete, they are discussed with the senior management and again compared against current business objectives.

In this step the cost and schedule also need to be defined, in broad strokes as a baseline for future estimations and to give the senior management a general idea of effort needed.

The appraisal scope consists of the appraisal reference model scope and the organizational scope to be examined during the appraisal. The reference model scope
must be determined and documented early in the planning process, identifying the relevant models and process areas.

For CMMI model that is being used here, the scope specification includes the continuous representation (and the capability levels included). In conjunction with the appraisal sponsor, the appraisal team leader is responsible for deciding which process areas to include in the scope of the appraisal and which model representation to use.

An organizational unit is the part of an organization that is the subject of an appraisal and to which the appraisal results will be generalized. This organizational unit may include the entire organization, one or more divisions within the organization, or one or more basic units and support functions within the organization.

3.1.1.2 Develop the Appraisal Plan

The appraisal plain is one of the essential parts of the appraisal. Developing a good appraisal plan with a good team is necessary for executing an efficient and effective appraisal.

The appraisal sponsor or senior site manager may identify candidate appraisal team members and appraisal participants. Review of the organizational unit structure or other site-specific information can also be useful for this identification. The participants can be specified in terms of roles or responsibilities initially, with specific names to be determined later.

Equipment and facilities are often negotiated with the organizational unit where the appraisal activities will be performed, but sometimes these equipment and facilities need to be acquired. A room allocated to the appraisal team is usually needed for private discussions and to protect the confidentiality of appraisal data. If possible, this room should be separate from the other rooms where interview sessions are held.

Cost and schedule may be developed using one of the three approaches:

- top-down based on sponsor objectives and constraints,
- bottom up based on results of other planning and preparation processes and activities,
- more generally using a combination of the two previous approaches.

Determining and communicating a schedule for the appraisal, and maintaining ongoing visibility as the details take form, are the primary responsibility of the appraisal team leader. Scheduling is an ongoing logistical task that requires the coordination and cooperation of many different groups of individuals.

A balance needs to be made between the needs of the sponsor for appraisal outputs against the resources available to conduct the appraisal, which determine the schedule constraints. Schedule and cost must be considered for the entire span of the appraisal activities. Effort estimates should be developed not only for the appraisal team, but also for the expected participants within the organizational unit.

The costs of multiple appraisals can be reduced by gathering and maintaining objective evidence for each instantiation. In addition to providing an effective mechanism for monitoring the process implementation and improvement progress of the organizational unit, this approach enables the ready availability and reuse of objective evidence for subsequent appraisals.
Chapter 3: Resolution

The schedule for the appraisal needs to be shared with a fairly wide audience because of all the parties involved, but the cost of the appraisal is often kept from wide view, due to the potentially sensitive nature of this information.

The logistical details of the appraisal are negotiated and documented. The appraisal team leader, supported by the appraisal coordinator, if the roles are not combined, manages planning tasks that document and communicate logistical arrangements. The tools commonly used for keeping track of these tasks are checklists and action item tracking mechanisms.

In the end of the appraisal planning **formal sponsor commitment** to the appraisal plan must be obtained. The appraisal plan constitutes a contract between the appraisal sponsor and the appraisal team leader, so it is vital that this agreement be formal. Without the formal agreement the whole appraisal would be at risk to not be executed correctly or to never be completed.

### 3.1.1.3 Training the assessment/appraisal team

To ensure a successful appraisal the team leader needs to gather a capable and knowledgeable team which can execute the appraisal correctly and efficiently. The appraisal team leader can choose to train a new team or form a team from team members who already have experience with similar endeavours and have cooperated on it, depending on the context and the responsibility to ensure that the team is ready to succeed rests with the appraisal team leader.

In the previous chapter the topics that are addressed in training in SCAMPI were defined here, because the context is a SME with agile practices some of them can be omitted and the essential ones kept.

- Teamwork & Human Factors: Changing ways: resistance to change, roles for change (inventors, sponsors, integrators, entrepreneurs, managers, experts), barriers to change, etc.; Personality profile (extroverted/introverted, sense/intuition, judging/perceiving, etc.); Facilitating Meetings: decision techniques, roles; Commitment process.
- Standard to be compared (CMMI-DEV v1.3 and SCRUM in this case) & Assessment Method.

The trained team members continue learning and improving throughout the appraisal process while being monitored by the team leader. Any conflicts and disagreements on appraisal plan and execution need to be discussed as they arise, and the team leader has the responsibility of maintaining the proper approach to the appraisal execution.

### 3.1.1.4 Analysing the risks

The appraisal as a complex and multi-layered event contains dependencies among events, people, and other resources, which means risk management needs to be planned and executed together with the rest of the appraisal. This is done by the appraisal team leader who is responsible for documenting and communicating risks and associated mitigation plans to the sponsor and appraisal team members.

Risk management is most effective when performed early in the appraisal planning process and is a continuing responsibility of the appraisal team leader throughout the appraisal. The level of effort devoted to risk-management activities is something the appraisal team leader must adjust to fit the situation.
3.1.4 Execution of appraisal

The first thing that needs to be done before the appraisal starts is to ensure that appraisal participants are appropriately informed of the appraisal process, purpose, and objectives and are available to participate in the appraisal process. Members of the organization who participate in the appraisal must be informed of their role and the expectations the sponsor and appraisal team have. Being well informed about the appraisal enables the appraisal to be less time and effort consuming as well as not to interrupt ongoing work. This communication is typically accomplished through a briefing in which the appraisal team leader provides an overview of the appraisal process, purpose, and objectives. Specific information about the scheduled events and the locations where they occur is also communicated during this presentation, as well as through ongoing contact between the appraisal coordinator and the members of the organization.

The goals of the appraisal are:

- Determining the strengths and weaknesses of an organization in relation to the CMMI-DEV - SCRUM.
- Identifying a small number of process improvements in key areas for the organization.
- Facilitating the beginning of a software process improvement effort (PI Project).

The appraisal is executed through questionnaire with the project leaders (PLs) and interviews with the PLs and other employees who are part of the appraisal scope, then the findings are presented and process area to be improved are selected. The following activities are carried out:

- Workshop and questionnaire.
- Preliminary response analysis.
- Preliminary selection of processes (PAs).
- Analysis of inconsistencies in the answers.
- Findings.
- Selection of processes to be improved and presentation of results.
- Action Plan Guide.

Workshop and questionnaire

There are multiple questionnaires, each questionnaire related to a specific process area that is being appraised. Project leaders that have projects within the selected appraisal scope fill out the questioners after being part of a workshop that explains the specific process area and how to give answers.

The questionnaires are based on the CMMI- DEV 1.3 – SCRUM model practices and sub practices and determine the level of compliance with the model for a specific process area. The questionnaires represent a mechanism of initial data collection for software process assessment and a framework for further discussions among the participants of the evaluation and assessment team members.

Each question in the questionnaire can be mapped to one specific sub practice, and the answers to the questions are based on the ISO 15504 standard with 4 possible discreet answers:
Chapter 3: Resolution

<table>
<thead>
<tr>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully achieved (F)</td>
</tr>
<tr>
<td>86-100% achieved</td>
</tr>
<tr>
<td>Largely achieved (L)</td>
</tr>
<tr>
<td>51-85% achieved</td>
</tr>
<tr>
<td>Partially achieved (P)</td>
</tr>
<tr>
<td>16-50% achieved</td>
</tr>
<tr>
<td>Not achieved (N)</td>
</tr>
<tr>
<td>0-15% achieved</td>
</tr>
<tr>
<td>Don’t know (blank)</td>
</tr>
</tbody>
</table>

*Figure 3.1 ISO 15504 standard discreet answers*

The assessment/appraisal team focuses on collecting data from the appraised organization to judge the extent to which the model is implemented. The essential part of this approach is the concept of coverage, which implies two things:

- the collection of sufficient data for each model component within the model scope selected by the sponsor,
- obtaining a representative sample of ongoing processes.

The samples taken out of the process areas being appraised are taken with 2 main goals:

- ensuring a representative sample of the organizational unit, and
- optimizing the effort for collection and analysis of objective evidence.

This means collecting data and information on all the appraisal reference model practices in the appraisal scope, across sampled process instantiations within the organizational unit being appraised.

After determining that sufficient coverage of the process areas and organizational unit has been obtained, appraisal findings and ratings can be generated. Goal ratings are determined within each process area, which collectively can be used to determine aggregate ratings for the individual process areas or for the organizational unit, as appropriate.

Once objective evidence on practice implementation has been verified, the team turns to characterizing the implementation of model practices. For each model practice, and each instance sampled, the team will document a characterization of the extent to which the model practice (or an acceptable alternative) has been implemented. The implementation-level characterizations are then aggregated to the organizational unit level.

Characterizations of practice implementation are used as a means to focus appraisal team effort on areas where professional judgment is needed, and to aid in reaching team consensus on the extent to which practices are implemented.
Chapter 3: Resolution

Preliminary response analysis

With all the questionnaires filled, a significant amount of necessary information is given to the appraisal team enabling them to estimate the general level of compliance of the different processes assessed and providing them with a comprehensive understanding of the range of organizational process maturity and granting them an overview of the strengths and weaknesses of each process.

In preparation for validating the verified information, the appraisal team generates preliminary findings that summarize potential strengths or weaknesses that support judgments about practice implementation. The preliminary findings are written in reference to the CMMI-DEV-SCRUM model practice.

The next step is further exploration of the strengths and weaknesses of the particular process areas. That is done by examining the weighted averages for each question to determine whether those are areas of strength or weakness for the project based on a predetermined percentage coverage of the process area (e.g., higher or lower than 75%).

Standard deviation of the answers to a particular question need to be lower or equal to 1. If the deviation is higher, it represents an inconsistency and needs to be resolved before it can be taken into the final findings.

**Preliminary selection of process areas (PAs)**

The initial questionnaires are the basis upon which the rest of the appraisal stand, the preliminary process area for improvement are based on the questionnaires. The main criteria for selecting a preliminary process area are:

- Business objectives.
- Coverage Level (lower than the passing threshold).
- Dependencies among PAs.

When the preliminary PAs have been selected, they are followed by interviews related to the PAs. The Interviews are conducted with senior management, project leaders and professionals working under the PLs. The Figure 3.2 was taken from AFIM and it depicts the process of selecting preliminary process areas. The depicted process was loosely followed without strict interview scripts.
Analysis of inconsistencies in the answers

The inconsistencies in the answers arises when the deviations between the answers of for the same PA or practise is big. That means that the difference in views between the different participants is big and further exploration is needed.

The interviews represent the core of the assessment process because they allow organizational software practitioners to express their views on the existing software process. Gathering opinions during interviews expands the knowledge gained from the responses of PLs to the questionnaires and begin the thinking process about the real improvements that should be implemented. All interviews are closed type, meaning only the participant and the Assessment Team members are present. Confidentiality needs to be kept and privacy guidelines followed. The format of the interview is standard with open questions combined with driven follow-up questions in a collaborative approach.

Findings

After generating results data through appraisal processes execution, the generated data needs to be evaluated in order to determine the maturity of the particular process areas. The purpose here is to rate goal satisfaction based on the extent of practice implementation throughout the organizational scope of the appraisal.

During this phase the data collected during the analysis of inconsistencies in the answers and the assessment team analysis is synthesized. The overall collected information is used to represent the views of the assessment team as well as provide the initial point for formulating recommendations.

Initial organizational capabilities are identified, and recommendations are given for the future recommended actions during this phase as well.

Feedback sessions with all the participants are held in order to:
Chapter 3: Resolution

- Confirm the findings.
- Discuss and determine the related consequences.
- Formulate recommendations.
- Protect confidentiality.

The judgment of goal satisfaction is based on and traceable to the extent of the implementation of practices associated with that goal (or alternative practices contributing equivalently to goal satisfaction).

Success in this phase is driven by team members’ ability to limit their focus to the data that support the judgments, and to avoid issues that threaten their ability to be objective. This activity can create a great deal of stress for team members under pressure to help their organization “do well.” The appraisal team leader must skilfully facilitate this activity when external pressures exist.

The judgments made about goal satisfaction are driven by the preliminary findings that were documented by the appraisal team and validated by appraisal participants as well as the extent of implementation of associated practices. The preliminary findings focus on exemplary practice implementations as well as gaps in the implementation of practices. When performing goal ratings, the team judges whether or not these weaknesses in the implementation of practices (in aggregate) threaten the organizational unit’s ability to satisfy the goals associated with the practices.

The appraisal team may produce ratings for process areas. Assigning process area ratings is an optional activity that is selected at the discretion of the appraisal sponsor and documented in the appraisal plan. Depending on the reference model chosen, the process area ratings may reflect a staged architecture (satisfied vs. unsatisfied) or a continuous architecture (capability levels 0 to 3).

**Selection of process areas to be improved and presentation of results**

With inconsistencies solved and the process areas appraised, the process areas to be improved are determined based on the preliminary selected areas and additional information gathered. The next action to be taken is meeting with managers in order to present strengths, process capabilities, outcomes, and consequences and recommendations of the assessment together with the business objectives.

After the ratings have been given and the appraisal finished it needs to be accepted by the team leader and the sponsor. After it has been accepted the assessment team discusses it with the senior management who also need to accept the results of the appraisal. The process improvement can start with a sound basis of the appraisal.

**Action Plan Guide**

During this phase a transition needs to be made from the formal assessment results towards the improvement (action) plan. New ideas and perspectives related to the process improvement can be introduced while in this phase. It serves as interface between the assessment and the beginning of the process improvement implementation.
3.2 Infrastructure and Actions Plans

With the assessment finished and process areas to be improved selected, planning and implementation of the process improvement lifecycle will be taken from AFIM. The Action Focus Improvement Model has been partly tailored to the needs of a SME with an already existing SCRUM practice. In this phase direct process improvement shall be addressed. That means creating plans for future improvement and applying them. The people connected to the process improvement are connected into an improvement infrastructure and together they follow improvement plans.

![Process Improvement Infrastructure](image)

**Figure 3.3 Improvement infrastructure from AFIM**

3.1.5 Infrastructure

The people involved in the process improvement endeavour are divided into:

- Management Steering Committee.
- Steering Committee.
- Software Engineering Process Group (SEPG).
- Working Groups (WG).

There is no need for liaisons because in SMEs with Agile environment the communication and division of task is more individual oriented, which means that there is no need for explicit representative of the working practitioners.

**Management Steering Committee**

A Management Steering Committee is made of senior management members and it provides support and credibility at the start of the improvement process. It also does the task of providing the required rationale of how the changes due to process improvement fit both in the culture and in the vision of the organization. This committee focuses on including the change effort to improve, within the goals of the organization:
Chapter 3: Resolution

- Expresses the entire organization the importance and need for change.
- Ensures that the resources needed for the working groups, pilot teams and SEPG are available when they are needed.

**Steering Committee**

This is an advisory committee usually made up of high level stakeholders and/or experts who provide guidance on key issues in the company. The committee specifies the technical and management activities to support the process improvement effort and oversees establishing Working Groups to meet specific organizational process improvement and supporting interaction between SEPG and the projects. It is also in charge of monitoring progress and conducts periodic follow-up reviews and analysing software process information provided by the Working Groups and the SEPG. In addition, one of its roles is also to advise and give recommendations to the Management Steering Committee.

**Working Groups (WG)**

A Working group oversees one process area to be improved. It develops the action plan for that specific area. The developing of the action plan entails:

- Develop the software process definition.
- Develop the appropriate standards and procedures of software.
- Identify, demonstrate and evaluate technologies.
- Develop and / or suggest training plans.

Additional commitments of the each working group is to report to the Steering Committee about action plan progress and to work with the SEPG to transfer new technologies throughout the organization.

The enthusiastic people from the organization are the first choice for working groups. When choosing the preliminary improvement areas, people prioritize improvements based on their enthusiasm for the improvement area. No commitment is implied at that time, however. Now that the improvement areas have been identified, the same people should be contacted to see if they are still interested. Their commitment and the commitment of their managers must be secured for them to work on the team.

**SEPG**

This Definition of SEPG is directly based on the definition from IDEAL (B. McFeeley, 1996) with consideration of needs of SME. The SEPG is the focal point for the organization’s SPI program. It is responsible for and facilitates the activities that relate to software process improvement, such as action planning, process improvement, technology improvement, and other activities. The SEPG coordinates and plans all of the organization's SPI programs. The SEPG also leads the organization's improvement efforts.

The SEPG must obtain and maintain management support for the initiative at all levels and across all functionality. The SEPG is assisted in accomplishing this by working with the Management steering committee(MSC) to demonstrate commitment to practitioners and management of the organization.

The SEPG will facilitate software process assessments and, along with the organization’s management and practitioners, will develop the SPI strategic action plan to guide the
efforts. The SEPG will also facilitate other base lining activities to provide definition for existing process definitions and measurement activities.

Another activity of the SEPG is to monitor all of the SPI activities in the organization. The SEPG will report the status of the various improvement activities that are in progress to the MSC. Timely reporting of SPI status will allow the MSC to make informed decisions that will support and enhance the success of the SPI program.

The SEPG needs to arrange or conduct training in process improvement and continuing education in other subjects relevant to the SPI program. Characteristics of members of the SEPG include experience as a software development practitioner, sound knowledge in one or more domains, and respect of their peers in the line organizations.

The SEPG members must support the SPI program, championing it to the rest of the organization. They must also have the capability to effectively serve as agents of change as new and improved processes and technologies are introduced to the organization. SEPG members are critical to the success of the SPI program.

3.1.6 Process Improvement Plan (PIP)

This guide recommends that 1-3% of an organization's personnel be applied to managing and executing SPI. SMEs require a strong commitment to sustain a SPI effort. Scale issues need to be addressed for organizations of this size, as they are probably not complex enough to warrant the typical SPI infrastructure. Regardless of size, it is recommended that at least one person be applied full time to facilitate and execute SPI (B. McFeeley, 1996)

The steps to be taken during the creation of process improvement plan are

- Establish Infrastructure.
- Provide Generic & Specific Training to WGs.
- Prioritize processes.
- Action Plans: (WG activities -for each WG)

This plan is a high-level plan with broad goals that outlines the SPI initiative that the organization will be following. Responsibility for developing this plan is shared among the steering group, the management steering group and the software engineering process group. Senior management has approval responsibility for this plan.

The purpose is to provide an overview and some guidelines to create the Process Improvement Plan (PIP) of the Organization. The PIP is a formal approach to create and implement new or improved processes that allow for or support the software development:

- will document a set of activities and their associated priorities that are aligned with strategic business objectives.
- shall contain the budget related to time, money and resources needed to conduct the planned activities.
- SEPG is responsible for coordinating the planning activity.
- The implementation of the Process Improvement Plan will need a cultural change within the organization.
Chapter 3: Resolution

This plan is created from input obtained from the questionnaires and interviews. Information gained from the questionnaires and interviews, combined with input from the organizations business needs are used to create the plan that will guide the SPI effort for the next few years.

The SPI plan contains the areas of improvement that will be addressed during the SPI activity, their relative priorities, and a description of the process that will be followed to accomplish the improvement.

These plans will provide guidance to the WGs that are formed to address a specific improvement activity from the SPI action plan. Usually there is a template of the format for this plan, partially completed by the SEPG and given to the WG at its start up. The WG will then complete the remainder of the template and submit the completed plan to the MSC for approval.

Establish infrastructure

The first step in establishing the Infrastructure is choosing its members based on the needs of each part of the infrastructure. This means first finding management members that will participate in the Management Steering Committee and the Steering Committee, and then choosing members for the SEPG and the Working Groups. Consequently, Working Groups need to be established for each process to be improved and a leader selected for each working group.

Provide Generic & Specific Training to WGs

After the infrastructure has been created, it needs to be strengthened by providing training and guidance to its members. The purpose of this activity is to train members of the working groups for a consistent approach to planning for the process improvement program and to develop skills in building a solid planning foundation upon which to sustain the process improvement program.

The training can be divided into generic and process specific.

The generic being areas that need to be addressed in training are:

- Process Definition, and Human Factors-Change Management.
- Process Model (CMMI-DEV and SCRUM).
- Planning & Monitoring.

The process specific training is related to the process area to be improved including activities, techniques, tools, documents and roles that need to be known in order for process improvement to go smoothly.

Action Plans: Plan the WG activities (for each WG)

Each of the working groups needs to have an action plan for their focus area. The focus plan is created with cooperation of the leader and the members of the WG with the leader making the final decisions. The action plan then needs to be approved by the Steering Committee.

The steps to be taken for each process after the action plan has been defined are:

1. Define the new process (defined process).
2. Put into practice (support from management and stakeholders).
3. Assess the new defined process.

**Define the new process**

The definition of a new process begins with understanding the existing process. Then the existing process should be refined to eliminate errors and reduce variations. Next the data from the questionnaire and interviews should be used to define needed improvements and with involving process stakeholders and understand their needs.

The defining of a new process deals with understanding a specific key process identified during the questionnaire and interviews phase and applying incremental refinements to the process.

For defining a new process, the steps to be taken there are:

- Collect data.
- Evaluate current and documented processes.
- Develop the initial process to be improved.
- Recommend tools and best practices to support the process.

The basis for making improvements to the existing processes or defining new ones will be the CMMI-DEV defined sub practices. Based on the coverage of each sub practice it can be seen if improvements are needed. Also, the sub practices can be used as a starting point for discussion about possible process improvement inside the working groups.

**Deploy to the organization**

When a new process for a certain process area has been designed it need to be deployed to the organization. In larger organization it would first be deployed inside one project at a time and after the improvement has been proven it would be implemented inside the whole company. In a SME the improve process should be applied on the next new project and if there are signs of improvement extend it to other current or to the future ones.

The first step in deploying the new improved process is to Institutionalize it. This means getting the go-ahead from the CEO and having a written document detailing the new process and what actions need to be taken to apply it.

Next step is providing training on the approved process to the people who will be connected to it, executing or monitoring it. During this training all the new practices introduced need to be explained in detail and the reason for their introduction explained so that there would be less resistance during the deployment of the new process.

While the improvement is in the beginning of deployment, it needs to be ensured that all support mechanisms are in place and functioning. This means there are mechanisms in place to prevent detrition of the new process due to lack of motivation or lack or resources to apply the new process correctly. Management needs to ensure sufficient resources for smooth deployment as well as continuous use of the new process. Sometimes it is also needed for the management to enforce the deployment of the new process, because of internal resistance.

In the Agile oriented companies, the environment and conditions change rapidly, and processes must evolve with them. This means the new process needs to be refined over
time using the information provided by staff and the metrics defined to assess the process.

**Assess the new defined process**

Assessing the new process that is being deployed to the company starts from the people deploying it. Direct feedback on how smooth the deployment is going is part of the assessment of the newly defined process. Next thing to be assessed about the new process is how maintainable it is, meaning how much effort is required to prevent it from deteriorating.

To see the improvement effects of the new process, some time need to pass. The time needed depends on the scale of the project its being deployed in as well as the rate at which the new process is being accepted.
4. Case study

In this chapter a case study applying the approach based on CMMI-DEV that was developed for process improvement using in small and medium sized software development enterprises that have embraced the agile development is presented. The first three phases of the process improvement lifecycle that were implemented will be presented. But firstly a general overview of case studies in software engineering will be depicted.

4.1 Case study methodology in software engineering

This subchapter is based on the (DE Perry, 2004) article and the (Robert K. Yin, 2002) book.

Case studies are a powerful and flexible empirical method. They are used for primarily for exploratory investigations, both prospectively and retrospectively, that attempt to understand and explain phenomenon or construct a theory. They are generally observational or descriptive in nature, though they can be relational as well. They can also be used in the validation of research results. Due to this dexterity, they have become popular in software engineering and are frequently used in papers to understand, to explain or to demonstrate the capabilities of a new technique, method, tool, process, technology or organizational structure.

Every type of empirical research has an implicit, if not explicit, research design. In the most elementary sense, the design is the logical sequence that connects the empirical data to the study’s initial research question and ultimately, to its conclusions (Robert K. Yin, 2002).

Case studies are good for answering detailed how and why questions and for gaining deep insights into chains of cause and effect. Also case studies are used for testing theories in complex settings where there is little control over the variables.

A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident (DE Perry, 2004).

This means that the use the case study method is used when the researches deliberately wants to cover contextual conditions-believing that they might be highly pertinent to researcher’s phenomenon of study.

The single-case study is an appropriate design under several circumstances. A single-case study is analogous to a single experiment, and many of the same conditions that justify a single experiment also justify a single-case study. One rationale for a single case is when it represents the critical case in testing a well-formulated theory (again, note the analogy to the critical experiment). The theory has specified a clear set of propositions as well as the circumstances within which the propositions are believed to be true. To confirm, challenge, or extend the theory, there may exist a single case, meeting all of the conditions for testing the theory. The single case can then be used to determine whether a theory’s propositions are correct or whether some alternative set of explanations might be more relevant.
4.2 Context

The case study was conducted in a company named Glownet, based in Madrid. The company organizes cashless payments on different kinds of events throughout the world. Their software part is the software they use in the devices to do cashless payment in different environments and circumstances combined with backend that enables customization and storing all the transactions done during an event. They also create reports based on the data gathered through events. The company is just now getting out of start-up phase and could benefit greatly in increasing their overall development process maturity.

The problems that needs to be addressed here is the limited quality and predictability of the development process through all the phases of development. Glownet uses agile development, more specifically a version of SCRUM tailored to its needs as a part of dynamic market environment and ever changing client requirements. The company has around 50 employees, which makes it a small-medium sized company.

The company organizes cashes payments on different kinds of events thought the world. The software they use is divided in 3 components:

- Software in the Android devices used to do cashless payment in different environments and circumstances.
- Backend in cloud using Ruby on Rails that enables customization and storing all the transactions done during an event.
- Creating reports based on the data gathered thought events.

The general structure of the development teams is shown in the Figure 4.1.

![Figure 4.1 Development teams structure](image)

Glownet uses agile development, more specifically a version of SCRUM tailored to its needs as a part of dynamic market environment with ever changing client requirements.
The Glownet engineering team are following best agile practices which ensure that the benefits of the methodology can be realized in the final product. The Glownet approach, while lacking detailed formal documentation, appears to remain flexible and effective through close communications and working relationships. Given the wide range of control tools used across the development process and the need for their ongoing configuration and management, it would be appropriate to assign a full time role to this work and so future revenue projections should incorporate this cost.

While Glownet is not an exclusively software development company, its aim for the future is to move from the hands on, on site approach to Software as a Service(SAAS) approach, by providing services to partners and event organizers without having to be physically present at the event. The current architecture of the Glownet payment system can be seen in this

![Glownet system architecture](image)

*Figure 4.2 Glownet system architecture*

### 4.2.1 SCRUM in Glownet

In Glownet development SCRUM is used with most of the defined processes and roles. Product backlog is created with cooperation from the Chief Technical Officer(CTO), heads od android, cloud and reports and input from head of operations during sprint planning. Estimates are also made on time required for each specific task.

The tool used for creating backlogs and monitoring current sprints is Jira. It provides a board similar to Kanban which enables to monitor and oversee the product backlog as well as the current state of the tasks in the sprint. The example of a SCRUM board can be seen in Figure 4.3
Chapter 4: Case study

Every day there is a Daily, when all the engineers meet and discuss the previous day. Each sprint lasts 3 weeks and is followed by a sprint review in which the new features or bug fixes are shown. After that in the sprint retrospective, tasks in that sprint are examined, the analysis is done if the estimation was close to the real time needed or not and adjustments are made for the tasks that were not completed.

4.3 Case study implementation strategy

The case study execution plan is based on the phases of the solution developed in the previous chapter. By following steps defined in the previous chapter and getting accustomed the real world environment, there were many problems that were encountered.

The phases followed in in this case study are:

- Obtaining the commitment.
- Executing the assessment of the current state.
- Solving inconsistencies in the assessment.
- Reporting assessment results.
- Creating new processes.
- Discussing the new processes.

4.4 Implementing the approach

4.4.1 Obtaining commitment

As highlighted in previous chapter, getting the process improvement (PI) commitment in a SME is difficult and strenuous endeavour. Even though the person from the
company who initiated the improvement endeavour was advocating for it there were many obstacles to be crossed.

The first obstacle was discussion about the need for any kind of improvement at that particular moment in time, since most smaller companies make an improvement effort only as a response to a hazard related to processes that had a major impact on the overall company business objectives. Since there was no major hazard as motivator the senior management needed to be convinced that applying the improvement would be worth the effort.

The next obstacle was finding funds in the company budget for the improvement endeavour. The costs in were minimal since the most of the improvement work would be done by the author who is motivated by implementing his research and would execute the improvement without additional monetary incentives. So, the author of this research is going to be the appraisal team leader, and there is another member for supporting the appraisal (i.e., the appraisal team is composed of two members).

One additional obstacle was agreeing on a schedule and improvement deadline, since there were always unavoidable things that needed to be executed right away or the company would lose clients or money.

After multiple discussions with the CTO, CFO, CEO and Human Resources manager of the company, an oral commitment to start the improvement endeavour was given without any written contract.

The main business objective that aligned with the improvement objective was improving processes in order to reduce wasted effort due to miscommunication and redundant actions, and in term improve the efficiency of the processes.

This objective can be generalised to most SMEs that have embraced any Agile approach. The lack of structure facilitates the creation of wasted effort due to lack of communication and clearly defined job scope for each person.

4.4.2 Appraisal plan

The agreed appraisal was for the CMMI level 2 maturity process areas with focus on management. That was the starting point for the appraisal plan.

In this particular case study, a lack of need for resources was presented due to the fact that the main executor of the appraisal had a different motivation for executing it, this meant drastically reduced cost and a highly motivated appraisal leader.

The scope of the appraisal was the whole software development division of the company, since the company is not purely a software development enterprise but has a software development division.

The output of the appraisal as per business objective was the appraisal of current state of the company processes and recommendation for improvement of certain processes or whole process areas in the context of increased efficiency.

The schedule for the appraisal plan had to be made flexible and “agile” because in Agile environment change of circumstances and priorities is constantly present. The plan schedule was made using the google tools for scheduling. An example is shown in Figure 4.4.
Chapter 4: Case study

The problems that were produced by having low cost estimation and flexible schedule will be discussed late in this chapter.

4.4.3 Training the appraisal team

The training of the appraisal team was not done in conventional presentation/workshop way. The appraisal team leader was also the main team member with an additional member only providing support for the appraisal. The supporting team member was first given all the supporting literature, presentations and papers related to the appraisal to be executed and then the steps to be taken were discussed in detail and all ambiguous actions cleared. The main role of the supporting assessment team member was to coordinate with the appraisal participants and update the schedule for any unexpected changes. All the main commitments of the appraisal were part of the team leader’s job.

4.4.4 Risk assessment

As with any larger endeavour within a company there are certain risks to successfully executing it. The improvement of processes in a SME carries with it some of the well known risks to the successful execution. The biggest risk to the execution of improvement is the loss of backing from the management of the company, during the execution. This is especially critical if the company has had previous bad experience with similar attempts. To remedy this risk regular meetings with the senior company management should be held to keep them updated and reinforce their determination for the process improvement endeavour.

Additional risk that needed to be considered were the lack of interest and motivation or even resistance from the appraisal participants and the loss of funding for the appraisal and improvement effort because of an external event.

Keeping the participants focused during the appraisal is the job of the management and that needs to be explicitly made clear. The appraisal leader that needs to keep the participants well informed about the progress and purpose of the appraisal. The loss of
funds for the appraisal is a type of risk that cannot really be prepared completely for and one of the solutions is freezing the appraisal until the business balance is good enough to resume the appraisal.

4.4.5 Executing the assessment

The assessment was executed by the assessment team leader with the help of the other team member for the logistics of the appraisal. The baseline for the appraisal was created using questionnaires created from the CMMI-DEV SCRUM mapping. Before answering the questions, four project leaders were part of the presentation that explained how to answer the questions and some of the terms that were created from the CMMI-SCRUM combination that are not part of the SCRUM defined terms and practises.

The process areas assessed were Project Planning, Configuration Management, Requirements Management, Project Monitoring and Control just like the tables generated for the mapping.

Four project leaders responded to the questions and with the predefined answers. Their answers were put into an Excel table with predetermined calculations formula defined in Chapter 3. Based on the answers from the appraisal, graphs were also generated for easier understanding of the results of the appraisal and general state of process areas. The SCRUM based questionnaire questions were understandable to the project leaders and only small specific sub practice related misunderstandings were presented but they were easily remedied. Figures 4.5- 4.8 depict coverage of specific practices in each appraised process area.

![Project Planning SPs Analysis](image)

*Figure 4.5 Project planning specific practice coverage*
Figure 4.6 Project Monitoring Control specific practice coverage

Figure 4.7 Configuration Management specific practice coverage
IN the end a summarized coverage of each of the process areas is shown in Figure 4.8.

There was one major problem that occurred during the execution of the initial appraisal. The problem was not respecting the schedule. The project leaders were often preoccupied with finishing their pending tasks and could not spend their time on the appraisal. Since the execution of appraisal did not require significant funds, it was not
Chapter 4: Case study

high enough on the priority list of the senior management to finish it. Additional challenge being an agile environment with ever-changing client requests and priorities. One of the PLs wrote:

“Sorry for the late sending of this... I was programming last week a bit with Kotlin, Spring, React.. and this questionnaire was a... a... something I wasn't motivated to do!”

Significant pressure had to be reapplied by the appraisal sponsor in order for the first phase of the appraisal to be successfully executed with significant delays.

4.4.5.1 Choosing preliminary process areas

Based on the questionnaires results, the worst covered process areas that were connected were chosen as preliminary process areas to be improved. The improvement areas chosen are:

- Project Planning.
- Project Monitoring Control.
- Requirements Management.

with a priority as in the order, starting with the Project Planning and continuing with the Project Monitoring Control.

4.4.5.2 Solving inconsistencies

As it can be seen in Figures 4.10 to 4.13, there were many sub practices with deviations higher than 1 in all of the process areas explored. Inconsistencies were solved by conducting interviews with the employees that work in the teams under project leaders that filled out the questionnaires. The interviews were done without explicit interview forms, relying on the list of inconsistencies and the communicative skills of the appraisal team leader and the working professionals. This was done easily because of the relaxed atmosphere in the Agile environment with people being open to talking and commenting the issues.
Chapter 4: Case study

The coverage of the preliminary process areas after solving the inconsistencies can be seen in Figures 4.14 to 4.16.

**Figure 4.14 Project planning specific practice coverage after resolving inconsistencies**

**Figure 4.15 Project Management Control specific practice coverage after resolving inconsistencies**
4.4.6 Reporting assessment results

The assessment results were presented and then discussed with the CTO and the appraisal sponsor. The main points of discussion were why some of the sub-practices had such a low coverage and if they were needed at all in the Agile environment.

The next discussion point were the preliminary process areas and why should they be the ones to be improved first. After seeing the data and discussing the implications, the CTO agreed to improve all the preliminary chosen process areas. The first process area to be improved being the Project planning.

4.4.7 Process improvement plan and infrastructure

4.4.7.1 Infrastructure

The general structure of the people working on the improvement endeavour was explained in the previous chapter. During the execution of this study case the infrastructure created followed the instructions given in the previous chapter. Adjustments needed to be made to suit the needs of the particular company. Both the management steering committee and the steering committee only had one member each. The CEO of the company represented the management steering committee and the CTO of the steering committee. The SEPG was made up of 2 product leaders that were also the part of the questionnaires.
4.4.7.2 Creating new processes

Project Planning

The first process area to be improved is the Project Planning based on the resolved results of the questionnaires and the discussion with the CTO on how to proceed, a working group with 3 members was formed.

The first step was reviewing each sub practice coverage after the inconsistencies were resolved. Figure 4.17 shows how sub practice coverage looks after resolving inconsistencies.

![Project Planning Answers Analysis](image)

**Figure 4.17 Project Planning specific sub practice coverage after solving inconsistencies**

The starting point of the new process design are the specific practices from CMMI-DEV that are more or less covered by the SCRUM practices in the company.

Improvements to be implemented are listed by specific practice:

**SP 1.1 Estimating the Scope of the project:**
- Reuse estimates from previous projects in order to improve the accuracy by introducing a knowledge data base.
- When in the initial stages of planning the new project internal documentation about the initial structure of the project should be created.
- Additional separate discussions should be held about externally acquired and reusable work products product.

**SP 1.2 Establish estimates of work product and task attributes:**
- During the pre-game introduce planning poker for estimating the cost needed for the individual work products.
- Establish a knowledge base to be used in current and future project estimations.

**SP 1.3 Define Project Lifecycle Phases:**
The lifecycle phases are defined by the SCRUM framework and are essential for successfully executing SCRUM based development. There is no immediate need for improving this specific practice.

**SP1.4 Estimate Effort and Cost:**
- Introduce knowledge data base.
Chapter 4: Case study

SP2.1 Establish the Budget and Schedule:
- Create a table of project work products estimations for the time and costs for their implementation and for the constraints.
- Create a dependency matrix between the work products.
- During each sprint review discuss if corrective action needs to be taken in relation to project estimates.

SP2.2 Identify Project Risks:
- Create project risks checklist.
- Use the filled checklist to present risks to stakeholders and monitor them during sprint reviews.

SP2.3 Plan Data Management:
- Create company level guidelines for handling date management and assign one person to oversee its implementation.

SP2.4 Plan the Project’s Resources:
- Start using specialised resource management software.

SP2.5 Plan Needed Knowledge and Skills:
- Introduce a new skill estimation discussion during each pre game planning.
- Document the required skills estimations.
- Update the estimation during each sprint review if needed.

SP 2.6 Plan Stakeholder Involvement:
- Company level general guidelines for stakeholder involvement is defined.
- Current stakeholder involvement is reviewed.

SP 2.7 Establish the Project Plan:
- Create a company level template for all future project plans

3.1 Review plans that affect the project:
- Before obtaining plan commitment, it is necessary to have a meeting with senior management about the plans affecting the project.

3.2 Reconcile work and resource levels:
- Before obtaining plan commitment, it is necessary to check if the estimated and available resources are aligned; if not, adjust the time estimations or task for additional resources.

3.3 Obtain plan commitment:
- One of the people from the senior management should sign a document approving the plan.

4.4.8 Discussing the new processes

The new process improvements and the new company level guide lines do not extensive effort to create and follow. New things that were introduced are:
- Knowledge data base.
Chapter 4: Case study

- Table of project work products estimations for the time and costs.
- Dependency matrix.
- Risk checklist.
- Guidelines.
- Resource management software.

Introducing a **knowledge database** for estimating future projects is the biggest change that was suggested. Implementing the knowledge database would require at least one full time worker to work on its creation and expansion. Since the environment is already SCRUM based the creation of the knowledge database also needs to be incremental, starting with the last finished product and continuing backwards.

**Estimation table** would just be a more structured way of displaying the estimated time and effort, and would make it easier to discuss about the estimates and reconcile the estimates with available resources.

Using **dependency matrices** for determining dependencies between the work products is a common practice during architecture phase of the project. Introducing the use of dependency matrices requires very little additional effort but gives a good starting point for considering dependencies.

**Risk checklist** is a good way to consider all the possible risks to the project without omitting any. It can be created on a company level and even adapter to the needs of the company from some existing risk checklist. It can later be used to communicate possible risks to the relevant stakeholders and to keep as a reminder of the possible risks during project execution.

Creating company level **guidelines** is one of the steps needed for transitioning from a start-up to a more mature company. They help with establishing new processes and overall development routine and also make it simpler for new employees to adapt to the work dynamic in the company.

**Resource management software** tools help companies effectively plan resources. Software has included everything from spreadsheets and applications to commercial-off-the-shelf products. Spreadsheets require manual data updates, so teams who start on spreadsheets tend to look for more powerful software later on. More powerful software may include collaboration, information-sharing, reporting, and utilization tools that become bulky and time-consuming in spreadsheets. For small companies, resource planning can be accomplished without software through regular team meetings and by using a whiteboard. Once companies reach a certain size, more formal tools are required to avoid resource shortages and maintain utilization.

Figure 4.18 depicts how the SCRUM process would look like if the improvements suggested are implemented. The blue part of the graph represents the new things added and the yellow part represents the existing parts of the SCRUM process. Some of the blue added parts are not completely new concepts and are only highlighted to increase focus on them.
Figure 4.18 Improved SCRUM process with project planning
Chapter 4: Case study

4.5 Conclusions

During the execution of the case study few main points needed to be addressed:

- Getting the commitment from the senior management.
- Problems with the execution of the appraisal.
- How the created questionnaires were received.
- Suggested improvements.

Even though the general guidelines for getting the commitment from senior management were defined in an SME with Agile environment obtaining the oral commitment, it was not sufficient for executing the appraisal and improvement smoothly. The schedule for starting the appraisal was always being delayed, because there would always be other priorities.

During the execution of the appraisal there were multiple delays and shifting of schedule because the participants kept postponing their involvement even they were all part of the presentation about the questionnaires. This created an additional problem of different Project Leaders (PLs) answering the questionnaires at different times and could be influenced by PLs that filled it out earlier.

The questionnaires from the mappings were well received, even people without a strong technical background could understand them based on their knowledge of the SCRUM framework. Only minor things that were directly related to CMMI-DEV had to be explained.

As far as creating new processes and suggesting improvements to the existing one are concerned first thing that needs to be mentioned is that CMMI-DEV specific practices serve as excellent guideline for brainstorming and suggesting improvements. With CMMI-DEV sub practices combined with the questionnaire responses used as an improvement starting point, the task of improving a process was easily divided into subtasks related directly to the sub practices. This made generating improvement ideas as well as upgrading the existing process progress very smoothly.

On a final note due to problems of constant delays already mentioned combined with the time limited nature of this thesis most of the improvements could not be seen through to the end. The author will continue with the improvements in the before mentioned company until all the process areas in CMMI-DEV maturity level 2 have been improved but the results of those effort shall be written in an extension of this thesis.
5. Conclusion

In this chapter a summation of all the work done in the thesis is presented as well as ideas for future research. First, an overall summary of the previous chapters is given, next research questions are revisited. Consequently, the flow of and the results of the case study are summarized. In the end limitations of the done work are explored and recommendations for further research work are given.

5.1 Summary

In the introduction of the thesis the context in which the thesis was being developed was described as well as the problems that it strives to solve or remedy. After that the importance of resolution is described to explain the reasons why the particular problem has been chosen as significant for being solved. At the end, the goals of the thesis are described, decomposed into sub goals and pragmatically instantiated through objectives. One of the main parts of the thesis and the one which took most effort to make is the state of the art. State of the art was directed at describing the current state of the art related to the goals and objectives of the thesis. It started with an overall overview of approaches whose combination was to be explored within the systematic literature review. The SLR was defined and executed with the goal of exploring the existing published approaches to solving the problems defined in the thesis and the shortcoming of the existing approaches as well as generating groundwork for creating a new or improved approach. While the starting point of the SLR was all Agile based practices, later SCRUM became the representing practice instead of general Agile software development paradigm.

Using the results from the SLR, a definition of an upgraded process improvement lifecycle was made, with all the steps to be taken to determine the current state of the processes and improve it. With the new improvement approach defined, a case study was conducted. The case study was conducted in a Madrid based SME, with all the plans, personnel and infrastructure involved. In the end, the results of the application of the improvement approach, defined previously, were presented.

5.2 Research question revisited

The initial research question and the starting point of the thesis was: if CMMI-DEV could be applied to small or medium sized enterprises(SMEs) that have embraced any form of Agile development. The reasons for creating this research questions were that the author had experience with the SMEs with Agile paradigm, and knowledge about CMMI-DEV. With the initial question decomposed into two in the Systematic Literature Review (SLR), there were two consequential questions during the SLR. First question that needed to be addressed is: if CMMI-DEV can be used at all in an Agile environment of an SME. The answer to that question was found to be overwhelmingly positive, with many different approaches showing positive results. During the SLR it was concluded that researching all types of combinations between any Agile practice and CMMI-DEV would be too diverse for the scope and limitations of a master thesis that aimed to also have a practical part. Focus of the research was shifted from all Agile practices to SCRUM. Other Agile practices, or Agile paradigm in general served as a guideline for shortcomings of Agile practices relative to the CMMI-DEV model. With the focus shifted
exclusively to SCRUM the second question was addressed. There were multiple cases of SMEs with SCRUM using CMMI for process improvement. From there, multiple points were made about using CMMI-DEV on SCRUM based SMEs. Those points were later used to devise an improvement approach for SMEs using SCRUM in particular. Based on the information gained from the SLR and the previous knowledge about the process improvement lifecycles, a new improvement approach was designed and applied in Madrid based SME to gain empirical results for the designed improvement approach.

5.3 Contributions and outcomes of the research

The starting point of the thesis was a question of using CMMI on SMEs who have embraced Agile paradigm to improve processes. First thing that was done was making an overview of Agile and CMMI based on the information from their creator and opinions of some members of academia. Next, a brief comparison was made and used as a starting point for the SLR conducted by the author. Conducting a SLR within limited timeframe and with limited student access was quite demanding work. Since there were direct, practical question as the basis for the systematic literature review, the result of the SLR are the answers to that question. An overview of the combination of CMMI-DEV with Agile practices was created and the direct experiences of using CMMI-DEV in SCRUM based environments were listed.

Using the overview and experiences as a basis, an approach was created for applying CMMI-DEV onto SME with SCRUM. The first step of that approach was creating a mapping between the CMMI-DEV level 2 specific practices and the practices of SCRUM. By creating this mapping, a direct connection between CMMI-DEV and SCRUM was established.

The next step was tailoring SCAMPI to the needs of a SME with SCRUM. This was done by first creating questionnaires that are based on the created mapping. The questionnaires were created using SCAMPI, CMMI-DEV process areas sub practices and knowledge the author has about SCRUM and CMMI. The created mapping was used in the assessment execution. The additional step in tailoring SCAMPI that needed to be taken was adjusting the requirements defined by SCAMPI to suit the needs of a SME. Additional information that was used for tailoring SCAMPI was gained from researching about the AFIM and IDEAL process improvement lifecycles.

With the SCAMPI tailored the next contribution was made by tailoring AFIM phases for process improvement planning and implementation using IDEAL and limitations that come from it being applied to a SME. The tailored phases of AFIM were combined with Tailored SCAMPI to create a new process improvement approach for SMEs using SCRUM.

A Case study about improving processes in a SME with SCRUM practices, using the created improved approach, was conducted. Most of the tasks that were the part of the Case study were executed by the author alone. This meant taking on many different roles in the improvement approach and a lot of invested effort. Due to time limitations and problems in case study execution, that were mentioned, the results of the case study are not conclusive and require additional research time and effort.
5.4 Limitations

The biggest limitation that was present is the limited time for writing the thesis. Most other limitations are derived from the time limit. The first limitation that could be derived is the time limit for the SLR, executing SLR requires a lot of effort for the systematic approach. The limited time forced the author to focus on a more specialised papers related to the research question and have a stricter acceptance criterion. The next limitation was that the case study was conducted in only one company due to limited time and available resources.

One additional limitation that was not time related was that all the research was done by a student. This implies limited access to some of the research (especially in Springer) as well as not being taken serious enough by the company that the case study was executed in. This caused the case study to extend much over the initial schedule and reduced the results available.

5.5 Recommendations for future work

The research done in this thesis is just a stepping stone for extended research into creating a hybrid process improvement approach between CMMI and Agile. Due to limitations mentioned in the previous subchapter a lot of work still remains to be done. The author will continue applying the created process improvement approach in the company that was the part of the case study. The created approach is more of a prototype than a wholesome, well defined improvement approach. The author will continue to improve it outside of scope of the thesis, first by designing and implementing improvements for all the mapped process areas and then conduct further research into mapping more process areas to SCRUM practices.

The first thing that can be done in some future research is extending the created CMMI –SCRUM mapping to all of the CMMI process areas at the level 2 maturity level.

Generalising the approach for improving all Agile based practised could also be one of the possible directions for future research. In this research the focus would be on the short coming that all the Agile based practices share, regardless of the specific practice.

Mapping CMMI-DEV to another Agile based practice, Extreme programming for example could also be part of some future research. This would be a start of aggregated approach that could encompass all the most popular Agile practices, with a general part that would address general Agile shortcoming and a part that would contain specified approaches for each Agile practice.
Chapter 5: Conclusion
Chapter 6: REFERENCES

6. REFERENCES


Chapter 6: REFERENCES


6.1 SLR PAPERS


Chapter 6: REFERENCES


Chapter 6: REFERENCES


Łukasiewicz, K., & Miler, J. (2012). Improving agility and discipline of software development with the SCRUM and CMMI. IET software, 6(5), 416-422.


7. Appendix

7.1 Configuration Management (CM)

7.1.1 Questions tailored for SCAMPI CM area

1.1 Identify Configuration Items
   1. All the Configuration Items (CIs) [1] are defined in the backlog.
   2. CIs have unique identifiers in the CM tools.
   3. The important characteristics of each configuration item are specified.
   4. Tools are used to follow the state and requirements of each CI.
   5. Each CI has a SCRUM member responsible for it.
   6. There are documents explaining connections between the CIs.

1.2 Establish a Configuration Management System
   1. The tools used for CMS (configuration management system) gives multiple levels of control (level of authorisation needed to make changes).
   2. Access control is provided to CMS.
   3. Configuration items can be stored and retrieved in a configuration management system.
   4. Configuration items can be transferred between control levels in the configuration management system.
   5. Archived versions of configuration items can be stored and recovered.
   6. Configuration management records can be stored, updated, and retrieved.
   7. Configuration management reports are created from the configuration management system.
   8. The contents of the configuration management system are preserved.
   9. Configuration management structure is revised as necessary.

1.3 Create or Release Baselines
   1. Before creating or releasing baselines of configuration items authorization is obtained [7].
   2. The tools used for CM enable creating or releasing baselines only from configuration items in the configuration management system.
   3. The set of configuration items that are contained in a baseline are documented using the CM tools.
   4. The CM tools make the current set of baselines readily available.

2.1 Track Change Requests
   1. Change requests to the CI are made and tracked using external tools.
   2. The impact of changes and fixes proposed in change requests are analysed.
   3. Change requests are Categorized and prioritized.
   4. Reviewed change requests are addressed in the next baseline with relevant stakeholders [3] and get their agreement.
   5. Status of change requests is tracked to closure.
2.2 Control Configuration Items

1. Changes to configuration items are controlled during the Sprint Reviews.
2. SCRUM team members obtain appropriate authorization before changed configuration items are entered into the configuration management system.
3. Check in and check out configuration items in the configuration management system for incorporation of changes is done in a manner that maintains the correctness and integrity of configuration items. [2]
4. During sprint reviews it is ensured that changes have not caused unintended effects on the baselines (e.g., ensure that changes have not compromised the safety or security of the system).
5. During development changes to configuration items and reasons for changes as appropriate are recorded.

3.1 Establish Configuration Management Records

1. Configuration management actions are recorded in sufficient detail, so the content and status of each configuration item is known, and previous versions can be recovered.
2. Product owner ensures that relevant stakeholders have access to and knowledge of the configuration status of configuration items.
3. The latest versions of CI making up the baseline are specified in CM tools.
4. The Version of configuration items that constitute a baseline are clearly defined.
5. Differences between successive baselines can be seen using the CM tools
6. The status and history (i.e., changes, other actions) of each configuration item is revised during meetings as necessary.

3.2 Perform Configuration Audits

1. Integrity of baselines is assessed by the SCRUM master and product owner.
2. Before the start of each sprint is confirmed that configuration management records correctly identify configuration items.
3. During the sprint review the structure and integrity of items in the configuration management system is reviewed.
4. During the sprint review the completeness, correctness, and consistency of items in the configuration management system is confirmed.
5. Confirm compliance with applicable configuration management standards and procedures.
6. Action items from the audit are tracked to closure.

[1] e.g. project plan, specifications, design documents, source code, test plans and test data, executables, make les, tools

[2] Examples of check-in and check-out steps include the following:
- Confirming that the revisions are authorized
- Updating the configuration items
- Archiving the replaced baseline and retrieving the new baseline
- Commenting on the changes made to the item
- Tying changes to related work products such as requirements, user stories, and tests

[3] stakeholders - Managers, staff, customers, end users, suppliers, and other relevant stakeholders are included in milestone reviews as appropriate.

[7] Baseline for the configuration items are the previous versions of items stored using the tools.

### 7.1.2 Answers for SCAMPI CM area

<table>
<thead>
<tr>
<th>Configuration Management</th>
<th># Project Leaders</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>#F</th>
<th>#L</th>
<th>#P</th>
<th>#N</th>
<th>Cq</th>
<th>Medq</th>
<th>Devq</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG 1</td>
<td>Establish Baselines</td>
<td>21</td>
<td>29</td>
<td>13</td>
<td>1</td>
<td>28%</td>
<td>38%</td>
<td>17%</td>
<td>1%</td>
<td>58%</td>
<td>66%</td>
<td>2.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SP 1.1</td>
<td></td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>33%</td>
<td>38%</td>
<td>13%</td>
<td>0%</td>
<td>62%</td>
<td>66%</td>
<td>2.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Subpractice 1.1.1</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.1.2</td>
<td>F</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>83%</td>
<td>2.50</td>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.1.3</td>
<td>F</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.1.4</td>
<td>F</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>83%</td>
<td>2.50</td>
<td>0.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.1.5</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>75%</td>
<td>2.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.1.6</td>
<td>N</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>41%</td>
<td>1.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 1.2</td>
<td></td>
<td>9</td>
<td>13</td>
<td>6</td>
<td>0</td>
<td>54%</td>
<td>4.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.1</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.2</td>
<td>F</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>92%</td>
<td>2.75</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.3</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>83%</td>
<td>2.50</td>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.4</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>58%</td>
<td>1.75</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.5</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>58%</td>
<td>1.75</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.6</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.7</td>
<td>N</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.8</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>75%</td>
<td>2.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.2.9</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>66%</td>
<td>2.00</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 1.3</td>
<td></td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>62%</td>
<td>4.44</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.3.1</td>
<td>N</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>41%</td>
<td>1.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.3.2</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>58%</td>
<td>1.75</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.3.3</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>75%</td>
<td>2.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subpractice 1.3.4</td>
<td>P</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>75%</td>
<td>2.25</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 7: Appendix

#### SG 2: Track and Control Changes

<table>
<thead>
<tr>
<th>Subpractice</th>
<th>F</th>
<th>P</th>
<th>L</th>
<th>L</th>
<th>%</th>
<th>Ccm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 2.1.1</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 2.1.2</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 2.1.3</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 2.1.4</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>75%</td>
</tr>
<tr>
<td>Subpractice 2.1.5</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>25%</td>
<td>75%</td>
</tr>
</tbody>
</table>

#### SG 3: Establish Integrity

<table>
<thead>
<tr>
<th>Subpractice</th>
<th>L</th>
<th>N</th>
<th>L</th>
<th>L</th>
<th>%</th>
<th>Ccm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 2.2.1</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>0%</td>
<td>75%</td>
</tr>
<tr>
<td>Subpractice 2.2.2</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 2.2.3</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>75%</td>
</tr>
<tr>
<td>Subpractice 2.2.4</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.2.5</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1%</td>
<td>1%</td>
</tr>
</tbody>
</table>

#### SP 3.1

<table>
<thead>
<tr>
<th>Subpractice</th>
<th>L</th>
<th>P</th>
<th>L</th>
<th>L</th>
<th>%</th>
<th>Ccm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 3.1.1</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.1.2</td>
<td>P</td>
<td>L</td>
<td>F</td>
<td>F</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.1.3</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.1.4</td>
<td>P</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.1.5</td>
<td>P</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>3%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 3.1.6</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>1%</td>
<td>2%</td>
</tr>
</tbody>
</table>

#### SP 3.2

<table>
<thead>
<tr>
<th>Subpractice</th>
<th>L</th>
<th>P</th>
<th>L</th>
<th>L</th>
<th>%</th>
<th>Ccm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 3.2.1</td>
<td>N</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.2.2</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>1%</td>
<td>3%</td>
</tr>
<tr>
<td>Subpractice 3.2.3</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>F</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.2.4</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Subpractice 3.2.5</td>
<td>L</td>
<td>N</td>
<td>L</td>
<td>L</td>
<td>0%</td>
<td>3%</td>
</tr>
<tr>
<td>Subpractice 3.2.6</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>1%</td>
<td>3%</td>
</tr>
</tbody>
</table>

#### Totals

<table>
<thead>
<tr>
<th>#F</th>
<th>#L</th>
<th>#P</th>
<th>#N</th>
<th>Ccm</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>79</td>
<td>37</td>
<td>7</td>
<td>64%</td>
</tr>
</tbody>
</table>

7.2 Project Management and Control (PMC)

7.2.1 Questions tailored for SCAMPI PMC area

1.1 Monitor project planning parameters

1. Progress is monitored against the schedule in the SCRUM meetings.
2. The project’s costs and expended effort are monitored in the SCRUM meetings.
3. The attributes of work products and tasks are monitored in the SCRUM meetings.
4. Resources provided and used are monitored in the SCRUM meetings.
5. **Monitor the knowledge and skills of project staff — inapplicable.**
6. Significant deviations in project planning parameters are documented during the SCRUM meetings.

1.2 Monitor commitments
1. **Commitments (both external and internal) are regularly reviewed during each sprint.**
2. Commitments that have not been satisfied or are at significant risk of not being satisfied are identified during each sprint.
3. The results of commitment reviews are documented.

1.3 Monitor project risks
1. The documentation of risks in the context of the project’s current status and circumstances is periodically reviewed during SCRUM meetings.
2. The documentation of risks is revised as additional information becomes available during review meetings.
3. The product owner communicates the risk status to relevant stakeholders.

1.4 Monitor data management
1. Periodically review data management activities against their description in the project plan.
2. Identify and document significant issues and their impacts.
3. Document results of data management activity reviews.

1.5 Monitor stakeholder involvement
1. Product owner periodically reviews the status of stakeholder involvement.
2. During Sprint planning and review significant issues and their impacts are identified and documented.
3. Product owner documents the results of stakeholder involvement status reviews.

1.6 Conduct progress reviews
1. Product owner regularly communicates status on assigned activities and work products to relevant stakeholders.
2. **The results of collecting and analysing measures of the sprint are reviewed during the sprint review meetings for controlling the project.**
3. Significant issues and deviations from the plan are identified and documented during sprint reviews meetings.
4. Change requests and problems identified in work products and processes are documented during sprint planning and sprint review meetings.
5. The results of reviews are documented.
6. Change requests and problem reports are tracked to closure.

1.7 Conduct milestone reviews
1. Milestone reviews with relevant stakeholders at meaningful points in the project’s schedule, such as the completion of selected phases during are conducted by the Product owner.
2. Commitments, the plan, status, and risks of the project are reviewed during Sprint reviews.
3. Significant issues and their impacts are identified and documented during Sprint reviews.
4. Results of the review, action items, and decisions are documented.
5. Action items are tracked until closure to closure.

2.1 Analyze issues
1. During Daily and review meetings issues are gathered for analysis.
2. Issues are analysed during review meetings to determine the need for corrective action.

2.2 Take corrective action
1. The appropriate actions needed to address identified issues are determined and documented during sprint planning and sprint review meetings.
2. Review and get agreement with Product owner on the actions to be taken.
3. Product owner negotiates changes to internal and external commitments.

2.3 Manage corrective action
1. Corrective actions are monitored for their completion by the SCRUM master.
2. The results of corrective actions are analysed during sprint reviews to determine the effectiveness of the corrective actions.
3. Appropriate actions to correct deviations from planned results are determined during sprint reviews and sprint planning.
### 7.2.2 Answers tailored for SCAMPI PMC area

<table>
<thead>
<tr>
<th>Project Monitoring and Control</th>
<th># Project Leaders</th>
<th>#F</th>
<th>#L</th>
<th>#P</th>
<th>#N</th>
<th>Cq</th>
<th>Medq</th>
<th>Devq</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SG 1</strong></td>
<td>Monitor the Project against the Plan</td>
<td>20</td>
<td>50</td>
<td>39</td>
<td>6</td>
<td>17%</td>
<td>43%</td>
<td>34%</td>
</tr>
<tr>
<td><strong>SP 1.1</strong></td>
<td></td>
<td>2</td>
<td>9</td>
<td>8</td>
<td>4</td>
<td>9%</td>
<td>39%</td>
<td>35%</td>
</tr>
<tr>
<td>Subpractice 1.1.1</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subpractice 1.1.2</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>N</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Subpractice 1.1.3</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.1.4</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.1.5</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.1.6</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>SP 1.2</strong></td>
<td></td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>8%</td>
<td>50%</td>
<td>33%</td>
</tr>
<tr>
<td>Subpractice 1.2.1</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.2.2</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.2.3</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>SP 1.3</strong></td>
<td></td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 1.3.1</td>
<td>P</td>
<td>L</td>
<td>P</td>
<td>P</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.3.2</td>
<td>P</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.3.3</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>P</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>SP 1.4</strong></td>
<td></td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>17%</td>
<td>42%</td>
<td>42%</td>
</tr>
<tr>
<td>Subpractice 1.4.1</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.4.2</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.4.3</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>P</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td><strong>SP 1.5</strong></td>
<td></td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>8%</td>
<td>50%</td>
<td>42%</td>
</tr>
<tr>
<td>Subpractice 1.5.1</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.5.2</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Subpractice 1.5.3</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
## Chapter 7: Appendix

<table>
<thead>
<tr>
<th>SP 1.6</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 1.6.1</td>
<td>F</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>46%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>29%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4%</td>
</tr>
<tr>
<td></td>
<td>83%</td>
<td>50%</td>
<td>50%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.6.2</td>
<td>F</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.6.3</td>
<td>L</td>
<td>N</td>
<td>L</td>
<td>P</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.6.4</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.6.5</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>P</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 1.7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 1.7.1</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.7.2</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.7.3</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.7.4</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.7.5</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SG 2 Manage Corrective Action to Closure</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>71%</td>
</tr>
<tr>
<td></td>
<td>31%</td>
<td>53%</td>
<td>16%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.1.1</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.1.2</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 2.2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 2.2.1</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.2.2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.2.3</td>
<td>L</td>
<td>F</td>
<td>F</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP 2.3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpractice 2.3.1</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.3.2</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 2.3.3</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Totals</th>
<th>#F</th>
<th>#L</th>
<th>#P</th>
<th>#N</th>
<th>Cpmc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>67</td>
<td>44</td>
<td>6</td>
<td>60%</td>
</tr>
</tbody>
</table>
7.3 Project Planning (PP)

7.3.1 Questions tailored for SCAMPI PP area

1.1 Estimate the scope of the project

1. At the beginning of the project all SCRUM roles are defined.
2. Product owner and SCRUM master define the work packages in sufficient detail so that estimates of project tasks, responsibilities, and schedule can be specified.
3. Products and product components to be externally acquired are identified during the pre-game.
4. Work products to be reused are identified during the pre-game.

1.2 Establish estimates of work product and task attributes

1. Technical approach [4] for the project is determined in the Pre-game.
2. Use appropriate methods to determine the attributes of the work products and tasks to be used to estimate resource requirements. [5]
3. Estimates of the attributes of work products and tasks are made during the SCRUM meetings [6].

1.3 Define project lifecycle

- Comes from SCRUM

1.4 Determine estimates of effort and cost

1. During sprint reviews and sprint retrospective historical data is used to improve estimates of labour hours and costs.
2. Infrastructure resource needs are considered in the development environment, the test environment, the production environment, the operational environment, or any appropriate combination of these environments when estimating effort and cost.
3. Before each sprint estimate of effort and cost is given based on the previous sprints

2.1 Establish the budget and schedule

1. Major milestones are identified during pre-game.
2. Schedule assumptions are identified during pre-game and before each sprint.
3. Constraints to development are identified during pre-game and before each sprint.
4. Task dependencies are identified during pre-game and before each sprint
5. The budget and schedule are established during pre-game and maintained and update during sprint planning.
6. Corrective action criteria.

2.2 Identify project risks
1. Potential risks are identified during pre-game, sprint planning and daily meetings.
2. Potential risks are documented.
3. Product owner reviews and obtains agreement with relevant stakeholders on the completeness and correctness of risks.
4. Risks are revised during meetings as appropriate.

2.3 Plan for data management
1. Requirements and procedures to ensure privacy and the security of data are established.
2. A mechanism to archive data and to access archived data is established.
3. The project data which needs to be identified, collected, and distributed is determined before each sprint.
4. The product owner determines the requirements for providing access to and distribution of data to relevant stakeholders.
5. When starting a project it is decided which project data and plans require version control or other levels of configuration control and establish mechanisms to ensure project data are controlled.

2.4 Plan for project resources
1. SCRUM process requirements are respected.
2. In the Pre-game and during sprint planning communication requirements are determined.
3. In the Pre-game and during sprint planning staffing requirements are determined or updated.
4. During spring planning and sprint review facility, equipment, and component requirements are determined.
5. During spring planning and sprint review other continuing resource requirements are determined.

2.5 Plan for needed knowledge and skills
1. In the Pre-game while creating product backlog the knowledge and skills needed to perform the project are identified.
2. During the Pre-game the knowledge and skills available is assessed.
3. During the Pre-game and sprint planning mechanisms for providing needed knowledge and skills.
4. Selected mechanisms are incorporated into the project plan.

2.6 Plan stakeholder involvement
- Should be defined in SCRUM
2.7 Establish the project plan
- Should be defined in SCRUM
3.1 Review plans that affect the project
1. All plans that affect the project to understand project commitments.
3.2 Reconcile work and resource levels
   1. Adjust the project plan to reconcile available and estimated.

3.3 Obtain plan commitment
   1. Product owner identifies needed support and negotiates commitments with relevant stakeholders.
   2. All organizational commitments, both full and provisional are documented, ensuring the appropriate level of signatories.
   3. Product owner and SCRUM master review internal commitments with senior management as appropriate.
   4. Product owner and SCRUM master review external commitments with senior management as appropriate.
   5. During sprint review meetings commitments regarding interfaces between project elements and other projects and organizational units are reviewed so that these commitments can be monitored.

[4] The technical approach defines a top-level strategy for development of the product. It includes decisions on architectural features, such as distributed or client/server; state-of-the-art or established technologies to be applied, such as robotics, composite materials, or artificial intelligence; and the functionality and quality attributes expected in the final products, such as safety, security, and ergonomics.

[5] Methods for determining size and complexity should be based on validated models or historical data.

The methods for determining attributes evolve as the understanding of the relationship of product characteristics to attributes increases.

[6] Examples of work products for which size estimates are made include the following:
   - Deliverable and non-deliverable work products
   - Documents and files
   - Operational and support hardware, firmware, and software
### 7.3.2 Answers tailored for SCAMPI PP area

<table>
<thead>
<tr>
<th>Project Planning</th>
<th># Project Leaders</th>
<th>#F</th>
<th>#L</th>
<th>#P</th>
<th>#N</th>
<th>Cq</th>
<th>Medq</th>
<th>Devq</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG 1</td>
<td>Establish Estimates</td>
<td>7</td>
<td>18</td>
<td>15</td>
<td>3</td>
<td>16.3%</td>
<td>41.9%</td>
<td>34.9%</td>
</tr>
<tr>
<td>SP 1.1</td>
<td></td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>25%</td>
<td>38%</td>
<td>38%</td>
</tr>
<tr>
<td>Subpractice 1.1.1</td>
<td>F</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Subpractice 1.1.2</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Subpractice 1.1.3</td>
<td>F</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Subpractice 1.1.4</td>
<td>F</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>SP 1.2</td>
<td></td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>8%</td>
<td>58%</td>
<td>25%</td>
</tr>
<tr>
<td>Subpractice 1.2.1</td>
<td>P</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0%</td>
<td>75%</td>
<td>25%</td>
</tr>
<tr>
<td>Subpractice 1.2.2</td>
<td>P</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0%</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 1.2.3</td>
<td>L</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>25%</td>
<td>75%</td>
<td>0%</td>
</tr>
<tr>
<td>SP 1.3</td>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>67%</td>
<td>33%</td>
<td>0%</td>
</tr>
<tr>
<td>Subpractice 1.3.1</td>
<td>L</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>67%</td>
<td>33%</td>
<td>0%</td>
</tr>
<tr>
<td>SP 1.4</td>
<td></td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>0%</td>
<td>33%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 1.4.1</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Subpractice 1.4.2</td>
<td>N</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0%</td>
<td>0%</td>
<td>75%</td>
</tr>
<tr>
<td>Subpractice 1.4.3</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0%</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>SG 2</td>
<td>Develop a Project Plan</td>
<td>19</td>
<td>54</td>
<td>24</td>
<td>4</td>
<td>62%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>19%</td>
<td>53%</td>
<td>24%</td>
<td>4%</td>
<td>66%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP 2.1</td>
<td>Subpractice 2.1.1</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.1.2</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.1.3</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.1.4</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.1.5</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.1.6</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>SP 2.2</td>
<td>Subpractice 2.2.1</td>
<td>L</td>
<td>L</td>
<td>F</td>
<td>L</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.2.2</td>
<td>L</td>
<td>N</td>
<td>L</td>
<td>P</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.2.3</td>
<td>F</td>
<td>L</td>
<td>L</td>
<td>P</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.2.4</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SP 2.3</td>
<td>Subpractice 2.3.1</td>
<td>F</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.3.2</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.3.3</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.3.4</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.3.5</td>
<td>P</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SP 2.4</td>
<td>Subpractice 2.4.1</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.4.2</td>
<td>F</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.4.3</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.4.4</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.4.5</td>
<td>L</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>SP 2.5</td>
<td>Subpractice 2.5.1</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.5.2</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>L</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.5.3</td>
<td>L</td>
<td>P</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Subpractice 2.5.4</td>
<td>L</td>
<td>N</td>
<td>L</td>
<td>L</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
7.4 Requirement Management (REQM)

7.4.1 Questions tailored for SCAMPI REQM area

Management (REQM)

1.1 Understand Requirements

1. There exist criteria for distinguishing appropriate requirements providers.
2. Objective criteria for the evaluation and acceptance of requirements is established in the Pre-game.
3. During Pre-game and sprint planning established requirements are reviewed to ensure that established criteria are met.
4. Product owner Reaches an understanding of requirements with requirements providers so that project participants can commit to them.

1.2 Obtain Commitment to Requirements

1. The impact of requirements on existing commitments is assessed during Planning meetings and Pre-game.
2. Product owner Negotiates and records the commitments with the senior members of the company.

1.3 Manage Requirements Changes
Chapter 7: Appendix

1. All requirements and requirements changes that are given to or generated by the project are documented.
2. Requirements change history, including the rationale for changes is maintained.
3. The impact of requirement changes from the standpoint of relevant stakeholders is evaluated during Planning and Review meetings.
4. Requirements and change data is available to the project.

1.4 Maintain Bidirectional Traceability of Requirements
1. Requirements traceability is maintained in the user stories to ensure that the source of lower level (i.e., derived) requirements is documented.
2. Requirements traceability from a requirement is maintained in the user stories to its derived requirements and allocation to work products.
3. Every requirement has a unique identifier, person who created it a who is responsible for it.

1.5 Ensure Alignment Between Project Work and Requirements
1. During Pre-game and sprint planning project plans, activities, and work products are reviewed for consistency with requirements and changes made to them.
2. During Pre-game and sprint planning the source of the inconsistency (if any) is identified.
3. During Pre-game and sprint planning any changes that should be made to plans and work products resulting from changes to the requirements baseline are identified.
4. Any necessary corrective actions are initiated.
### Chapter 7: Appendix

#### 7.4.2 Answers tailored for SCAMPI REQM area

![Table](image-url)

**Totals**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18</td>
<td>34</td>
<td>16</td>
<td>0</td>
<td>26%</td>
<td>50%</td>
<td>24%</td>
<td>0%</td>
</tr>
<tr>
<td>SCAMPI</td>
<td>67%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

- A: # Project Leaders
- B: # PM
- C: # Customer
- D: # Requirements
- E: # Requirements
- F: # Requirements
- G: # Requirements
- H: # Requirements
- I: # Customer
- J: # Requirements
- K: # Requirements
- L: # Requirements
- M: # Requirements
- N: # Requirements
- O: # Requirements
- P: # Requirements
- Q: # Requirements
- R: # Requirements
- S: # Requirements
- T: # Requirements
- U: # Requirements
- V: # Requirements
- W: # Requirements
- X: # Requirements
- Y: # Requirements
- Z: # Requirements

The table above represents the distribution of project leaders across different subpractices within the SCAMPI REQM area. The percentages indicate the proportion of project leaders in each subpractice, with 0% indicating no project leaders in that subpractice. The totals at the bottom reflect the overall distribution of project leaders and requirements across the entire SCAMPI REQM area.