NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

MAPPING CMMI PROCESS AREAS TO AGILE BEST PRACTICES

Ana Mafalda de Sousa Morais

Dissertation report presented as partial requirement for obtaining the master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2018

Title:
Subtitle:

Student
full name

MGI

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

MAPPING CMMI PROCESS AREAS TO AGILE BEST PRACTICES

by

Ana Mafalda de Sousa Morais

Dissertation report presented as partial requirement for obtaining the master's degree in Information Management, with a specialization in Information Systems and Technologies Management.

Advisor: Vitor Manuel dos Santos

November 2018

ACKNOWLEDGEMENTS

I would first like to thank my dissertation advisor, Professor Vitor Manuel dos Santos, of the Nova Information Management School. Throughout this course, whether in his office, by skype or by mobile phone, Professor Victor was always available for my doubts and insecurities. Whether I ran into a dead end or had a question about my research or writing he was there to guide me in the right direction whenever he thought I needed it.

I would also like to thank the companies that contributed to the study. Their contribution, not only with the survey but also with their willingness to help and give their opinions, was essential. Without their help, this study would not be possible.

To all my colleagues and friends from work that help me in the gloomy hours, when everything seems too much to handle. Studying and working at the same time is a challenge, but you were always there to help in any way possible. Thank you.

To my friends for supporting me, being patient, understanding my absence and always being there to make me believe that I can do it. Thank you for everything.

Finally, I want to express my very profound gratitude to my family and my boyfriend. The decision I made to move from Porto to Lisbon to do these masters has affected them in many ways, but I am very grateful to be able to look back and appreciate their unfailing support and continued encouragement from the very first day when all this was just a will. This accomplishment would not have been possible without them. Thank you for all the support.

ABSTRACT

To maintain competitiveness, software development companies are pressured to deliver products in less time, without compromising on quality and budget. To meet this demand, companies often adopt Agile software development techniques that enable shorter delivery times through constant smaller deliveries and shorter interactive cycles in software development processes. Although the innovation and the decrease of development time provided by these techniques, companies have perceived that quality is a differentiated factor and feel the urge to maintain the quality of their software to stand out from the competitors. At this point CMMI is presented as a reference model that contains a set of practices which lead to the maturity of organizations with focus on the improvement of organization processes and reduction of processes risk of failure increasing quality. In contrast with Agile, that puts individuals and their interactions in higher importance than processes and tools and where being adaptable to changes is more important than following strictly what was planned, CMMI it is a strict traditional approach that implies extensive formalism and focus on the processes. Derived of those apparent opposite beliefs, Agile development methods and CMMI best practices are frequently perceived to be at odds with each other. There is a great discussion about CMMI ability to be Agile and Agile methods to adapt to CMMI requirements without losing agility. This study aims to map CMMI for development process areas to Agile best practices to help in the understanding of the compatibilities and incompatibilities regarding the integration of CMMI and Agile. That will guide companies into the successful integration of Agile and maturity models together taking full advantage of their capabilities leading to enhanced software development. With the integration of those two approaches it is expected that Agile practices can help mature organizations to become more flexible, and CMMI could help Agile organizations to increase processes quality, fulfilling their goals and having their competitiveness sustained.

KEYWORDS

Software Development; Process Improvement; CMMI; Maturity Model; Agile Methods

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CM Configuration Management

CMM Capability Maturity Model

CMMI Capability Maturity Model Integration

CMMI-DEV Capability Maturity Model Integration for Development

DAR Decision Analysis and Resolution

FDD Feature-Driven Development

IPM Integrated Project Management

IT Information Technology

LS Largely Supported

MA Measurement and Analysis

OPD Organizational Process Definition

OPF Organizational Process Focus

OPM Organizational Performance Management

OT Organizational Training

PI Product Integration

PMC Project Monitoring and Control

PP Project Planning

PPQA Process and Product Quality Assurance

PS Partially Supported

QPM Quantitative Project Management

RD Requirements Development

REQM Requirements Management

RSKM Risk Management

S Supported

SAM Supplier Agreement Management

SEI Software Engineering Institute

SPI Software Process Improvement

TS Technical Solution

U Unsupported

UML Unified Modeling Language

VAL Validation

VER Verification

XP Extreme Programming

1. INTRODUCTION

We are currently living in an information age and as such the number of new software development companies has been growing substantially. With the increasing number of emerging software development companies, the need for each one of those companies to stand out in the market is higher than ever. Companies seek for excellence and differentiation in the market to leverage their business over their competitors and overcome this global competitiveness. More than ever companies focus on improving their processes through the search for higher quality, shorter cycle times and lower costs to fulfill customer requests.

In pursuance of those goals companies attempt to find ways to comply with the requirements of the market: lower delivery times, higher quality and lower prices.

To shorten development times and due to the accelerated rhythm of changes, companies adopt many times Agile methodologies for software development, those methods praise for "individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan" (Beck et al., 2001).

By other hand, the search for quality products has been motivating software development companies to improve their processes by the implementation of maturity models that ensure quality and excellence. Capability Maturity Model Integration (CMMI) is a process performance improvement model for competitive organizations that want to accomplish operations with high-performance. It helps identify and improve company key capabilities and elevate performance, quality, and profitability.

According to Matalonga (2012) Agile and CMMI methodologies have already been recognized to be compatible and synergistic. However, remains a great debate about the capability of CMMI to be Agile and Agile methodologies to adjust to CMMI metrics without losing agility. This study aims to develop this topic and contribute to a common understanding on how these two methods can work together and how they can take advantage of each other, through the study of the compatibilities and incompatibilities of these two approaches.

1.1. BACKGROUND AND PROBLEM DEFINITION

Two events occured in the beginning of the century that revolutionized the software and systems engineering community.

One of those events was the release of the Capability Maturity Model Integration (CMMI). It presented a wider, comprehensive model that took an approach centered in organizations processes allowing the alignment of operations to organizational goals, contributing to the improving of organizational performance and product quality.

The other event was the release of the Agile Manifesto. The signatories of this manifesto uncover better ways of developing software, it propose a common ground through collaboration while agreeing on a core set of people centered values in support of the emerging methods and techniques they were developing (Jeff Dalton et al., 2016).

CMMI it is a maturity model which aims to improve organizations processes capability, in other words, aims to improve processes ability to achieve the desired result, increasing process or product quality. CMMI was built considering three main dimensions: people, tools and procedures. The process serves to unite these dimensions. These methods are considered a symbol of traditional methods in software engineering.

Aiming to achieve higher quality and customer satisfaction, many companies, especially those using disciplined methods, rely on worldwide recognition models like CMMI that act as an indicator for organizational maturity. In fact, CMMI has promised better quality of products through process improvement. Generally, it means that customers have more reliability and trust in companies that have higher CMMI certification.

Staples & Niazi (2008) considered that the major reason behind the adoption of CMMI methodologies are the improvement of product quality and project performance (e.g., development time, development cost, and productivity).

Those CMMI models are oriented in two viewpoints, the quality of software processes and the quality of the software product. The quality of software processes intends to evaluate and improve the processes of the software development life cycle. The perspective on software product quality is the evaluation of the produced software product to guarantee its quality.

Both these perspectives are connected since the processes quality influences directly the product quality. This study will be addressed only to the quality question regarding software development processes.

Agile software development describes a set of principles for software development. It believes in adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages quick and flexible responsiveness to change. These principles support the definition and ongoing evolution of several software development methods.

Companies urge to decrease time to market and comply with the constant changes that arise at any stage and Agile methods propose to fulfill those requirements. Agile methods propose early and frequent deliveries, low ceremonies, lightweight documentation and working software. Gandomani & Zulzalil (2013) declare such values as the motivations behind both customers and software companies to use Agile methods for software development.

These values put Agile far away from the traditional, procedural and strict methods like CMMI.

When these two methodologies emerged the differing cultural context of their early adopters lead to misperceptions around the relationship between CMMI and Agile (Jeff Dalton et al., 2016). Nowadays the Software Engineering Community still not accept totally the idea of a unique software engineering discipline with traditional and innovative methods combined.

The division between Agile and traditional software engineering still exists and with CMMI being assigned to the traditional area, the discussion between Agile vs CMMI is a battle in the debate for software engineering discipline (Matalonga, 2012).

Software Engineering Institute (SEI), entity responsible for providing advanced software engineering practice by qualifying software quality grades, also consider that exists misunderstandings regarding the using of both methods simultaneously and believe there are two primary reasons for the discord between the Agile and CMMI communities (Glazer, Dalton, Anderson, Konrad, & Shrum, 2008):

1. Early adopters of both CMMI and Agile methods had opposite software development paradigms. CMMI adopters were developers of large-scale, risk averse, often with high levels of management supervision and hierarchical defined governance, while the early adopters of Agile methodologies were generally focused on smaller, single-team development projects with unstable requirements in a software-only environment;

2. These two extremes lead to the misconstruction of inaccurate information about CMMI and Agile and the misuse of both resulted in misperceptions in CMMI adopters about Agile and vice-versa. These negative perceptions that position CMMI and Agile at odds with each other derive mainly from the following factors:

a. Misuse — CMMI models practices were sometimes misused or applied to development activities that may have already been perceived by software development teams as productive without them;

b. Lack of Accurate Information — A lack of accurate information about CMMI in the Agile community and vice versa;

c. Terminology Difficulties — The use of specific terminology in CMMI (e.g., discipline, quality assurance, and predictability) and Agile methods (e.g., continuous integration, test-driven development, and collective code ownership) that carries context-specific connotations and is thus easily misunderstood and abused;

d. Top-Down Versus Bottom-Up Approach — The introduction of an approach that sometimes favors one voice (i.e., management versus practitioner) over the other, which neglects the other important voice in how to effectively run the business;

Another reason for many of the conflicts that arise when using the CMMI together with an Agile approach is the differing views on just what Agile is. Some view Agile simply as quick when making a decision or light when it comes to writing things down, but these popular misunderstandings of agility have led many organizations down unsuccessful paths (Paul E. McMahon, 2010).

Despite those remaining misperceptions, some organizations take advantage of CMMI and Agile methods together completing each other by creating synergies that benefit the organizations. Agile methods provide software development how-to's that are missing from CMMI best practices that work well, CMMI provides the systems engineering practices that help enable an Agile approach on large projects. CMMI also provides the process management and support practices that help deploy, sustain, and continuously improve the deployment of an Agile approach in any organization (Glazer et al., 2008).

1.2. STUDY RELEVANCE

Throughout this century software development increased in importance, complexity and size and the informalities very common in the early steps in this area proven to be one of the major reasons for failure.

In the present-day highly competitive world, software development companies feel the urge of outstand from the competitors and assure to their customers the development of tailor made, cost effective, high quality and scalable tailor-made solutions that could function faster, deliver quick results and be adaptable.

Thus, the needs for standards and processes for software development increase exponentially to reach excellency in the processes. The necessity of selecting and following formal process for software development provide the desired discipline to deliver highly quality products that assure business success and avoid wastage of time, money and resources.

However, as explained above, there is still many doubts about those standards and specially regarding CMMI and Agile co-existence still exists a lot of misunderstandings. These two methods working together are proven that can get improvements on business performance but there still many topics to enlighten and many misconceptions to clarify.

This study will update the state of art regarding the acceptance between the integrations of CMMI and Agile methods as it proposes to contribute to the advancement of knowledge and the clarification of misinterpretations regarding those methods and their compatibility. It could help to clarify those doubts and contribute to help companies in the implementation of this methods together efficiently.

For the software engineering community, this study will help to bring closure to the still not settled “Agile vs Traditional” debate. The interpretation of CMMI in an Agile context is still needed (Matalonga, 2012).

This study will help customers and organizations which are using CMMI as a quality indicator and process improvement model to mitigate their worries about the compatibilities between CMMI and Agile methodologies (Gandomani & Zulzalil, 2013).

Organizations would be well-advised and embrace both methods as complementary and with a shared vision : “delivering a high-quality product to the customer on time” (Glazer, Boehm, & Turner, 2010).

Major improvements in software development community must come from the understanding and acceptance of the differences between CMMI and Agile methods as well as the exploration of the advantages of both, that should enable new ways of combining their ideas leading to better approaches of software development which outcome in better development processes. Agile community might consider benefic and interesting the results from experimenting with this higher-level concept of certainty (Glazer et al., 2008).

1.3. GOALS

Considering the problem identified previously, regarding the perceptions of CMMI best practices and Agile development methods at odds with each other, this study purposes to execute an extensive research about CMMI and Agile methodologies, understand the stigmas regarding its co-existence, present their compatibilities and incompatibilities and shown how one method could improve the other. This study also intends to decrease confusion and conflict in the adoption of both approaches together.

The main goal of this study is to provide a matrix of the compatibility between CMMI-DEV process areas and agile best practices.

It should guide companies into the successful integration of those methodologies taking full advantage of their capabilities and leading to enhanced software development enabling the creation of products that meet the requirements of today's market.

In pursuance of the study main goal, specific objectives have been defined and enumerated:

- Deepen knowledge regarding CMMI and Agile methodologies;
- Analyze how companies are adopting CMMI and Agile together nowadays;
- Understand the difficulties and the advantages resultants of that integration.

The following specific questions are identified and at the end of these study should be answered.

Q1. Is the achievement of higher CMMI maturity levels important for companies?

Q2. Did companies consider compatible and valuable the coexistence between CMMI objectives and Agile best practices?

Q3. What are the points of greater and lesser compatibility between these two methodologies?

1.4. DOCUMENT STRUCTURE

This document is organized as follows: The first chapter contains an introduction to the addressed problem and a brief description of the problem and study relevance and goals.

Chapter two explores the concepts found in the literature regarding the topic of this dissertation. It is made an introduction to maturity models and further an extensive analysis on CMMI, exploring their models, representations, levels of capability and maturity as well as CMMI-DEV process areas. This chapter also contains a contextualization regarding software development methodologies, focusing on agile software development and exploring some of the most known agile methodologies and their best practices. Finally, a comparison between CMMI and Agile is made.

On chapter three, the research methodology used is described and chapter four presents the results of the conducted study and a discussion about the results.

Chapter five presents the conclusions of this work and what we can learn from it. This chapter contains a description of each answer gathered in the responses of the survey.

Finally, chapter six explore the limitations of this work and recommendations for future works.

2. LITERATURE REVIEW

2.1. MATURITY MODELS

A Maturity Model is a proved valuable technique in the measurement of different aspects of a process or an organization. Represents a path towards increasingly organized and systematic way of doing business in organizations. (Proença & Borbinha, 2016)

Maturity models are based on the premises that people, organizations, functional areas, processes, etc., evolve through a process of development or growth in the direction of a more advanced maturity, going through a distinct number of levels. A level in the model is a base from which an evolution to a higher maturity level can be planned and implemented (Goksen, Cevik, & Avunduk, 2015).

These models provide organizations a measuring for auditing and benchmarking, a measuring of progress assessment against objectives and an understanding of strengths, weaknesses and opportunities.

Architecturally, maturity models typically have “levels” along an evolutionary scale that defines measurable transitions from one level to another. Each level has unique attributes and if an organization demonstrates these attributes it has achieved both the level and the capabilities that the level represents (Caralli, Knight, & Montgomery, 2012).

Instead of creating their own maturity models, institutions usually adopt some parts of existing maturity models like COBIT, ITIL, PMI, CMMI, ISO among others.

Maturity models emerged in the early 1990s and the pioneer was the Capability Maturity Model (CMM), developed by the Software Engineering Institute (SEI).

Capability Maturity Model for the software development is a framework, which describes the key elements of an effective software development process. The CMM describes an evolutionary improvement path from an AdHoc, immature process to a mature disciplined process. It covers the practices for planning, engineering and managing software development and maintenance. These key practices improve the ability of the organization to meet the goals for cost, schedule, functionality and product quality (Goksen et al., 2015).

In 1997, the authority decided that the maturity models that are distributed across different areas were collected under one roof and so control can be done more easily. After this stage, the CMM model reaches the dimension of "Capability Maturity Model Integration - CMMI” on the basis of integration (Goksen et al., 2015).

Nevertheless, those ready to use models have some advantages and disadvantages. Between the advantages are the fact that are ready to be used so there is no need to spend time and think about the design, it was tested, and it continues to be developed since there are many people and institutions that uses and contribute to the improvement of the models.

Still there are some disadvantages like their low flexibility that may generalize and not express the actual dynamics of the institution.

2.2. CAPABILITY MATURITY MODEL INTEGRATION

The Capability Maturity Model Integration (CMMI) is one of the multiple ready to use maturity models. It was developed by the Software Engineering Institute (SEI) and it is presented as a process improvement maturity model for the development of products and services (Paul E. McMahon, 2010).

CMMI defines the most important elements that are required to build great products, or deliver great services, and wraps them all up in a comprehensive model that provides a clear definition of what an organization should do to promote behaviors that lead to improved performance.

Figure 1: Expected results of the implementation of CMMI (INSTITUTE, 2017)

The goal of CMMI is process improvement through a Software Process Improvement (SPI) framework. CMMI defines processes and activities to implement in the project and the way to be carried out (Majumdar, Ashiqe-Ur-Rouf, Islam, & Arefeen, 2011).

The main objective of CMMI is to reduce the cost of implementing improvements in processes by eliminating inconsistencies and establishing guidelines to assist organizations at various stages of a software project (planning, management, and others) (Selleri Silva et al., 2015).

2.2.1. Models

CMMI models are collections of best practices that help organizations to improve their processes. These models are developed by product teams with members from industry, government, and the Software Engineering Institute (SEI).

CMMI has the principle that quality is influenced by the process and aims at the organization processes. It offers four models to be used in different environments adapted to the business and processes of the companies: CMMI for Development, used in the process of developing products and services; CMMI for Acquisition, used in the processes of acquisition or outsourcing of goods and services; CMMI for Services, used in service delivery processes; and People CMM, used in the process of developing a capable workforce (Souza & Gomes, 2015).

That four models and corresponding focuses are showed on Figure 2.

Figure 2: CMMI Models (Institute, 2017)

The one which applies to product development industry is the model CMMI for development (CMMI-DEV) that focuses on engineering or developing products and services. It consists of best practices that address development activities that cover the product life cycle from conception through delivery and maintenance. This model is the focus of this study.

The CMMI-DEV model provides guidance for improving organization's capability to develop quality products and services that meet the needs of customers and end users. These best practices will help organizations to improve efficiency, speed, and product quality fueled by a lower number of defects (Software Engineering Institute, 2010).

2.2.2. Representations

According to CMMI model there are two different approaches to process improvement. These approaches are called "representations". Both representations provide ways to improve companies processes to achieve business objectives, and both provide the same essential content and use the same model components (Software Engineering Institute, 2010).

An organization can choose the approach to be used to improve its processes as needed. Choosing the Continuous representation allows organizations to define one or more specific process areas in which they want to increase capacity. The Staged representation allows the increase of all organizational maturity.

Both capability levels and maturity levels provide a way to improve the processes of an organization and measure how well organizations can and do improve their processes. However, the associated approach to process improvement is different.

The differences between the structures are subtle but significant and are explored in the following chapters.

2.2.2.1. Continuous

Using the continuous representation enables organizations to achieve "capability levels" that characterize the state of the organization's processes relative to an individual process area.

The continuous representation is concerned with selecting both a particular process area to improve and the desired capability level for that process area. (Software Engineering Institute, 2010)

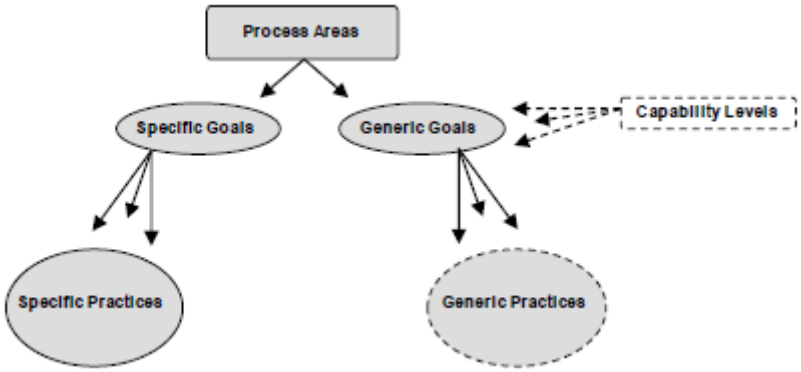


Figure 3: Continuous representation structure (Software Engineering Institute, 2010)

The four capability levels, each a layer in the foundation for continuing process improvement, are designated by: 0. Incomplete; 1. Performed; 2. Managed and 3. Defined.

A capability level for a process area is achieved when all the generic goals are satisfied up to that level.

2.2.2.2. Staged

The staged representation uses "maturity levels" to characterize the overall state of the organization's processes relative to the model as a whole.

The staged representation is concerned with selecting multiple process areas to improve within a maturity level, whether individual processes are performed or incomplete is not the primary focus (Software Engineering Institute, 2010). This representation is the focus of this study because nowadays it is the most used in IT companies.

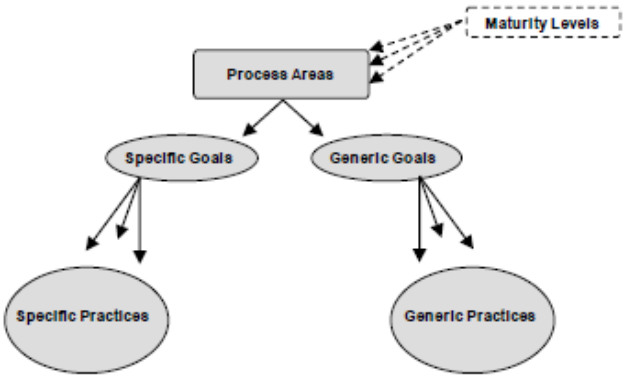


Figure 4: Staged representation structure (Software Engineering Institute, 2010)

The five maturity levels, each a layer in the foundation for continuing process improvement, are designated by: 1. Initial; 2. Managed; 3. Defined; 4. Quantitatively Managed and 5. Optimizing.

The maturity levels are measured by the achievement of the specific and generic goals associated with each predefined set of process areas.

2.2.3. Maturity Levels

CMMI methodology in staged representation, provides five maturity levels that demonstrate a visible path for improvement. As an organization advances its capabilities, it can expect to achieve a higher maturity level by identifying areas of improvement, working to correct these areas, and integrating solutions across the organization.

Each maturity levels comprise a predefined set of process areas. The maturity levels are measured by the achievement of the specific and generic goals that apply to each predefined set of process areas that improve the organization's overall performance (Institute, 2017).

According to Software Engineering Institute, studies have shown that organizations do their best when they focus their process improvement efforts on a manageable number of process areas at a time and that those areas require increasing sophistication as the organization improves (Software Engineering Institute, 2010).

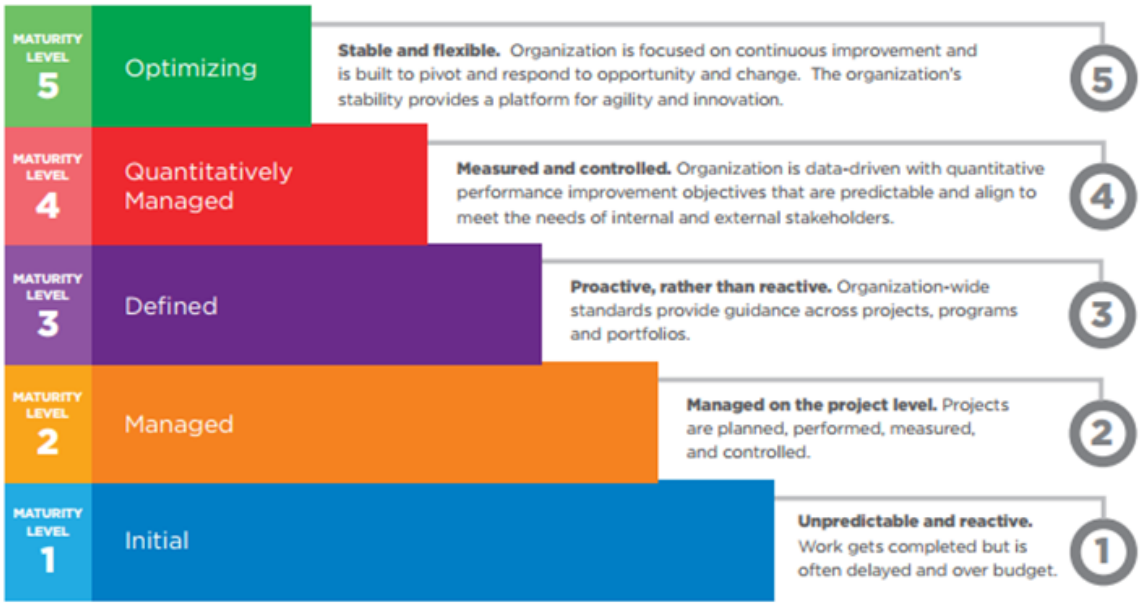


Figure 5: CMMI Maturity Levels (Institute, 2017)

These five maturity levels represent the expected performance index of an organization.

When using the staged representation, companies attain high maturity when achieve maturity level 4 or 5. Achieving maturity level 4 involves implementing all process areas for maturity levels 2, 3, and 4. Likewise, achieving maturity level 5 involves implementing all process areas for maturity levels 2, 3, 4, and 5 (Software Engineering Institute, 2010).

2.2.3.1. Maturity Level 1 - Initial

At maturity level 1, processes are usually ad hoc and chaotic. Organizations do not use any methodology and their processes are unpredictable and reactive (Soares & de Lemos Meira, 2013). That scenario usually means that companies does not have a stable environment and the success in these organizations depends on the competence and heroics of the people in the organization and not on the use of proven processes.

According to Software Engineering Institute, maturity level 1 organizations often produce products and services that work, however, they frequently exceed the budget and schedule of their projects.

This organizations are characterized by a tendency to over commit, abandon processes in time of crisis, and not be able to repeat their past successes (Software Engineering Institute, 2010).

2.2.3.2. Maturity Level 2 - Managed

At maturity level 2, an organization has achieved all the specific and generic goals of the maturity level 2 process areas. In other words, the projects of the organization have ensured that requirements are managed and that processes are planned, performed, measured, and controlled.

The process discipline reflected by maturity level 2 helps to ensure that existing practices are retained during times of stress. When these practices are in place, projects are performed and managed according to their documented plans.

According to Software Engineering Institute at this level both requirements, processes, work products, and services are managed. The status of the work products and the delivery of services are visible to management at defined points.

Commitments are established among relevant stakeholders and are revised as needed. Work products are reviewed with stakeholders and are controlled and the work products and services satisfy their specified requirements, standards, and objectives (Software Engineering Institute, 2010).

2.2.3.3. Maturity Level 3 - Defined

At maturity level 3, an organization has achieved all the specific and generic goals of the process areas assigned to maturity levels 2 and 3.

Maturity level 3 processes are well characterized and understood, and are described in standards, procedures, tools, and methods.

At maturity level 3 typically processes are described in more detail and more rigorously than at maturity level 2. Further, processes are managed more proactively using an understanding of the interrelationships of the process activities and detailed measures of the process, its work products, and its services (Software Engineering Institute, 2010).

2.2.3.4. Maturity Level 4 - Quantitatively Managed

At maturity level 4, an organization has achieved all the specific goals of the process areas assigned to maturity levels 2, 3, and 4 and the generic goals assigned to maturity levels 2 and 3.

At maturity level 4 sub processes are selected that significantly contribute to overall process performance. These selected sub processes are controlled using statistical and other quantitative techniques.

Quantitative objectives for quality and process performance are established and used as criteria in managing processes. Quantitative objectives are based on the needs of the customer, end users, organization, and process implementers. Quality and process performance are understood in statistical terms and are managed throughout the life of the processes.

For these processes, detailed measures of process performance are collected and statistically analyzed. Special causes of process variation are identified and, where appropriate, the sources of special causes are corrected to prevent future occurrences.

Quality and process performance measures are incorporated into the organizations measurement repository to support fact-based decision making in the future.

A critical distinction between maturity level 3 and maturity level 4 is the predictability of process performance. At maturity level 4, the performance of processes is controlled using statistical and other quantitative techniques and is quantitatively predictable. At maturity level 3, processes are only qualitatively predictable (Software Engineering Institute, 2010).

2.2.3.5. Maturity Level 5 - Optimizing

At maturity level 5, an organization has achieved all the specific goals of the process areas assigned to maturity levels 2, 3, 4, and 5 and the generic goals assigned to maturity levels 2 and 3.

Processes are continually improved based through both incremental and innovative technological improvements on a quantitative understanding of the common causes of variation inherent in processes.

Quantitative process-improvement objectives for the organization are established, continually revised to reflect changing business objectives, and used as criteria in managing process improvement.

The effects of deployed process improvements are measured and evaluated against the quantitative process-improvement objectives. Both the defined processes and the organization's set of standard processes are targets of measurable improvement activities.

Optimizing processes that are agile and innovative depends on the participation of an empowered workforce aligned with the business values and objectives of the organization. The organization's ability to rapidly respond to changes and opportunities is enhanced by finding ways to accelerate and share learning. Improvement of the processes is inherently part of everybody's role, resulting in a cycle of continual improvement.

A critical distinction between maturity level 4 and maturity level 5 is the type of process variation addressed. At maturity level 4, processes are concerned with addressing special causes of process variation and providing statistical predictability of the results. Though processes may produce predictable results, the results may be insufficient to achieve the established objectives. At maturity level 5, processes are concerned with addressing common causes of process variation and changing

the process (that is, shifting the mean of the process performance) to improve process performance (while maintaining statistical predictability) to achieve the established quantitative process-improvement objectives (Software Engineering Institute, 2010).

2.2.4. Process Areas

Process areas represent a set of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area. These process areas are organized according to the maturity levels, indicating which process areas to implement to achieve each maturity level.

At maturity level 1, processes are usually ad hoc and chaotic so there are no CMMI process areas involved.

Table 1 represent all the CMMI-DEV process areas by maturity level.

Table 1: CCMI-DEV Process Areas by Maturity Level (Software Engineering Institute, 2010)

Maturity Level	Process Area	Category
Maturity Level 2 Managed	Configuration Management (CM)	Support
	Measurement and Analysis (MA)	Support
	Project Monitoring and Control (PMC)	Project Management
	Project Planning (PP)	Project Management
	Process and Product Quality Assurance (PPQA)	Support
	Requirements Management (REQM)	Project Management
	Supplier Agreement Management (SAM)	Project Management
Maturity Level 3 Defined	Decision Analysis and Resolution (DAR)	Support
	Integrated Project Management (IPM)	Project Management
	Organizational Process Definition (OPD)	Process Management
	Organizational Process Focus (OPF)	Process Management
	Organizational Training (OT)	Process Management
	Product Integration (PI)	Engineering
	Requirements Development (RD)	Engineering
	Risk Management (RSKM)	Project Management
	Technical Solution (TS)	Engineering

	Validation (VAL)	Engineering
	Verification (VER)	Engineering
Maturity Level 4 Quantitatively Managed	Organizational Performance Management (OPM)	Process Management
	Quantitative Project Management (QPM)	Project Management
Maturity Level 5 Optimizing	Causal Analysis and Resolution (CAR)	Support
	Organizational Process Performance (OPP)	Process Management

2.2.4.1. Configuration Management (CM)

The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

The Configuration Management process area involves the following activities:

- Identifying the configuration of selected work products that compose baselines at given points in time;
- Controlling changes to configuration items;
- Building or providing specifications to build work products from the configuration management system;
- Maintaining the integrity of baselines Providing accurate status and current configuration data to developers, end users, and customers.

The work products placed under configuration management include the products that are delivered to the customer, designated internal work products, acquired products, tools, and other items used in creating and describing these work products.

CM process area contains the following specific goals and practices (Software Engineering Institute, 2010):

❖ SG 1 Establish Baselines

SP 1.1 Identify Configuration Items

SP 1.2 Establish a Configuration Management System

SP 1.3 Create or Release Baselines

❖ SG 2 Track and Control Changes

SP 2.1 Track Change Requests

SP 2.2 Control Configuration Items

- ❖ SG 3 Establish Integrity

SP 3.1 Establish Configuration Management Records

SP 3.2 Perform Configuration Audits

In agile environments, configuration management (CM) is important because of the need to support for frequent changes, frequent (usually daily) compilations, multiple baselines, and multiple CM workspaces (for example, for individuals, teams, and even for peer programming). Agile teams can get bogged down if the organization does not:

1) automate CM (for example, create scripts, status accounting, integrity check)

2) implement CM as a single set of standard services.

At the outset, an Agile team must identify the individual who will be responsible for ensuring that the CM is implemented correctly. At the beginning of each iteration, the CM support needs are confirmed again. CM is carefully integrated into the rhythms of each team with a concentrate on minimizing the distraction of the team to do the job. (Software Engineering Institute, 2010)

2.2.4.2. Measurement and Analysis (MA)

The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability used to support management information needs.

The Measurement and Analysis process area involves the following activities:

- Specifying objectives of measurement and analysis so that they are aligned with identified information needs and project, organizational, or business objectives;
- Specifying measures, analysis techniques, and mechanisms for data collection, data storage, reporting, and feedback;
- Implementing the analysis techniques and mechanisms for data collection, data reporting, and feedback;
- Providing objective results that can be used in making informed decisions and taking appropriate corrective action.

MA contains the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Align Measurement and Analysis Activities

SP 1.1 Establish Measurement Objectives

SP 1.2 Specify Measures

SP 1.3 Specify Data Collection and Storage Procedures

SP 1.4 Specify Analysis Procedures

- ❖ SG 2 Provide Measurement Results

SP 2.1 Obtain Measurement Data

SP 2.2 Analyze Measurement Data

SP 2.3 Store Data and Results

SP 2.4 Communicate Results

2.2.4.3. Project Monitoring and Control (PMC)

The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

PMC covers the following specific goals and practices (Software Engineering Institute, 2010):

- SG 1 Monitor the Project Against the Plan
  - SP 1.1 Monitor Project Planning Parameters
  - SP 1.2 Monitor Commitments
  - SP 1.3 Monitor Project Risks
  - SP 1.4 Monitor Data Management
  - SP 1.5 Monitor Stakeholder Involvement
  - SP 1.6 Conduct Progress Reviews
  - SP 1.7 Conduct Milestone Reviews
- SG 2 Manage Corrective Action to Closure
  - SP 2.1 Analyze Issues
  - SP 2.2 Take Corrective Action

SP 2.3 Manage Corrective Actions

In Agile environments, the sustained involvement of customers and potential end users in project product development activities can be crucial to project success; thus, the involvement of the client and the end user in the project activities should be monitored. (Software Engineering Institute, 2010)

2.2.4.4. Project Planning (PP)

The purpose of Project Planning (PP) is to establish and maintain plans that define project activities.

Project Planning process area involves the following activities:

- Developing the project plan;
- Interacting with relevant stakeholders appropriately;
- Getting commitment to the plan;
- Maintaining the plan.

PP specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Establish Estimates

SP 1.1 Estimate the Scope of the Project

SP 1.2 Establish Estimates of Work Product and Task Attributes

SP 1.3 Define Project Lifecycle Phases

SP 1.4 Estimate Effort and Cost

- ❖ SG 2 Develop a Project Plan

SP 2.1 Establish the Budget and Schedule

SP 2.2 Identify Project Risks

SP 2.3 Plan Data Management

SP 2.4 Plan the Project's Resources

SP 2.5 Plan Needed Knowledge and Skills

SP 2.6 Plan Stakeholder Involvement

SP 2.7 Establish the Project Plan

- ❖ SG 3 Obtain Commitment to the Plan

SP 3.1 Review Plans That Affect the Project

SP 3.2 Reconcile Work and Resource Levels

SP 3.3 Obtain Plan Commitment

In Agile environments, performing incremental development involves planning, monitoring, controlling, and re-planning more frequently than in more traditional development environments. While a high-level plan for the overall project or work effort is typically established, teams will estimate, plan, and carry out the actual work an increment or iteration at a time. Teams typically do not forecast beyond what is known about the project or iteration, except for anticipating risks, major events, and large-scale influences and constraints. Estimates reflect iteration and team specific factors that influence the time, effort, resources, and risks to accomplish the iteration. Teams plan, monitor, and adjust plans during each iteration as often as it takes (e.g., daily). Commitments to plans are demonstrated when tasks are assigned and accepted during iteration planning, user stories are elaborated or estimated, and iterations are populated with tasks from a maintained backlog of work. (Software Engineering Institute, 2010)

2.2.4.5. Process and Product Quality Assurance (PPQA)

The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products.

The Process and Product Quality Assurance process area involves the following activities:

- Objectively evaluating performed processes and work products against applicable process; descriptions, standards, and procedures;
- Identifying and documenting noncompliance issues;
- Providing feedback to project staff and managers on the results of quality assurance activities;
- Ensuring that noncompliance issues are addressed.

PPQA process area contains the following specific goals and practices (Software Engineering Institute, 2010):

❖ SG 1 Objectively Evaluate Processes and Work Products

SP 1.1 Objectively Evaluate Processes

SP 1.2 Objectively Evaluate Work Products

❖ SG 2 Provide Objective Insight

SP 2.1 Communicate and Resolve Noncompliance Issues

SP 2.2 Establish Records

In Agile environments, teams tend to focus on immediate needs of the iteration rather than on longer term and broader organizational needs. To ensure that objective evaluations are perceived to have value and are efficient, discuss the following early: (1) how objective evaluations are to be done, (2) which processes and work products will be evaluated, (3) how results of evaluations will be integrated into the team's rhythms (e.g., as part of daily meetings, checklists, peer reviews, tools, continuous integration, retrospectives). (Software Engineering Institute, 2010)

2.2.4.6. Requirements Management (REQM)

The purpose of Requirements Management (REQM) is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products.

REQM specific goals and practices (Software Engineering Institute, 2010):

❖ SG 1 Manage Requirements

SP 1.1 Understand Requirements

SP 1.2 Obtain Commitment to Requirements

SP 1.3 Manage Requirements Changes

SP 1.4 Maintain Bidirectional Traceability of Requirements

SP 1.5 Ensure Alignment Between Project Work and Requirements

In Agile environments, requirements are communicated and tracked through mechanisms such as product backlogs, story cards, and screen mock-ups. Commitments to requirements are either made collectively by the team or an empowered team leader. Work assignments are regularly (e.g., daily, weekly) adjusted based on progress made and as an improved understanding of the requirements and solution emerge. Traceability and consistency across requirements and work products is addressed through the mechanisms already mentioned as well as during start-of-iteration or end-of-iteration activities such as —retrospectives and —demo days. (Software Engineering Institute, 2010)

2.2.4.7. Supplier Agreement Management (SAM)

The purpose of Supplier Agreement Management (SAM) is to manage the acquisition of products and services from suppliers.

The Supplier Agreement Management process area involves the following activities:

- Determining the type of acquisition;
- Selecting suppliers;
- Establishing and maintaining agreements with suppliers;
- Executing supplier agreements;
- Accepting delivery of acquired products;
- Ensuring successful transition of acquired products.

SAM covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Establish Supplier Agreements

SP 1.1 Determine Acquisition Type

SP 1.2 Select Suppliers

SP 1.3 Establish Supplier Agreements

- ❖ SG 2 Satisfy Supplier Agreements

SP 2.1 Execute the Supplier Agreement

SP 2.2 Accept the Acquired Product

SP 2.3 Ensure Transition of Products

2.2.4.8. Decision Analysis and Resolution (DAR)

The purpose of Decision Analysis and Resolution (DAR) is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.

A formal evaluation process is a structured approach to evaluating alternative solutions against established criteria to determine a recommended solution.

A formal evaluation process involves the following actions:

- Establishing the criteria for evaluating alternatives;
- Identifying alternative solutions;
- Selecting methods for evaluating alternatives;
- Evaluating alternative solutions using established criteria and methods;
- Selecting recommended solutions from alternatives based on evaluation criteria.

DAR specific goals and practices (Software Engineering Institute, 2010):

❖ SG 1 Evaluate Alternatives

SP 1.1 Establish Guidelines for Decision Analysis

SP 1.2 Establish Evaluation Criteria

SP 1.3 Identify Alternative Solutions

SP 1.4 Select Evaluation Methods

SP 1.5 Evaluate Alternative Solutions

SP 1.6 Select Solutions

2.2.4.9. Integrated Project Management (IPM)

The purpose of Integrated Project Management (IPM) is to establish and manage the project and the involvement of relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

Integrated Project Management involves the following activities:

- Establishing the project's defined process at project startup by tailoring the organization's set of standard processes;
- Managing the project using the project's defined process;
- Establishing the work environment for the project based on the organization's work environment standards;

- Establishing teams that are tasked to accomplish project objectives Using and contributing to organizational process assets;
- Enabling relevant stakeholders' concerns to be identified, considered, and, when appropriate, addressed during the project;
- Ensuring that relevant stakeholders perform their tasks in a coordinated and timely manner; address project requirements, plans, objectives, problems, and risks; fulfill their commitments; and identify, track, and resolve coordination issues.

IPM covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Use the Project's Defined Process

SP 1.1 Establish the Project's Defined Process

SP 1.2 Use Organizational Process Assets for Planning Project Activities

SP 1.3 Establish the Project's Work Environment

SP 1.4 Integrate Plans

SP 1.5 Manage the Project Using Integrated Plans

SP 1.6 Establish Teams

SP 1.7 Contribute to Organizational Process Assets

- ❖ SG 2 Coordinate and Collaborate with Relevant Stakeholders

SP 2.1 Manage Stakeholder Involvement

SP 2.2 Manage Dependencies

SP 2.3 Resolve Coordination Issues

2.2.4.10. Organizational Process Definition (OPD)

The purpose of Organizational Process Definition (OPD) is to establish and maintain a usable set of organizational process assets, work environment standards, and rules and guidelines for teams.

OPD specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Establish Organizational Process Assets

SP 1.1 Establish Standard Processes

SP 1.2 Establish Lifecycle Model Descriptions

SP 1.3 Establish Tailoring Criteria and Guidelines

SP 1.4 Establish the Organization's Measurement Repository

SP 1.5 Establish the Organization's Process Asset Library

SP 1.6 Establish Work Environment Standards

SP 1.7 Establish Rules and Guidelines for Teams

2.2.4.11. Organizational Process Focus (OPF)

The purpose of Organizational Process Focus (OPF) is to plan, implement, and deploy organizational process improvements based on a thorough understanding of current strengths and weaknesses of the organization's processes and process assets.

OPF covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Determine Process Improvement Opportunities

SP 1.1 Establish Organizational Process Needs

SP 1.2 Appraise the Organization's Processes

SP 1.3 Identify the Organization's Process Improvements

- ❖ SG 2 Plan and Implement Process Actions

SP 2.1 Establish Process Action Plans

SP 2.2 Implement Process Action Plans

- ❖ SG 3 Deploy Organizational Process Assets and Incorporate Experiences

SP 3.1 Deploy Organizational Process Assets

SP 3.2 Deploy Standard Processes

SP 3.3 Monitor the Implementation

SP 3.4 Incorporate Experiences into Organizational Process Assets

2.2.4.12. Organizational Training (OT)

The purpose of Organizational Training (OT) is to develop skills and knowledge of people so they can perform their roles effectively and efficiently.

An organizational training program involves the following activities:

- Identifying the training needed by the organization;
- Obtaining and providing training to address those needs;
- Establishing and maintaining a training capability;
- Establishing and maintaining training records;

- Assessing training effectiveness.

OT specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Establish an Organizational Training Capability

SP 1.1 Establish Strategic Training Needs

SP 1.2 Determine Which Training Needs Are the Responsibility of the Organization

SP 1.3 Establish an Organizational Training Tactical Plan

SP 1.4 Establish a Training Capability

- ❖ SG 2 Provide Training

SP 2.1 Deliver Training

SP 2.2 Establish Training Records

SP 2.3 Assess Training Effectiveness

2.2.4.13. Product Integration (PI)

The purpose of Product Integration (PI) is to assemble the product from the product components, ensure that the product, as integrated, behaves properly (i.e., possesses the required functionality and quality attributes), and deliver the product.

PI covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Prepare for Product Integration

SP 1.1 Establish an Integration Strategy

SP 1.2 Establish the Product Integration Environment

SP 1.3 Establish Product Integration Procedures and Criteria

- ❖ SG 2 Ensure Interface Compatibility

SP 2.1 Review Interface Descriptions for Completeness

SP 2.2 Manage Interfaces

- ❖ SG 3 Assemble Product Components and Deliver the Product

SP 3.1 Confirm Readiness of Product Components for Integration

SP 3.2 Assemble Product Components

SP 3.3 Evaluate Assembled Product Components

SP 3.4 Package and Deliver the Product or Product Component

In Agile environments, product integration is a frequent, often daily, activity. For example, for software, working code is continuously added to the code base in a process called ―continuous integration.‖ In addition to addressing continuous integration, the product integration strategy can address how supplier supplied components will be incorporated, how functionality will be built (in layers vs. ―vertical slices‖), and when to ―refactor.‖ The strategy should be established early in the project and be revised to reflect evolving and emerging component interfaces, external feeds, data exchange, and application program interfaces. (Software Engineering Institute, 2010)

2.2.4.14. Requirements Development (RD)

The purpose of Requirements Development (RD) is to elicit, analyze, and establish customer, product, and product component requirements.

All development projects have requirements. Requirements are the basis for design. The development of requirements includes the following activities:

- Elicitation, analysis, validation, and communication of customer needs, expectations, and constraints to obtain prioritized customer requirements that constitute an understanding of what will satisfy stakeholders;
- Collection and coordination of stakeholder needs;
- Development of the lifecycle requirements of the product;
- Establishment of the customer functional and quality attribute requirements;
- Establishment of initial product and product component requirements consistent with customer requirements.

RD covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Develop Customer Requirements

SP 1.1 Elicit Needs

SP 1.2 Transform Stakeholder Needs into Customer Requirements

- ❖ SG 2 Develop Product Requirements

SP 2.1 Establish Product and Product Component Requirements

SP 2.2 Allocate Product Component Requirements

SP 2.3 Identify Interface Requirements

- ❖ SG 3 Analyze and Validate Requirements

SP 3.1 Establish Operational Concepts and Scenarios

SP 3.2 Establish a Definition of Required Functionality and Quality Attributes

SP 3.3 Analyze Requirements

SP 3.4 Analyze Requirements to Achieve Balance

SP 3.5 Validate Requirements

In Agile environments, customer needs and ideas are iteratively elicited, elaborated, analyzed, and validated. Requirements are documented in forms such as user stories, scenarios, use cases, product backlogs, and the results of iterations (working code in the case of software). Which requirements will be addressed in a given iteration is driven by an assessment of risk and by the priorities associated with what is left on the product backlog. What details of requirements (and other artifacts) to document is driven by the need for coordination (among team members, teams, and later iterations) and the risk of losing what was learned. When the customer is on the team, there can still be a need for separate customer and product documentation to allow multiple solutions to be explored. As the solution emerges, responsibilities for derived requirements are allocated to the appropriate teams. (Software Engineering Institute, 2010)

2.2.4.15. Risk Management (RSKM)

The purpose of Risk Management (RSKM) is to identify potential problems before they occur so that risk handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

Risk management can be divided into the following parts:

- Defining a risk management strategy;
- Identifying and analyzing risks;
- Handling identified risks, including the implementation of risk mitigation plans as needed.

RSKM specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Prepare for Risk Management

SP 1.1 Determine Risk Sources and Categories

SP 1.2 Define Risk Parameters

SP 1.3 Establish a Risk Management Strategy

- ❖ SG 2 Identify and Analyze Risks

SP 2.1 Identify Risks

SP 2.2 Evaluate, Categorize, and Prioritize Risks

- ❖ SG 3 Mitigate Risks

SP 3.1 Develop Risk Mitigation Plans

SP 3.2 Implement Risk Mitigation Plans

In Agile environments, some risk management activities are inherently embedded in the Agile method used. For example, some technical risks can be addressed by encouraging experimentation (early ―failures‖) or by executing a ―spike‖ outside of the routine iteration. However, the Risk Management process area encourages a more systematic approach to managing risks, both technical and non-technical. Such an approach can be integrated into Agile's typical iteration and meeting rhythms; more specifically, during iteration planning, task estimating, and acceptance of tasks. (Software Engineering Institute, 2010)

2.2.4.16. Technical Solution (TS)

The purpose of Technical Solution (TS) is to select, design, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product related lifecycle processes either singly or in combination as appropriate.

This process area focuses on the following:

- Evaluating and selecting solutions that potentially satisfy an appropriate set of allocated functional and quality attribute requirements;
- Developing detailed designs for the selected solutions;
- Implementing the designs as a product or product component.

TS covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Select Product Component Solutions

SP 1.1 Develop Alternative Solutions and Selection Criteria

SP 1.2 Select Product Component Solutions

- ❖ SG 2 Develop the Design

SP 2.1 Design the Product or Product Component

SP 2.2 Establish a Technical Data Package

SP 2.3 Design Interfaces Using Criteria

SP 2.4 Perform Make, Buy, or Reuse Analyses

- ❖ SG 3 Implement the Product Design

SP 3.1 Implement the Design

SP 3.2 Develop Product Support Documentation

In Agile environments, the focus is on early solution exploration. By making the selection and tradeoff decisions more explicit, the Technical Solution process area helps improve the quality of those decisions, both individually and over time. Solutions can be defined in terms of functions, feature sets, releases, or any other components that facilitate product development. When someone other than the team will be working on the product in the future, release information, maintenance logs, and other data are typically included with the installed product. To support future product updates, rationale (for trade-offs, interfaces, and purchased parts) is captured so that why the product exists can be better understood. If there is low risk in the selected solution, the need to formally capture decisions is significantly reduced. (Software Engineering Institute, 2010)

2.2.4.17. Validation (VAL)

The purpose of Validation (VAL) is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment.

VAL specific goals and practices (Software Engineering Institute, 2010):

- SG 1 Prepare for Validation

SP 1.1 Select Products for Validation

SP 1.2 Establish the Validation Environment

SP 1.3 Establish Validation Procedures and Criteria

- SG 2 Validate Product or Product Components

SP 2.1 Perform Validation

SP 2.2 Analyze Validation Results

2.2.4.18. Verification (VER)

The purpose of Verification (VER) is to ensure that selected work products meet their specified requirements.

VER process area contains the following specific goals and practices (Software Engineering Institute, 2010):

- SG 1 Prepare for Verification

SP 1.1 Select Work Products for Verification

SP 1.2 Establish the Verification Environment

SP 1.3 Establish Verification Procedures and Criteria

- SG 2 Perform Peer Reviews

SP 2.1 Prepare for Peer Reviews

SP 2.2 Conduct Peer Reviews

SP 2.3 Analyze Peer Review Data

- ❖ SG 3 Verify Selected Work Products

SP 3.1 Perform Verification

SP 3.2 Analyze Verification Results

In Agile environments, because of customer involvement and frequent releases, verification and validation mutually support each other. For example, a defect can cause a prototype or early release to fail validation prematurely. Conversely, early and continuous validation helps ensure verification is applied to the right product. The Verification and Validation process areas help ensure a systematic approach to selecting the work products to be reviewed and tested, the methods and environments to be used, and the interfaces to be managed, which help ensure that defects are identified and addressed early. The more complex the product, the more systematic the approach needs to be to ensure compatibility among requirements and solutions, and consistency with how the product will be used. (Software Engineering Institute, 2010)

2.2.4.19. Organizational Performance Management (OPM)

The purpose of Organizational Performance Management (OPM) is to proactively manage the organization's performance to meet its business objectives.

Business objectives that this process area might address include the following:

- Improved product quality (e.g., functionality, quality attributes);
- Increased productivity;
- Increased process efficiency and effectiveness;
- Increased consistency in meeting budget and schedule;
- Decreased cycle time;
- Greater customer and end-user satisfaction;
- Shorter development or production time to change functionality, add new features, or adapt to new technologies;
- Improved performance of a supply chain involving multiple suppliers;
- Improved use of resources across the organization.

OPM process area contains the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Manage Business Performance

SP 1.1 Maintain Business Objectives

SP 1.2 Analyze Process Performance Data

SP 1.3 Identify Potential Areas for Improvement

- ❖ SG 2 Select Improvements

SP 2.1 Elicit Suggested Improvements SP 2.2 Analyze Suggested Improvements

SP 2.3 Validate Improvements

SP 2.4 Select and Implement Improvements for Deployment

- ❖ SG 3 Deploy Improvements

SP 3.1 Plan the Deployment

SP 3.2 Manage the Deployment

SP 3.3 Evaluate Improvement Effects

2.2.4.20. Quantitative Project Management (QPM)

The purpose of Quantitative Project Management (QPM) is to quantitatively manage the project to achieve the project's established quality and process performance objectives.

The Quantitative Project Management process area involves the following activities:

- Establishing and maintaining the project's quality and process performance objectives;
- Composing a defined process for the project to help to achieve the project's quality and process performance objectives;
- Selecting sub processes and attributes critical to understanding performance and that help to achieve the project's quality and process performance objectives;
- Selecting measures and analytic techniques to be used in quantitative management;
- Monitoring the performance of selected sub processes using statistical and other quantitative techniques;
- Managing the project using statistical and other quantitative techniques to determine if the project's objectives for quality and process performance are being satisfied;
- Performing root cause analysis of selected issues to address deficiencies in achieving the project's quality and process performance objectives.

QPM covers the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Prepare for Quantitative Management

SP 1.1 Establish the Project's Objectives

SP 1.2 Compose the Defined Process

SP 1.3 Select Sub processes and Attributes

SP 1.4 Select Measures and Analytic Techniques

- ❖ SG 2 Quantitatively Manage the Project

SP 2.1 Monitor the Performance of Selected Sub processes

SP 2.2 Manage Project Performance

SP 2.3 Perform Root Cause Analysis

2.2.4.21. Causal Analysis and Resolution (CAR)

The purpose of Causal Analysis and Resolution (CAR) is to identify causes of selected outcomes and act to improve process performance.

The Causal Analysis and Resolution process area involves the following activities:

- Identifying and analyzing causes of selected outcomes. The selected outcomes can represent defects and problems that can be prevented from happening in the future or successes that can be implemented in projects or the organization;
- Taking actions to complete the following:
    - Remove causes and prevent the recurrence of those types of defects and problems in the future;
    - Proactively analyze data to identify potential problems and prevent them from occurring;
    - Incorporate the causes of successes into the process to improve future process performance.

CAR process area contains the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Determine Causes of Selected Outcomes

SP 1.1 Select Outcomes for Analysis

SP 1.2 Analyze Causes

- ❖ SG 2 Address Causes of Selected Outcomes

SP 2.1 Implement Action Proposals

SP 2.2 Evaluate the Effect of Implemented Actions

SP 2.3 Record Causal Analysis Data

2.2.4.22. Organizational Process Performance (OPP)

The purpose of Organizational Process Performance (OPP) is to establish and maintain a quantitative understanding of the performance of selected processes in the organization's set of standard processes in support of achieving quality and process performance objectives, and to provide process performance data, baselines, and models to quantitatively manage the organization's projects.

The Organizational Process Performance process area involves the following activities:

- Establishing organizational quantitative quality and process performance objectives based on business objectives;
- Selecting processes or sub processes for process performance analyses;
- Establishing definitions of the measures to be used in process performance analyses;
- Establishing process performance baselines and process performance models.

OPP process area contains the following specific goals and practices (Software Engineering Institute, 2010):

- ❖ SG 1 Establish Performance Baselines and Models

SP 1.1 Establish Quality and Process Performance Objectives

SP 1.2 Select Processes

SP 1.3 Establish Process Performance Measures

SP 1.4 Analyze Process Performance and Establish Process Performance Baselines

SP 1.5 Establish Process Performance Models

2.3. Software Development Methodologies

A software development methodology is a guideline that shows the way of managing a software development project. There are several methodologies but there is no methodology that could fill all the situations and projects. Every organization implements their software development projects management in a unique way, which vary from one project to the next.

The appropriate selection of a management structure can make a big difference in achieving a successful end result when measured in terms of cost, meeting deadlines, client happiness, robustness of software, or minimizing expenditures on failed projects (Young, 2010).

Vijayasarathy and Butler (2016) survey results indicate that although agile methodologies are more prevalent than ten years ago, traditional methodologies are still popular and organizations also use multiple methodologies on projects. The percentage of use of those different software development approaches are represented on Figure 6.

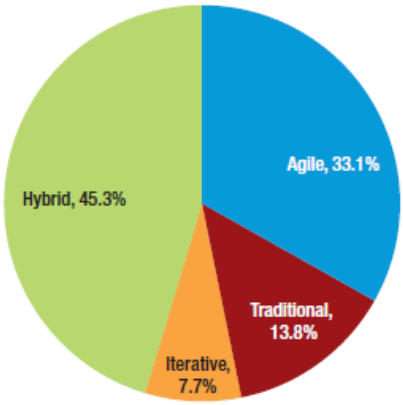


Figure 6: Software development approaches (Vijayasarathy & Butler, 2016)

Furthermore, the choice of methodologies of this survey respondents is associated with certain organizational, project, and team characteristics as shown on Table 2.

Table 2: Characteristics of projects following software development approaches (Vijayasarathy & Butler, 2016)

Approach	Characteristics		
	Organizational	Project	Team
Agile	Moderate revenue A small number of employees	Low budget Medium to high criticality	One team Small team
Traditional	High revenue A large number of employees	High budget High criticality	Multiple teams Medium team
Iterative	A small number of employees	Medium budget Medium to high criticality	One team Small team
Hybrid	Organization size unimportant	Medium budget High criticality	Small team

2.4. AGILE SOFTWARE DEVELOPMENT

Agile software development is considered a set of methods and practices applied to software development projects that are based on the values and principles expressed in the Manifesto for Agile Software Development and the corresponding twelve principles.

According to a recent survey (VERSIONONE.COM & COLLAB.NET, 2018) enterprise agility is increasing throughout organizations and across almost all industries at an accelerated rate. For the sixth year in a row, respondents continue to cite the same top benefits from adopting agile, they were the accelerated delivery, better project visibility, improved team productivity, and management of changing priorities, as shown on Figure 7.

Figure 7: Reasons for Adopting Agile (VERSIONONE.COM & COLLAB.NET, 2018)

2.4.1. Agile Manifesto

In 2001 a group of software developers drafted the manifesto for agile development. The manifesto called for the use of iterative methods for product development and emphasized the following four values:

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

At first glance, these four values appear in direct conflict with the traditional systems engineering processes; i.e., they appear to promote an undisciplined approach. But rather than being undisciplined, there is actually a rigid methodology to agile processes, as they require both consensus among the team and a high level of discipline to follow and execute the agreed upon rules and methods (Darrin & Devereux, 2017).

The Manifesto for Agile Software Development is based on twelve principles (Beck et al., 2001):

1. Customer satisfaction by early and continuous delivery of valuable software
2. Welcome changing requirements, even in late development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

2.4.2. Agile Methodologies

Being Agile isn't as simple as following one single methodology. In fact, Agile encompasses many different practices and frameworks, often referred to as "the Agile umbrella".

There is a great variety of agile software development frameworks but some of them are more popular and widely used.

In this chapter is made a small summary of some of the most used frameworks in agile software development according to the agile manifesto that highlights as key agile methodologies of software project management Extreme programming (XP), scrum and feature drive development (Beck et al., 2001).

In a recent report from the annual study of agile state we can observe that the most used methodologies by the respondents organizations are scrum and an hybrid between scrum and XP (VERSIONONE.COM, 2017). For that reason, the chosen focus methodologies of this study are Scrum and XP.

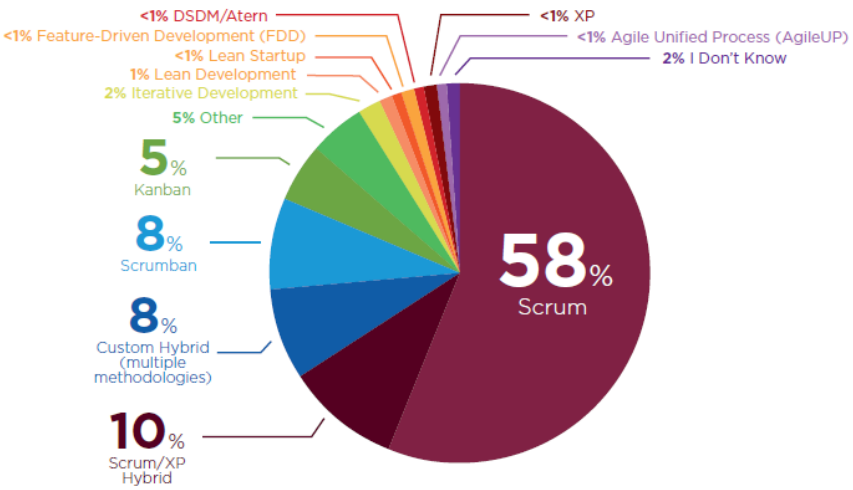


Figure 8: Agile Methodologies Used (VERSIONONE.COM, 2017)

2.4.2.1. Extreme Programming (XP)

XP is an agile methodology targeted on Software Engineering, and pays greater attention to programming than to management, as the former is the focus of Scrum, which is the reason why these methodologies are normally used together. It was created by Kent Beck in 1996 and seeks to improve a software project by using five essential values: communication, simplicity, feedback, respect and courage (Soares & de Lemos Meira, 2013).

XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development. Though larger projects have reported success, XP is set up for small groups of programmers, between 2 and 12 (Wells, 1999b).

Asking questions, negotiating scope and schedules, and creating functional tests require more than just the developers to be involved in producing the software, for that reason, XP teams includes not only the developers, but the managers and customers as well, all working together.

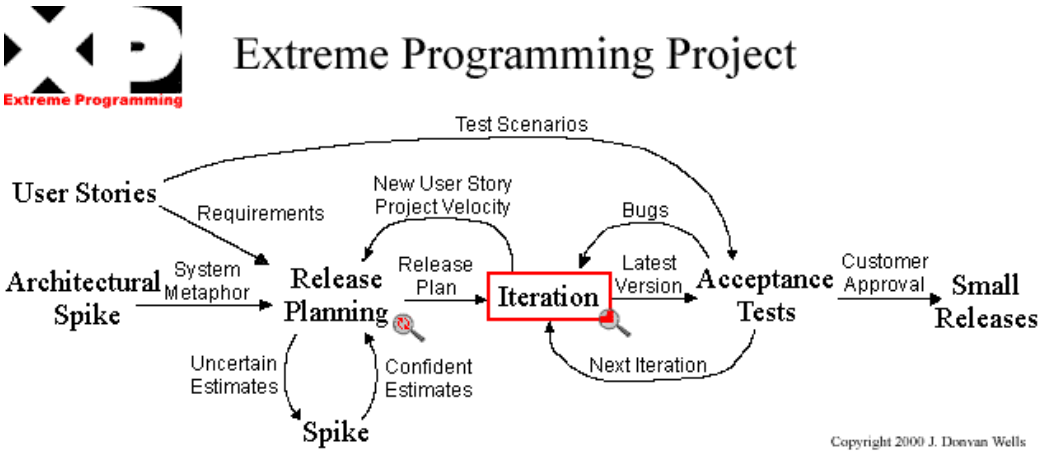


Figure 9: XP Programming Project (Wells, 1999a)

2.4.2.2. Scrum

Scrum is a process framework that has been used to manage work on complex products since the early 1990s. Scrum is not a process, technique, or definitive method. Rather, it is a framework within which organizations can employ various processes and techniques. Scrum makes clear the relative efficacy of product management and work techniques so that organizations can continuously improve the product, the team, and the working environment (Schwaber & Sutherland, 2017).

Scrum is a framework for planning and monitoring a project. Since it is iterative and incremental, it works well in an environment of constant change. It supplies self-managing teams and proposes a form of flexible and adaptable work, not only in relation to the scope and requirements of a project, but also with regard to the exchange of teams, tools, programming languages, etc. (Soares & de Lemos Meira, 2013).

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master. Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

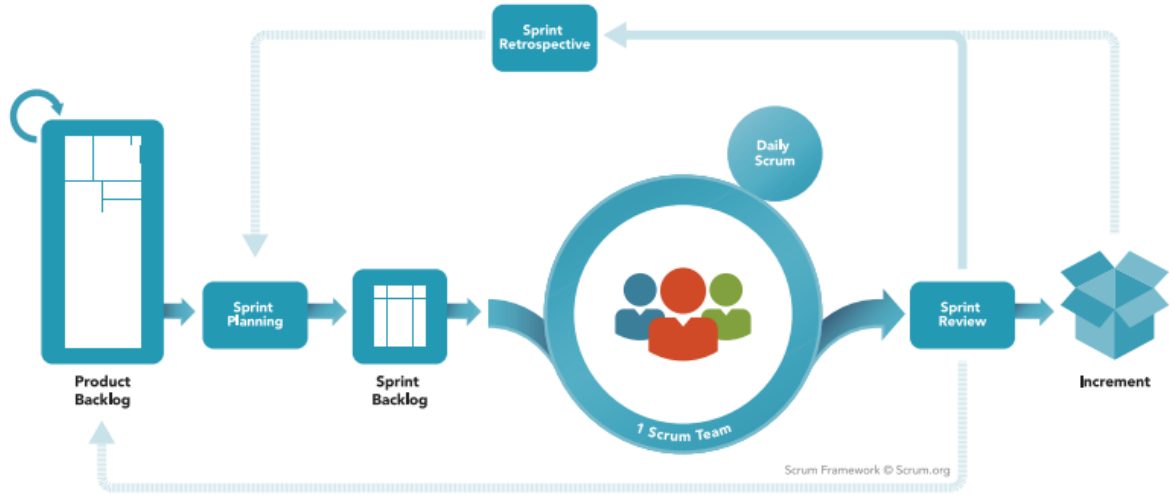


Figure 10: The scrum framework (Scrum.org, 2018)

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of the product ensure a potentially useful version of working product is always available (Schwaber & Sutherland, 2017).

When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone. The successful use of Scrum depends on people becoming more proficient in living these five values.

2.4.3. Agile Best Practices

As developed throughout these chapter, there are so many different Agile frameworks, a challenge many organizations or individuals face is deciding which framework to use and in which context. Different agile methodologies have different best practices. So Deloitte Consulting developed the map displayed on Figure 11 of the different Agile methodologies and their best practices that could help organizations to succeed in adopting Agile ways of working, understanding the right method to be used to address the specific problem or need of the company (Webb, 2016).

Agile is represented as a highly interconnected landscape of practices transporting ideas across zones to value. There is no starting point, nor an express line or a direct route suiting all conditions.

Exploring the agile landscape allow us to agree that organizations need to try, learn and adapt practices to their specific situations. No one framework is better that another, although when it is contextually applied, a framework or selection of practices from multiple frameworks will be best suited.

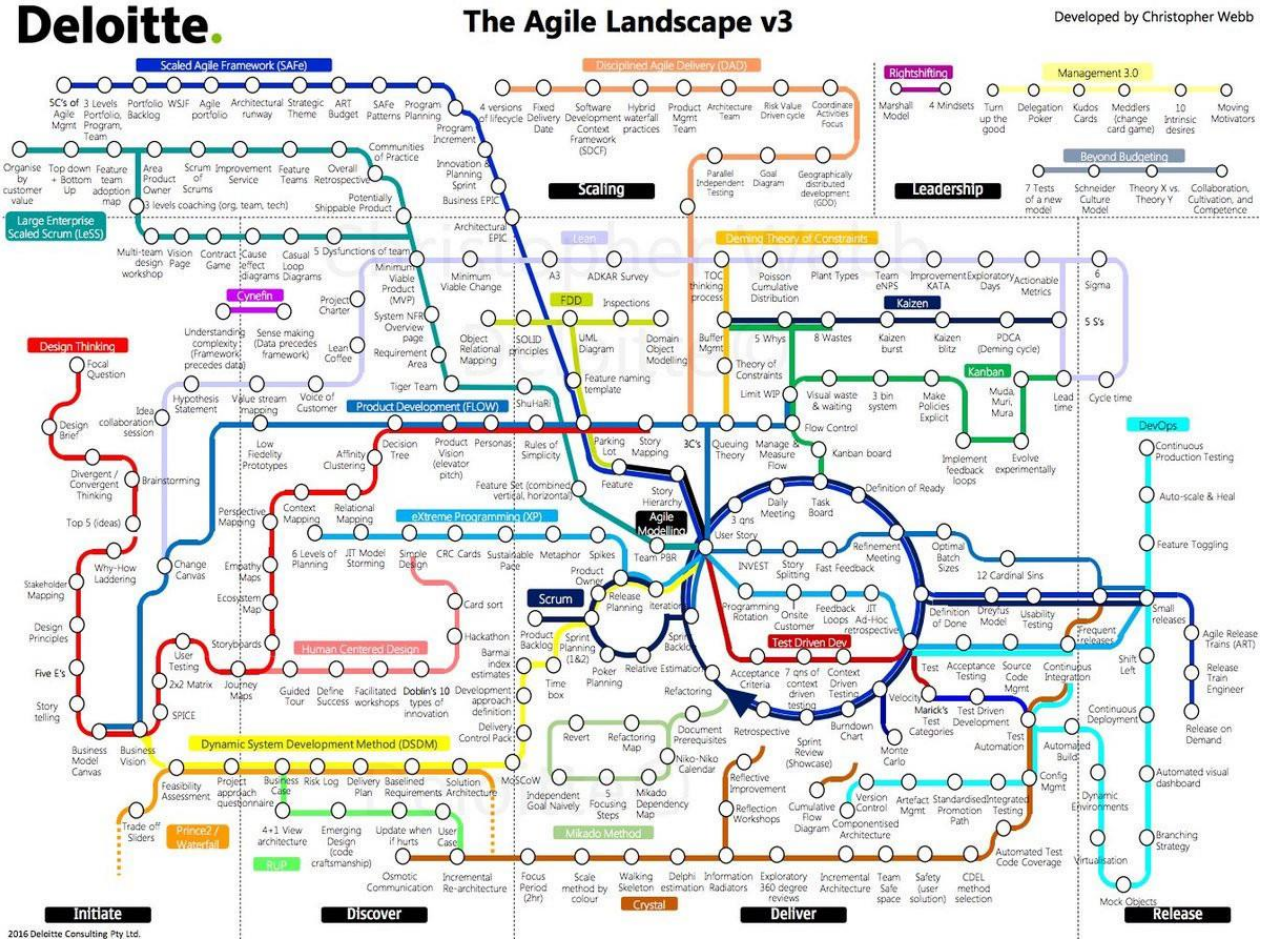


Figure 11 : Deloitte Agile Methodologies and their best practices (Webb, 2016)

In the following pictures from (Selleri Silva et al., 2015) and (VERSIONONE.COM, 2017) are visible the most used agile practices. That common practices will be the selected to be on focus on this study.

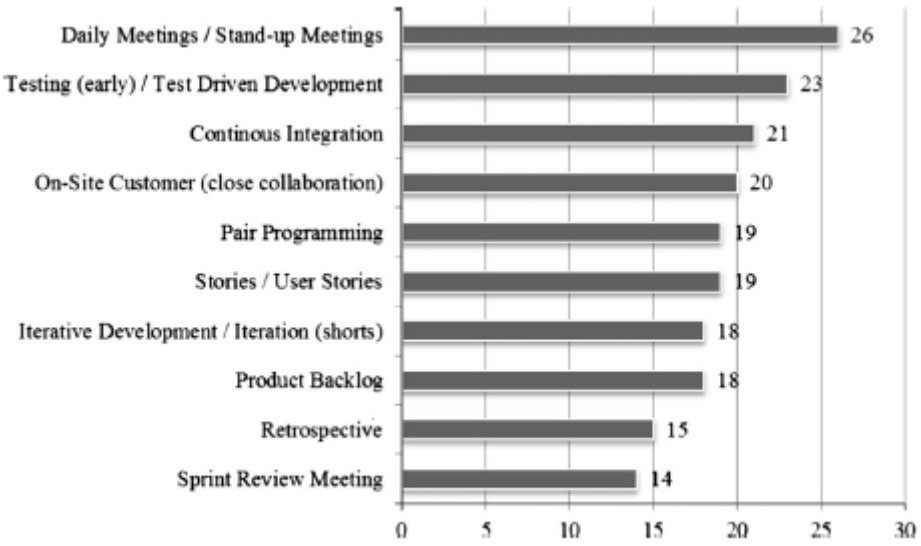


Figure 12: Agile practices most cited in studies (Selleri Silva et al., 2015)

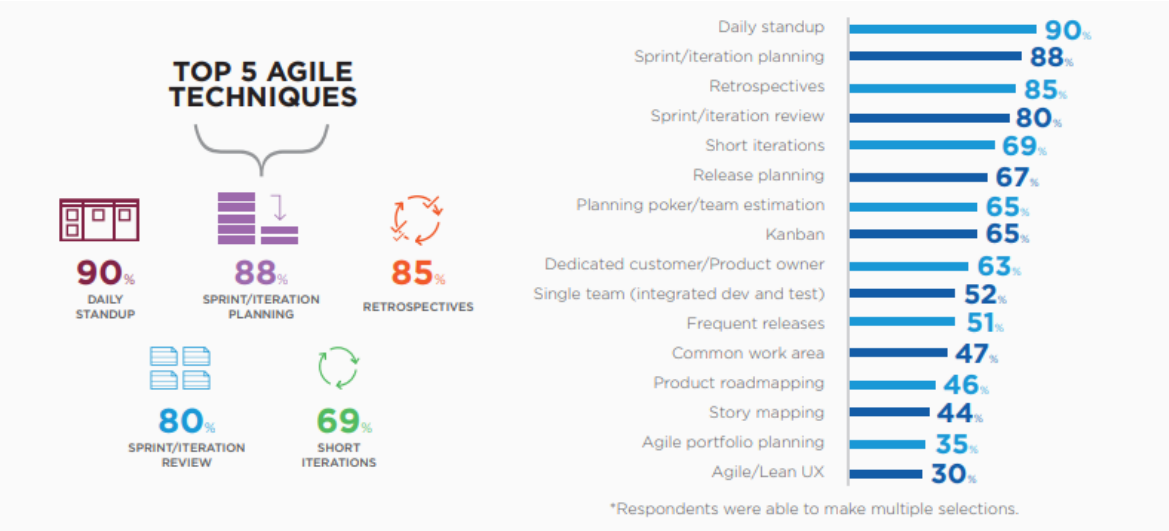


Figure 13: Agile Techniques Employed (VERSIONONE.COM & COLLAB.NET, 2018)

For this study six of the most used best practices in agile were selected to make the map of compatibility between the CMMI process areas and these agile best practices. The selected practices were Daily Meetings / Daily Standup, Customer Collaboration, Iterative Development / Short Iterations, Retrospectives, Sprint/Iteration Review and Sprint/Iteration Planning.

2.5. CMMI AND AGILE INTEGRATION

CMMI and agile focus on software process from different perspectives. While CMMI focuses on a strict, predictable, well-documented and plan-driven process, agile emphases on individual collaboration, embracing change and light-weight formalities. Both of two approaches are attractive and useful on their own. The main issue is compatibility of these approaches (Gandomani & Zulzalil, 2013).

Comparisons between CMMI and Agile have often been criticized comparing them like oil and water. However, the literature has summarized that CMMI and agile are compatible because agile methods are development process descriptions and CMMI is a reference process model that it is used for appraisals and improvements. This means, CMMI tells us what to do, while agile methods tell us how to do it (Paper, Garbajosa, & Calvo-manzano, 2009).

There are plenty of studies regarding the integration of CMMI with agile. According to Jeff Dalton et al. the percentage of CMMI appraisals with an agile component has been growing in the last decade.

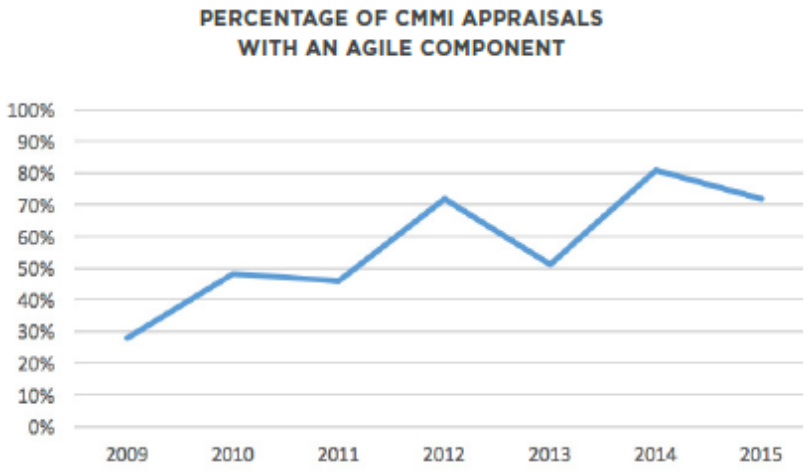


Figure 14: Percentage of CMMI Appraisals with an agile component (Jeff Dalton et al., 2016)

While on the face they seem like competing notions, at the heart the agile project teams and the CMMI process teams are both on a quest to improve the productivity of their software projects and the quality of the products they deliver. When examined at a deeper level it is easy to see that there is a great deal of synergy between the two and when this synergy is realized, and exploited organizations will be able to optimize their software development processes.

According to (ICEAA, 2017) there is no reason CMMI and Agile cannot be successfully applied together. Some failures have been noted but most of them are associated with one of the following situations:

• CMMI implementation is an exact interpretation of the generic and specific practices guidelines from the CMMI manual – creating a heavy document, rigid interpretation of CMMI

• Agile implementation is not so much agile as it is a wild west interpretation of agile where there is no planning, process or oversight

• CMMI team has had a bad experience with agile

• Agile team has had a bad experience with CMMI (or CMM)

• Organizational culture creates barrios to prevent either the adoption of agile or adoption of CMMI

In their study, Gandomani & Zulzalil, run a questionnaire on each process area and the compatibility with agile, the result is shown on Table 3. And we can conclude from the results that while agile is supportive and neutral in lower levels, there are major conflicts in higher levels. Especially when CMMI stresses on the notion of organization, agile is in conflict with it (Gandomani & Zulzalil, 2013).

Table 3: Study participants agreement on CMMI process areas compatibility with agile (Gandomani & Zulzalil, 2013)

CMMI level	Process Area	The Dominant View	Percentage
2	Measurement and Analysis (MA)	Neutral	60.3
2	Supplier Agreement Management (SAM)	Neutral	65.4
2	Configuration Management (CM)	Neutral	70.3
2	Project Planning (PP)	Support	66.2
2	Project Monitoring and Control (PMC)	Neutral	61.3
2	Process and Product Quality Assurance (PPQA)	Neutral	63.4
2	Requirements Management (REQM)	Support	69.8
3	Integrated Project Management (IPM)	Support	65.5
3	Decision Analysis and Resolution (DAR)	Conflict	55.2
3	Organizational Process Focus (OPF)	Conflict	55.3
3	Organizational Process Definition (OPD)	Support	54.7
3	Organizational Training (OT)	Support	55.7
3	Risk Management (RSKM)	Support	60.4
3	Requirements Development (RD)	Support	58.0
3	Product Integration (PI)	Support	59.0
3	Technical Solution (TS)	Support	48.3
3	Validation (VAL)	Support	49.6
3	Verification (VER)	Support	53.2
4	Organizational Process Performance (OPP)	Conflict	55.1
4	Quantitative Project Management (QPM)	Support	49.4
5	Organizational Performance Management (OPM)	Support	52.7
5	Causal Analysis and Resolution (CAR)	Neutral	44.8

In their study, (Sutherland et al., 2007), conclude that a CMMI level 5 company is able to deliver what the customer has ordered on schedule, cost and quality using 69% effort compared to a CMMI Level

1 company. On Figure 15 its visible that replacing some core processes with Scrum drives cost down another 34%, cuts process overhead by more than 50% and drives defects down by 40%.

That show us that the synergy between these two methodologies could in fact work perfectly.

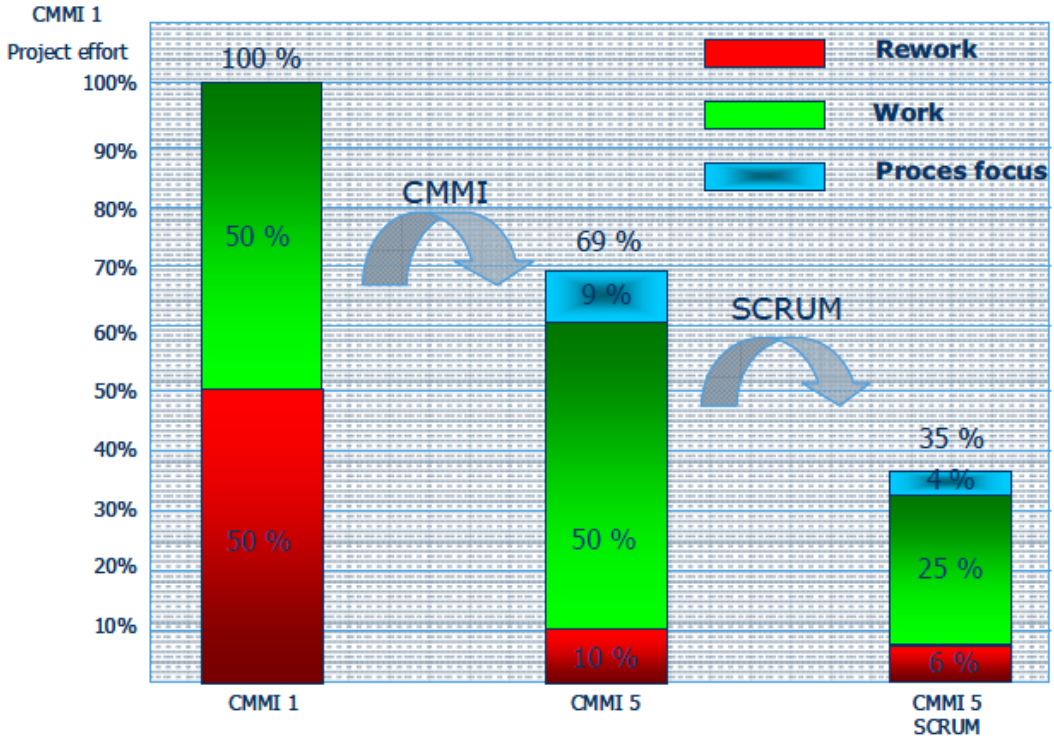


Figure 15: CMMI and Scrum Productivity Gains (Sutherland et al., 2007)

CMMI provides insight into what processes are needed to maintain a disciplined mature organization capable of predicting and improving performance of the organization and projects. Scrum provides guidance for efficient management of projects in a way that allows for high flexibility and adaptability. When mixing the two, a magic potion emerges, where the mindset from Scrum ensures that processes are implemented efficiently while embracing change, and CMMI ensures that all relevant processes are considered.

Individually CMMI and Scrum have proven benefits but also pitfalls. An Agile company may implement Scrum correctly but fail due to lack of institutionalization, or insufficient execution of engineering or management processes. CMMI can help Agile companies to institutionalize Agile methods more consistently and clarify what processes need improvement.

A company can comply with CMMI but fail to reach optimal performance due to poor process implementation.

Scrum and other Agile methodologies can guide such companies towards more efficient implementation of CMMI process requirements (Sutherland et al., 2007).

3. METHODOLOGY

This study comprises an exploratory research, which pursues to know the facts and phenomena related to the subject in study in order to provide greater familiarity with the problem.

It involves a bibliographical exploration that will take place with the objective of gather knowledge about CMMI and Agile methodologies, collecting and analyzing the main information about the theme and its study in the real context.

The most suited approach to the studied situation is to carry out surveys with people who could experience the problem and can share experiences to gather information about the Portuguese reality on this subject compared with the reality found in the bibliography. For the selection of those companies a research was made to find Portuguese companies that correspond to the requirements of the study.

The surveys enable the exploration, analyses and description of the specific situation that each company faces, to understand how companies are integrating CMMI and Agile, how they are taking advantage of that and what outcomes they accomplish with adoption of CMMI and Agile methods in software development.

This methodology is qualitative, when considering that the dissertation counts on surveys for interpretation of the phenomenon in analysis and data from that studies will be carried out in an inductive way.

The objective is to take considerations about the surveys and compare it with the cases present in the literature to expose their synthesis and to conclude the research questions by promoting the discussion and confrontation with theory.

3.1. SURVEY SUBJECTS

With the purpose of understand how Portuguese companies are adopting CMMI and Agile together nowadays, a survey was carried out in some Portuguese software development companies that have CMMI certification.

CMMI institute make available the appraisal results of all companies that obtained CMMI certification. These results are published online and work as a knowledge base of companies that have CMMI certification in Portugal, their CMMI level and who is the sponsor of this certification. The subjects of this interview were these sponsors, or someone nominated by them.

Since sampling for companies which have CMMI for development certification and follow agile methodologies is not very large, we decide to conduct interviews at two companies per CMMI level since we assume that companies in the same level conduct the same practices. However, we can't fulfill this objective since there is no companies with CMMI-DEV level 4 in Portugal and there is only one company with CMMI level 5.

For that reason, the subject of this study were two companies with CMMI Level 2, two companies with CMMI Level 3 and one with CMMI Level 5.

The list of Table 4 represents the Portuguese companies that are certified in CMMI and that are used for this research. (INSTITUTE, n.d.)

We choose to guarantee the confidentiality of the companies that participate in the study given that our sample of Portuguese companies that fulfilled the requirements to participate in the study was small. For that reason, organization names were not disclosed choosing to rank companies with numbers from 1 to 5.

Table 4: Portuguese companies with CMMI-DEV Staged representation (INSTITUTE, n.d.)

Organization	Organizational Unit	CMMI Level
Company 1	Software Development Projects	CMMI-DEV v1.3(Staged): Maturity Level 2
Company 2	Technology (P.TEC)	CMMI-DEV v1.3(Staged): Maturity Level 2
Company 3	Software Development	CMMI-DEV v1.3(Staged): Maturity Level 3
Company 4	Software Development	CMMI-DEV v1.3(Staged): Maturity Level 3
Company 5	Delivery	CMMI-DEV v1.3(Staged): Maturity Level 5

The survey was conducted in these five companies and allowed us to understand the position of the companies regarding CMMI, Agile and the combination of both.

3.2. SURVEY QUESTIONS

The survey is composed of general company questions followed by questions to understand which methodologies are used in the company regarding CMMI and Agile, why they are used, what are their advantages and disadvantages, what has changed since they implemented the methodologies and their ambitions for the future regarding the will to reach higher CMMI levels (Questions 1-29).

Finally, to understand the opinion of the companies regarding CMMI objectives coexistence with Agile practices, a correlation matrix was carried out between each one of the 22 CMMI process areas and the most commonly used agile practices (Questions 30-51).

Depending on company responses to questions, the path followed in the survey is different. The survey question flow represents the way forward for each response and is shown in Figure 16.

In annex A1. Survey Questions, the questions corresponding to the question flow above are presented.

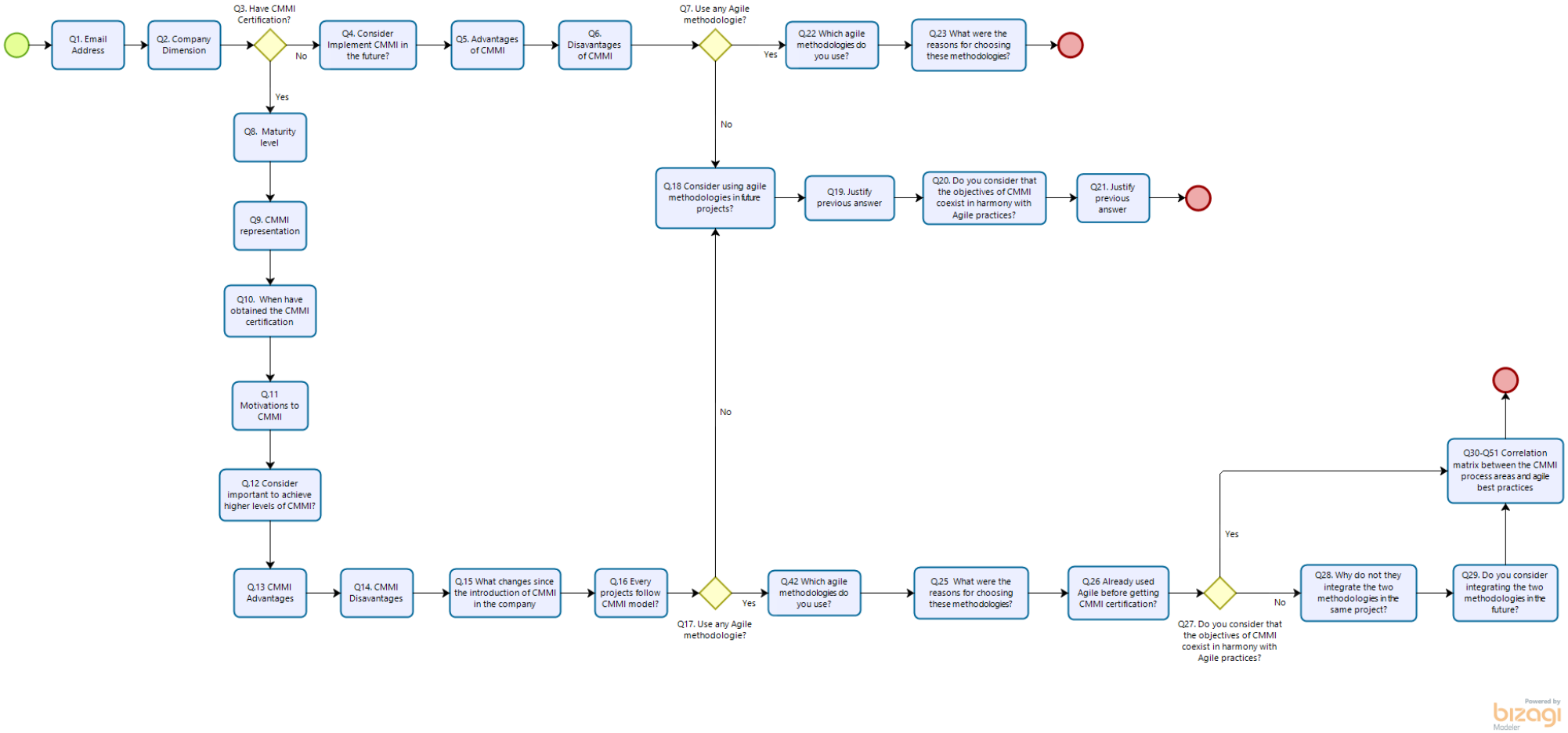


Figure 16: Survey Question Flow

4. RESULTS AND DISCUSSION

4.1. RESULTS

We were able to successfully get the involvement of all the companies that we target as relevant for the study. So, in total we gather five responses from five different companies. The subjects that answer the survey questions were highlighted by the company as the most knowledgeable within the company regarding CMMI and Agile.

The first sections of the survey aim to know more about the company and to classify it regarding the use of agile methodologies and CMMI and their motivations.

According to their dimension, 80% of the companies targeted for this study are classified as large companies since they employ more than 250 employees. The remaining 20%, corresponding to 1 company, were classified as medium company as it hires less than 250 employees.

All the companies participating in this study are certified in CMMI. Two of them have Managed CMMI (Level 2), another two have Defined CMMI level (Level 3) and one have Optimizing CMMI level (Level 5).

The company that has CMMI certification the longest is the company that owns CMMI Level 5. The company has been certified in CMMI for more than 5 years.

According to the answers of Question 11 (Q11 of Figure 16) the motivations for choosing to work with CMMI were the following:

Company 1 enhances processes and quality improvements as well as international visibility. Company 2 emphasizes the improvement of the implementation of development processes and the added value that CMMI brings in which makes it possible for the company to compete for certain projects. Company 3 highlights the recognition of the robustness of their development system by customers. Company 4 values the improvements in the quality of its processes of innovation and development of projects or products. In their observation, this way the whole company can follow processes well established at the organizational level. Company 5 classifies as greater motivation to obtain this certification the differentiation between competition and the fact that it is a way of improving internal execution.

In relation to the importance of achieving high CMMI levels, 60% of the companies do not consider it important to achieve higher levels of CMMI. Only two of the companies believe that it is important to achieve high maturity levels.

The advantages of CMMI model identified by the companies involved in this study are shown in the Figure 17.

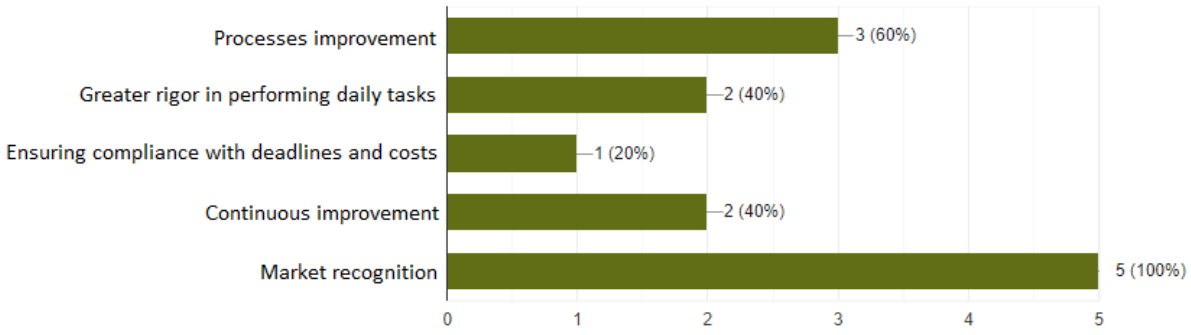


Figure 17: CMMI Advantages

The disadvantages of CMMI model identified by the companies involved in this study are shown in the Figure 18.

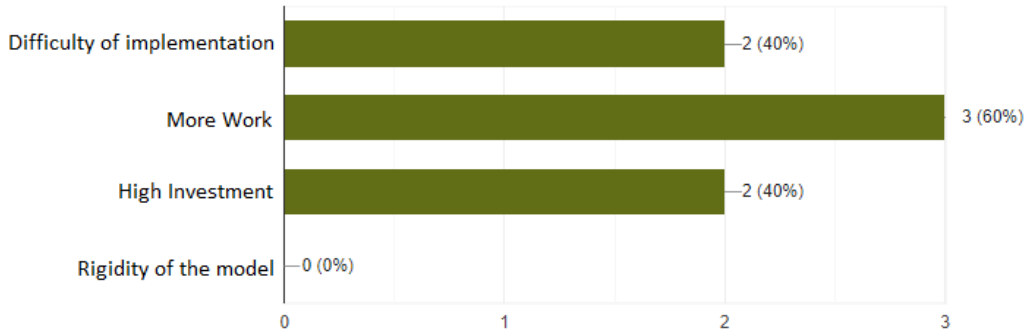


Figure 18: CMMI Disadvantages

To understand the major differences between before and after CMMI we asked the companies what has changed since the introduction of this model in the company. The Company 1 emphasized that there are now well-defined processes and control mechanisms that did not exist before. Company 2 indicates that there is currently a greater rigor in the implementation of metrics and indicators of progress and monitoring of projects. The company 3 has chosen to disclose the obligation to periodically verify that processes continue to comply with CMMI objectives as one of the biggest changes. The biggest change for company 4 was the improvement in the quality of deliveries, documentation and evidence. The company 5 emphasized that since CMMI Level 3 there is a greater predictability, less rework, better customer satisfaction and more project control.

All companies from this study follow an agile methodology. The methodologies used by the companies are represented with their respective weight in Figure 19.

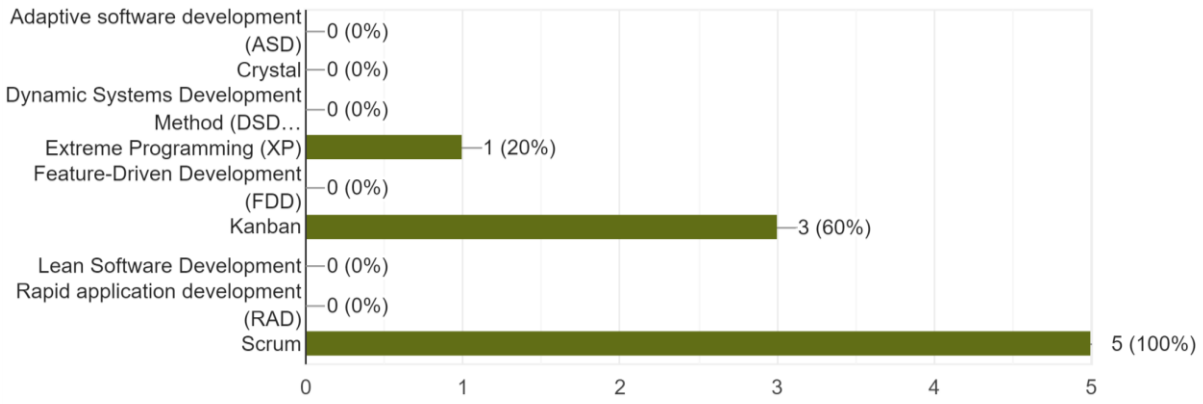


Figure 19: Agile methodologies used by companies involved in the study

There are numerous different reasons behind the chosen of an Agile methodology. We try to comply the most commons and Figure 20 represents the ones chosen by the companies in this study.

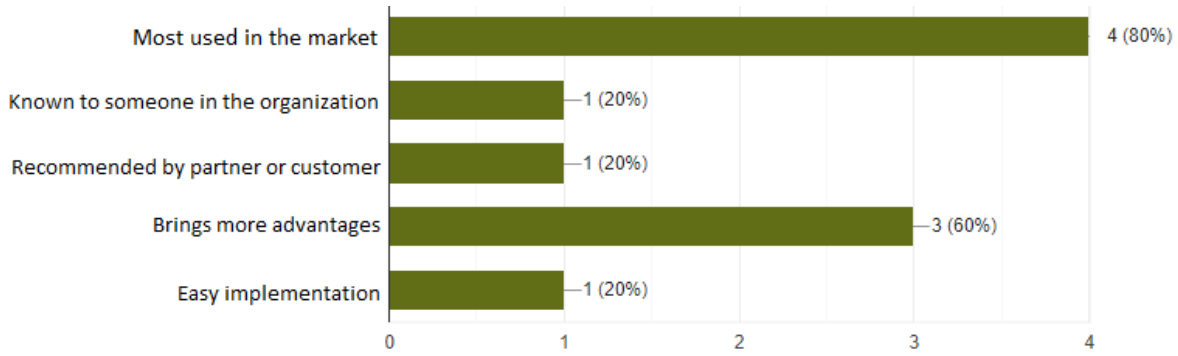


Figure 20: Reasons for choosing Agile Methodology

Previously to the implementation of CMMI model in this companies, 60% (3 companies) already used Agile and 40% (2 companies) did not use Agile previously to the implementation of CMMI model.

One of the biggest questions addressed by this study are the compatibility versus incompatibility between CMMI and Agile, for that reason we asked companies if they considered that the objectives of the CMMI coexist in harmony with Agile practices. The graphic of Figure 21 represents the answers obtained, four companies, corresponding to 80%, think that these two methodologies coexist harmoniously while one company considered they did not coexist.

The company that consider that the harmoniously coexistence is not possible between these two methodologies is a company with CMMI level 2 and although they sometimes use these methodologies simultaneously they consider that Scrum is geared towards projects of fixed teams, with open scope and costs that does not fit well in the reality of the company, but some clients insist on using scrum, their insistence is perhaps for being a buzzword in the market. CMMI's view of the company is that it is about processes and projects in general, but more suited for closed scope projects. Although this company considers that perhaps it will integrate the two methodologies in the future.

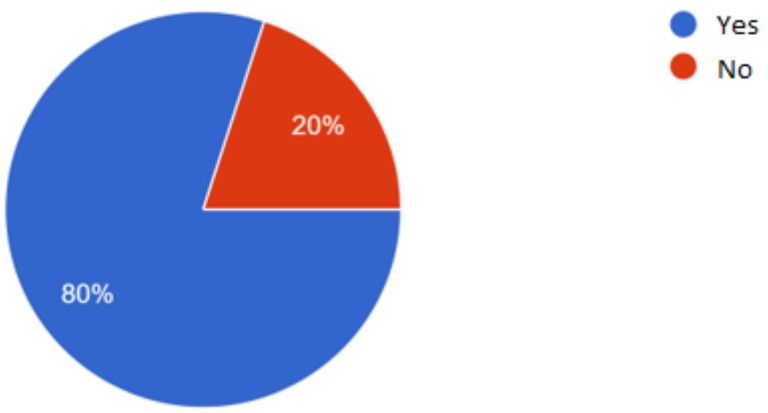


Figure 21: Graphic representing the answers to question Q27- “Do you consider that the objectives of CMMI coexist in harmony with Agile practices?”

The next section of the survey aimed to understand the companies' position on the harmony between the objectives of each CMMI Process Area and agile best practices. For that purpose, a matrix between every process area in CMMI maturity levels and the top 6 Agile best practices found in literature.

Therefore, for each CMMI level procedural area, companies select the option that considered most appropriate or the one that best fits company's reality regarding the integration of the Agile practice with the procedural area in question. Each maturity level process areas were mapped with agile best practices in terms of levels of support. These levels of support were classified as Unsupported, Partially Supported, Supported and Largely Supported.

For CMMI Level 2 exists seven process areas, each one of these areas were mapped with the agile best practices as shown in the following graphs.

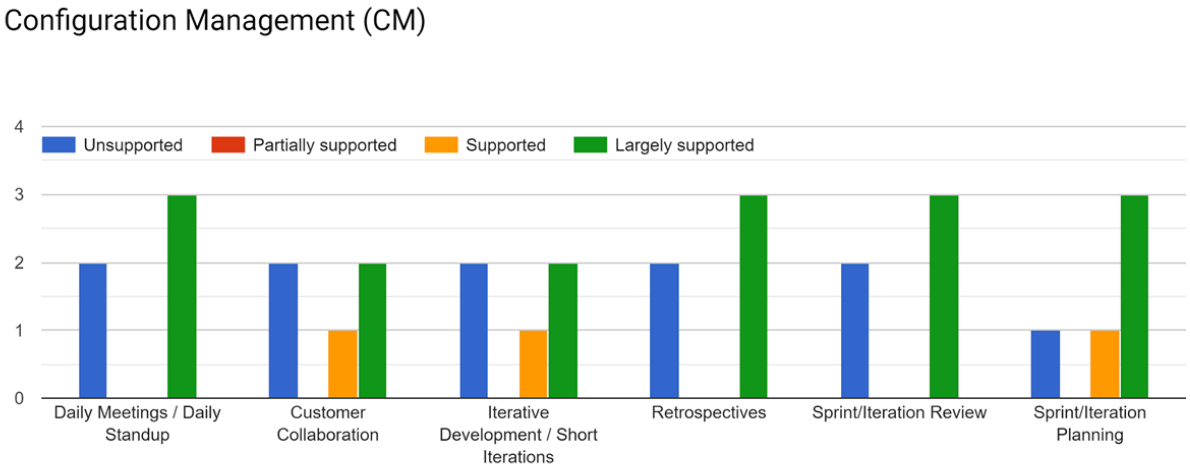


Figure 22: Configuration Management support with Agile best practices

Measurement and Analysis (MA)

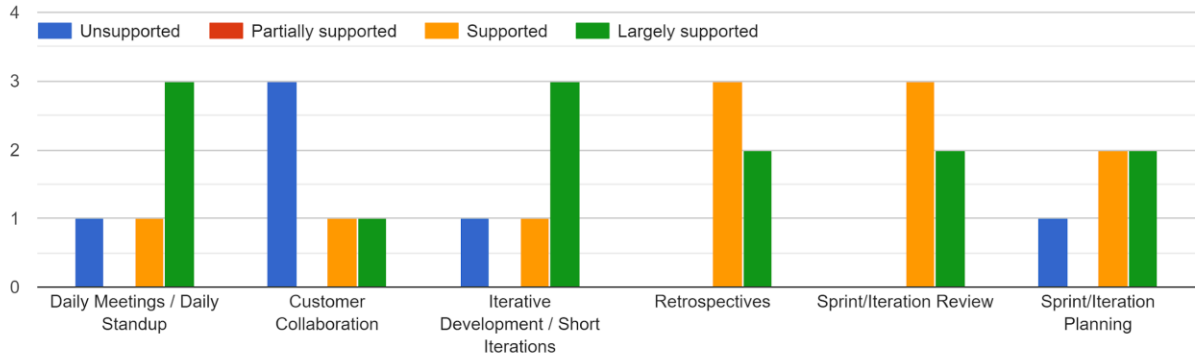


Figure 23: Measurement and Analysis support with Agile best practices

Project Monitoring and Control (PMC)

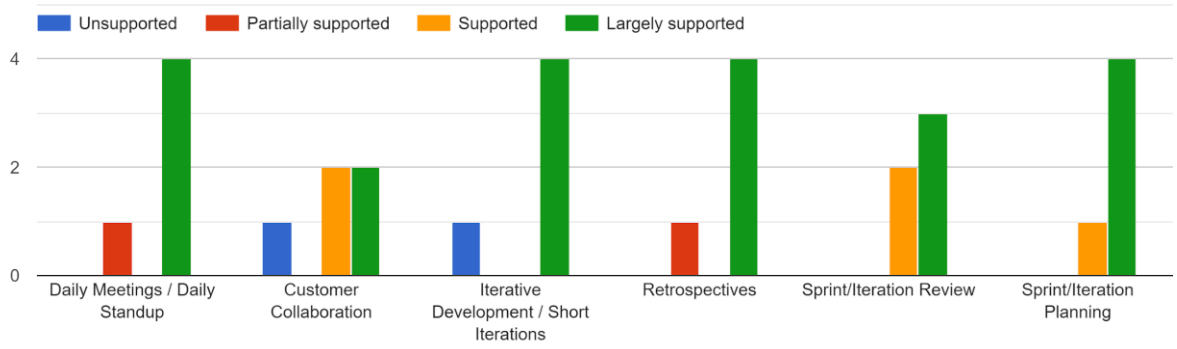


Figure 24: Project Monitoring and Control support with Agile best practices

Project Planning (PP)

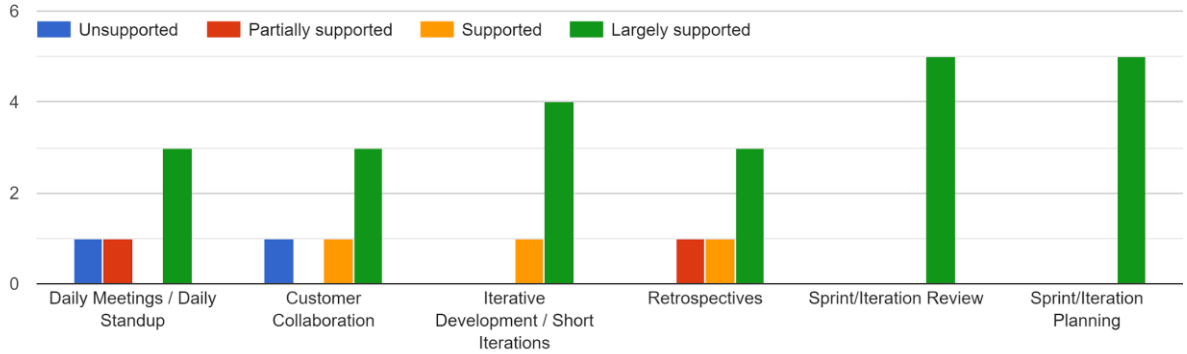


Figure 25: Project Planning support with Agile best practices

Process and Product Quality Assurance (PPQA)

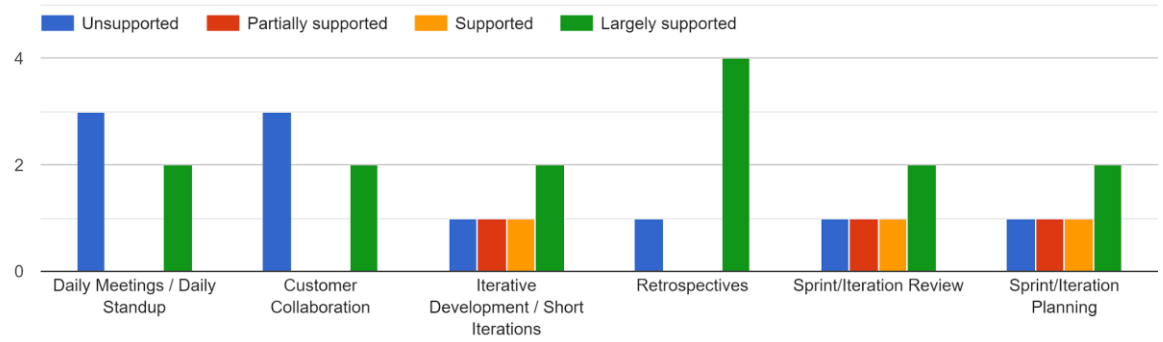


Figure 26: Process and Product Quality Assurance support with Agile best practices

Requirements Management (REQM)

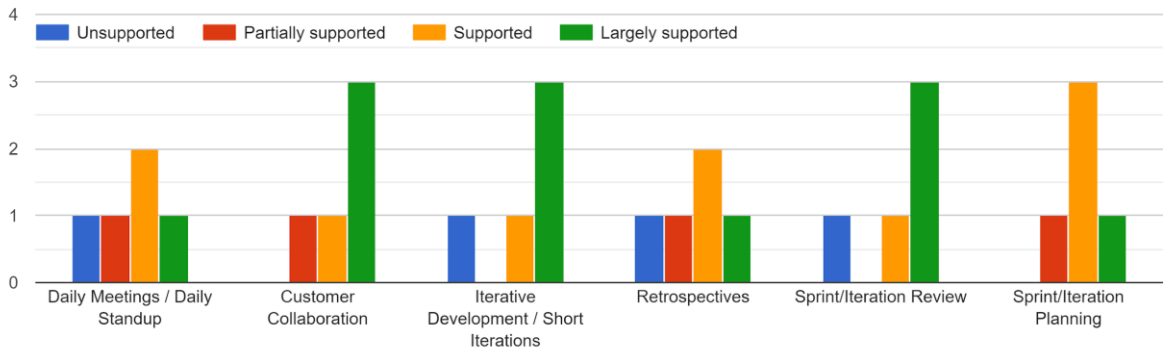


Figure 27: Requirements Management support with Agile best practices

Supplier Agreement Management (SAM)

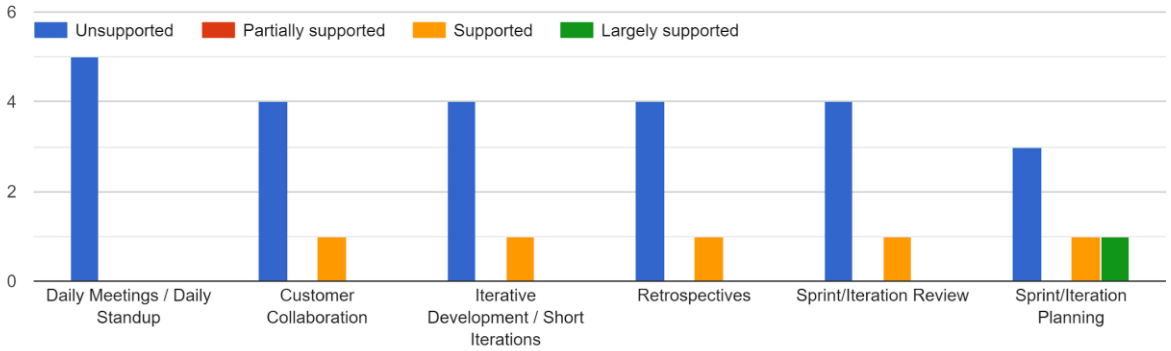


Figure 28: Supplier Agreement Management support with Agile best practices

For CMMI Level 3 exists eleven process areas, each one of these areas were mapped with the agile best practices as shown in the following graphs.

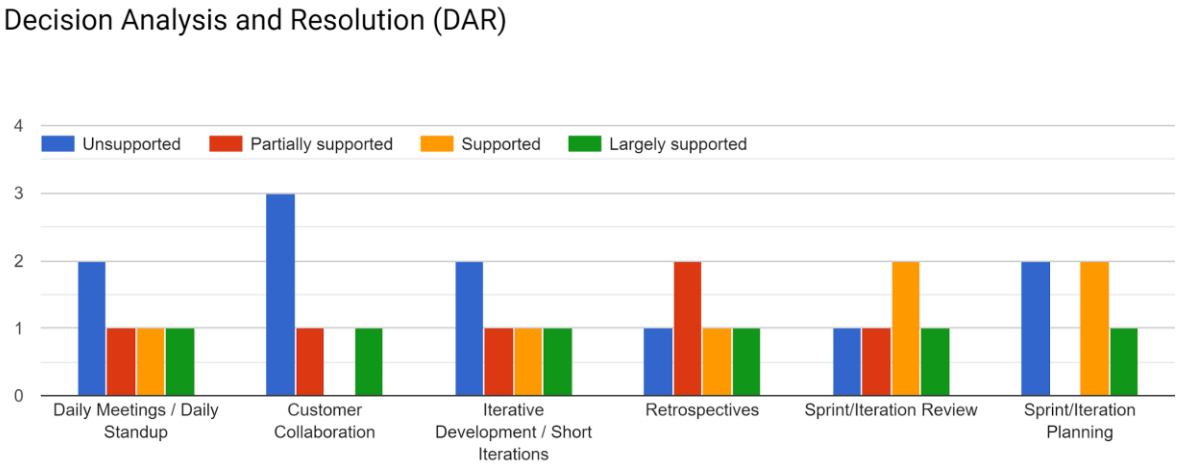


Figure 29: Decision Analysis and Resolution support with Agile best practices

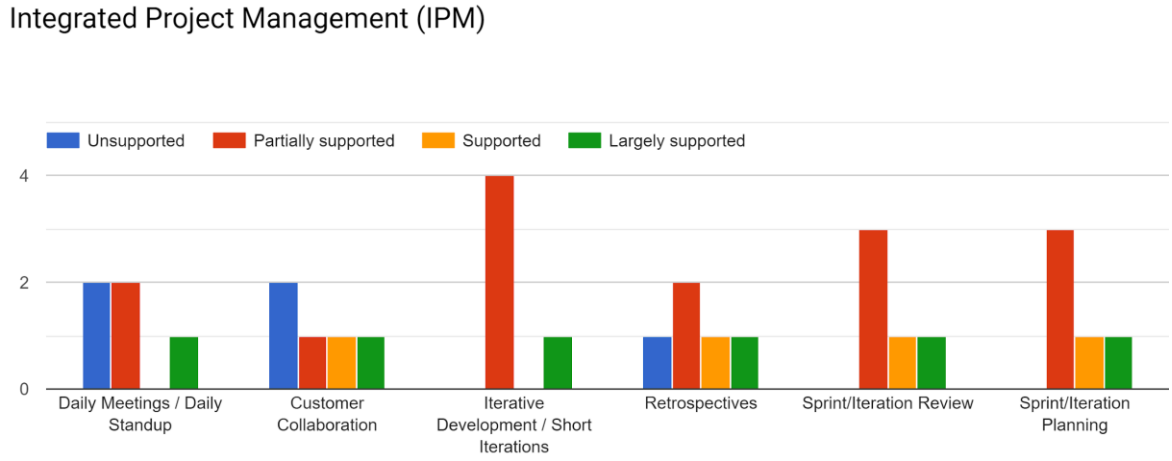


Figure 30: Integrated Project Management support with Agile best practices

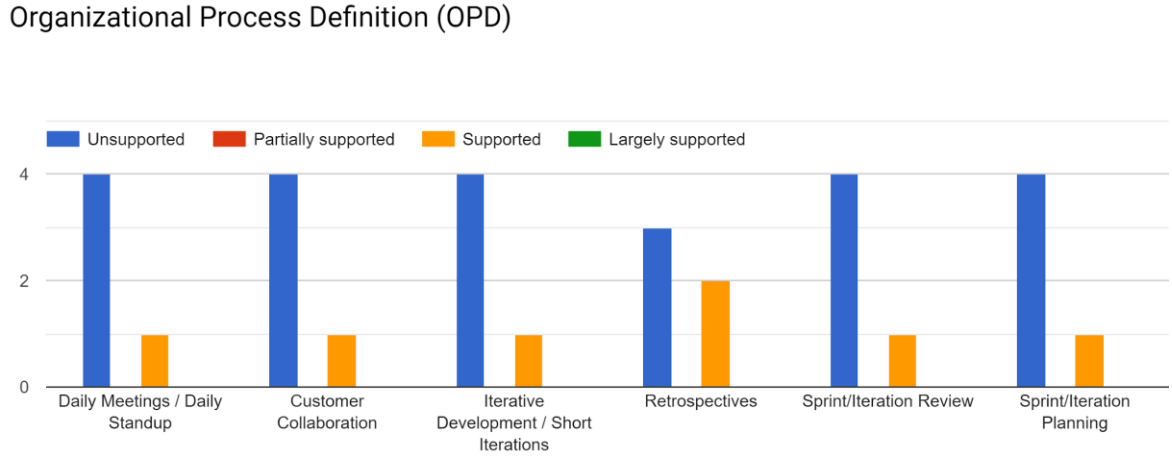


Figure 31: Organizational Process Definition support with Agile best practices

Organizational Process Focus (OPF)

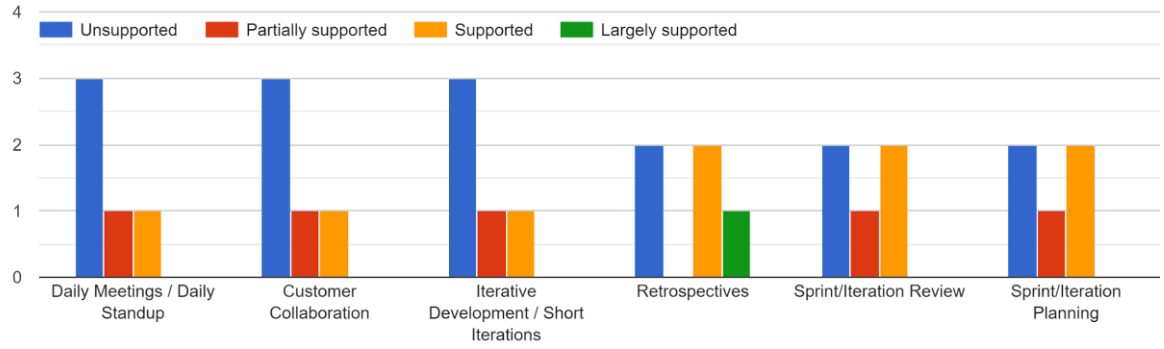


Figure 32: Organizational Process Focus support with Agile best practices

Organizational Training (OT)

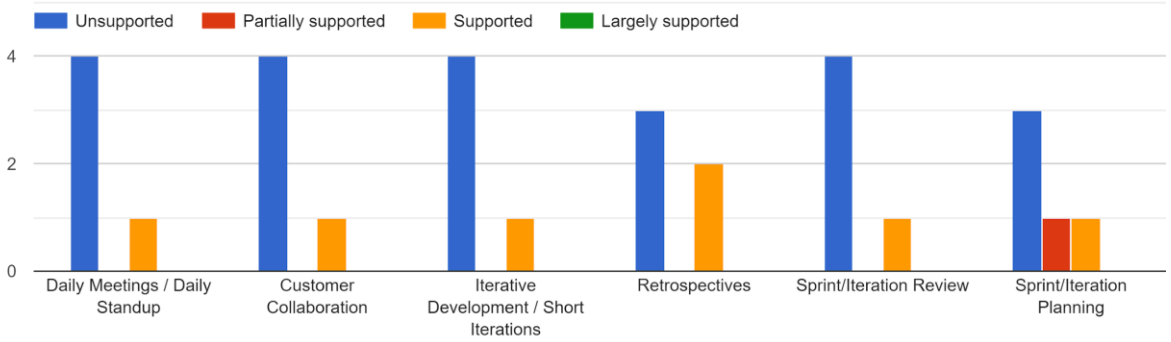


Figure 33: Organizational Training support with Agile best practices

Product Integration (PI)

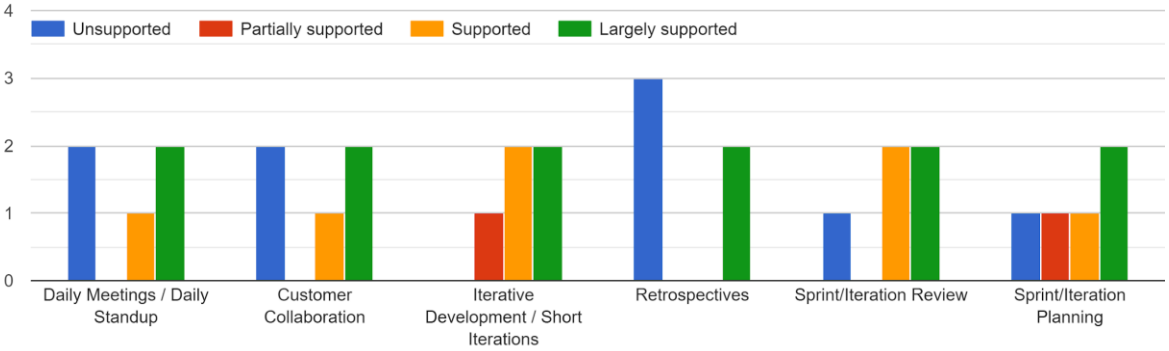


Figure 34: Product Integration support with Agile best practices

Requirements Development (RD)

Figure 35: Requirements Development support with Agile best practices

Risk Management (RSKM)

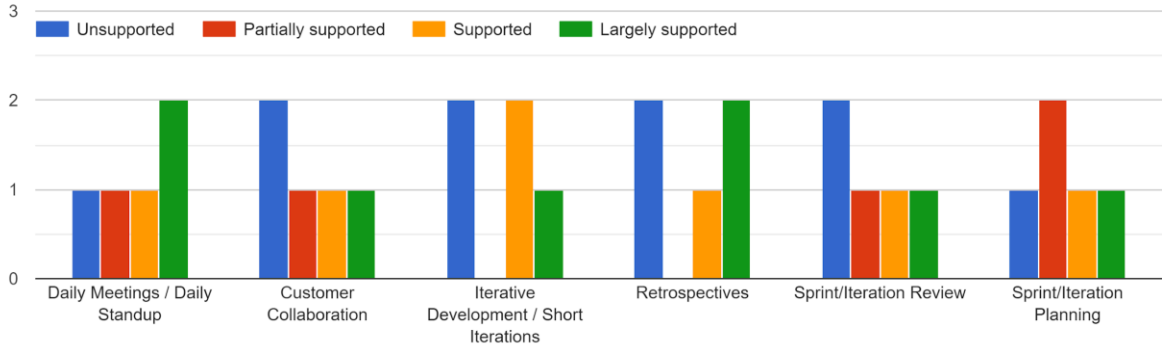


Figure 36: Risk Management support with Agile best practices

Technical Solution (TS)

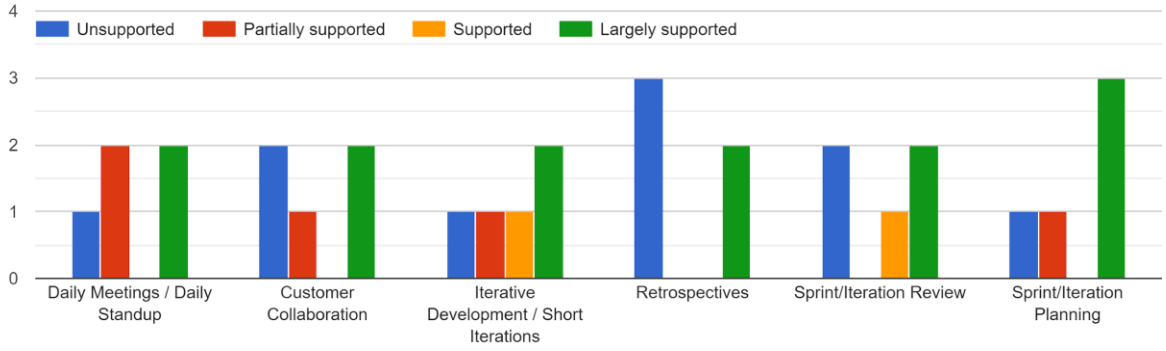


Figure 37: Technical Solution support with Agile best practices

Validation (VAL)

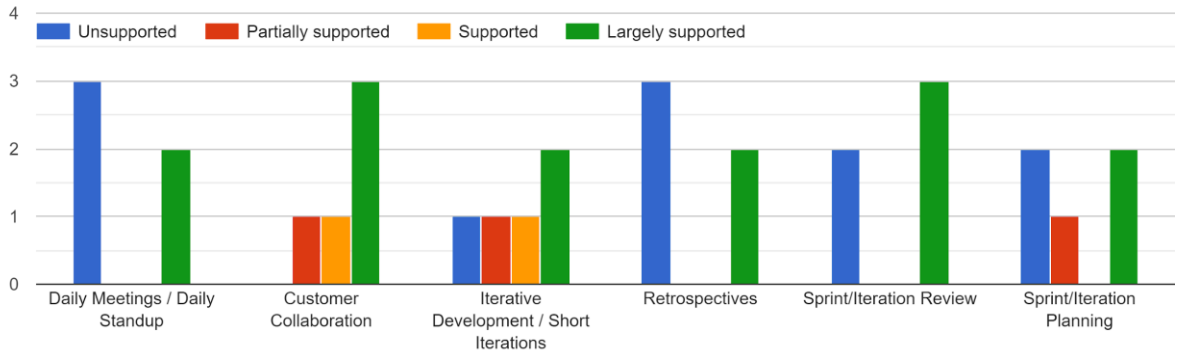


Figure 38: Validation support with Agile best practices

Verification (VER)

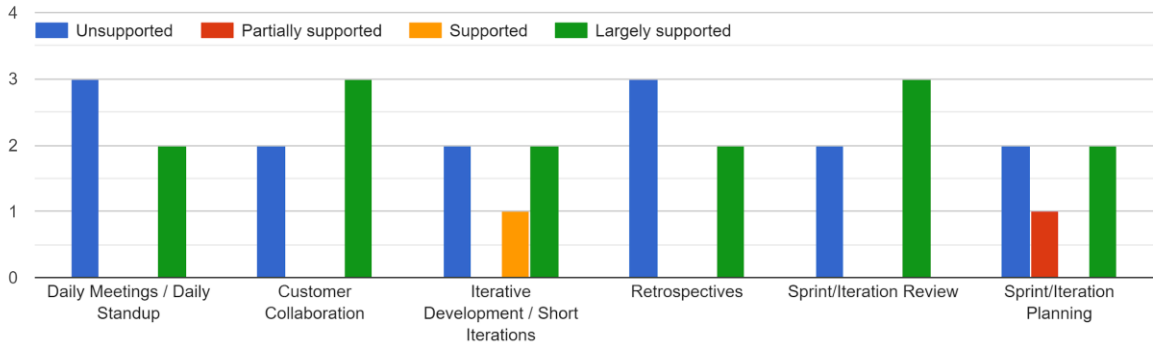


Figure 39: Verification support with Agile best practices

For CMMI Level 4 exists two process areas, each one of these areas were mapped with the agile best practices as shown in the following graphs.

Organizational Performance Management (OPM)

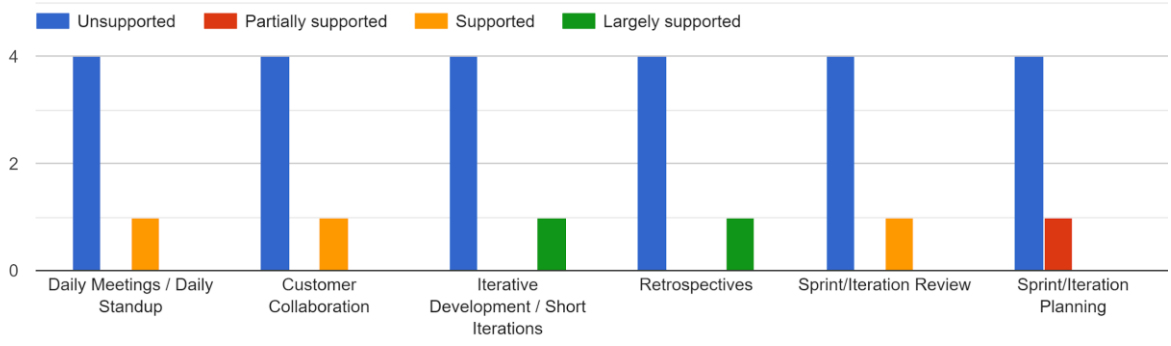


Figure 40: Organizational Performance Management support with Agile best practices

Quantitative Project Management (QPM)

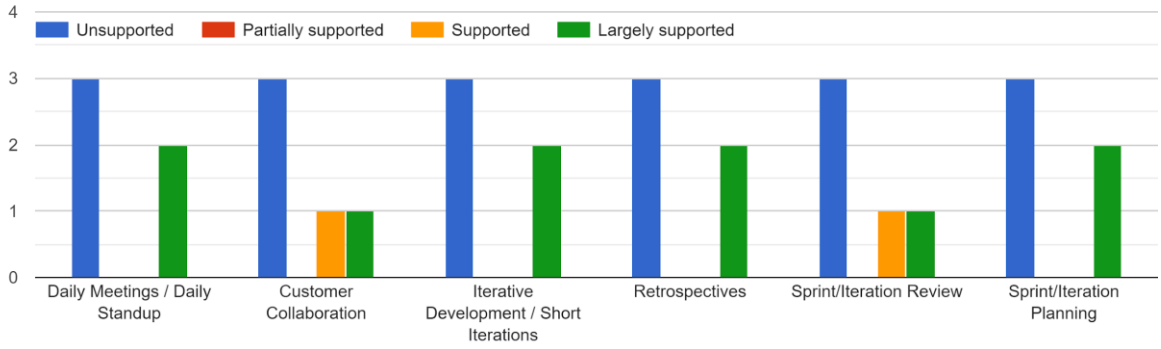


Figure 41: Quantitative Project Management support with Agile best practices

For CMMI Level 5 exists two process areas, each one of these areas were mapped with the agile best practices as shown in the following graphs.

Causal Analysis and Resolution (CAR)

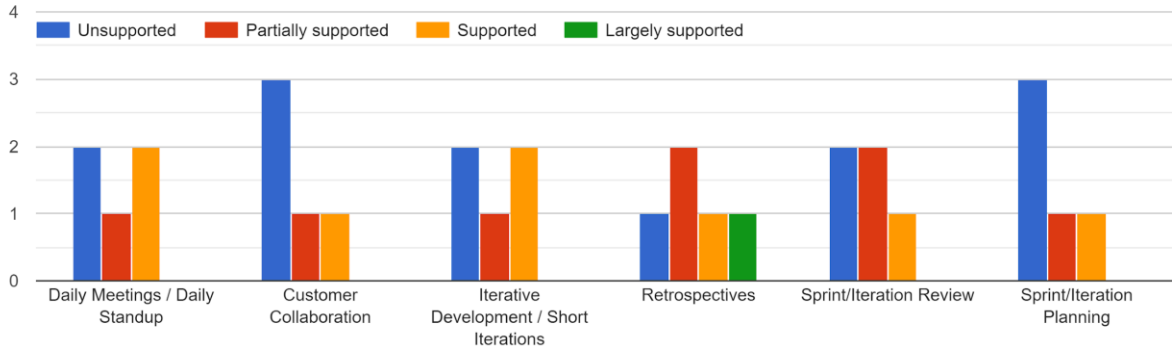


Figure 42: Causal Analysis and Resolution support with Agile best practices

Organizational Process Performance (OPP)

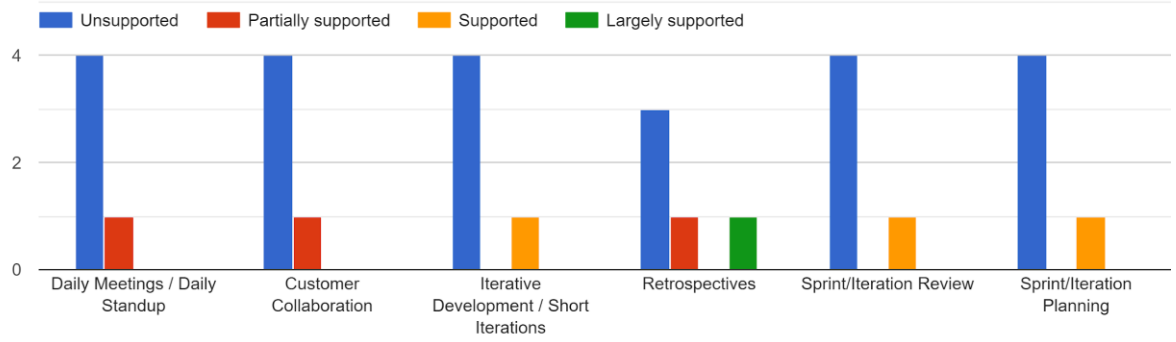


Figure 43: Organizational Process Performance support with Agile best practices

4.2. DISCUSSION

By analyzing the results obtained in the survey some conclusions were taken and are scrutinized in this chapter.

Due to the short sample of companies in Portugal that use CMMI for development and that simultaneously use agile in their projects, we assume that the companies are representative of the level of CMMI in which they are since all the companies must follow the same directives and therefore have similar ways of execution as others of the same CMMI level. For this reason, we can say that companies are representative of the universe of companies that are at the level of CMMI in which they are.

Company 1 and company 4 are early adopters of CMMI. Company 5 is an experience company regarding CMMI model and Agile methodologies.

Among the motivations for choosing to work with CMMI the common denominator between all companies is the capability to improve processes, the great recognition from the market and differentiation of its competitors.

Analyzing the will of companies to achieve higher CMMI level we conclude that, in general, they do not intend to get a higher level of CMMI. And maybe for that reason there is no company with CMMI for development level 4 in Portugal. The level 5 company present in the study continues to consider it important to achieve high levels of CMMI. We can conclude that the will to achieve better levels is not a standard within the CMMI-compliant enterprises of lower levels (Level 2, 3 and 4).

The great advantages of the CMMI identified by the companies are very consistent among them, all the companies consider that the market recognition and the improvement of the processes are the great advantages of the CMMI certification.

The disadvantages also were quite consistent among them all, they identify the extra work that implies following the CMMI model as the major drawback. Apparently by the observed responses there is a greater work at lower CMMI levels since the companies that selected more work as one of the disadvantages of the model were the companies that are in lower CMMI levels. The difficulties of model implementation and the high investment are also presented as disadvantages by three companies spread in all CMMI levels.

CMMI helps in process improvement that outcomes in improvements on final product quality.

That decision could have something to do with the visibility that this certification give to companies, but after that is irrelevant in terms of visibility if higher levels are achieved or not. Companies may think that achieving higher levels is more work with no much return.

From company 5 input we realize that after level 3 there is a greater predictability, less rework, better customer satisfaction and more project control. But for companies that did not achieve those levels that advantages are not that valued.

The agile methodology that is used by all companies is Scrum and the major reason to chose this is because it is the most widely used in the market and the one that brings more advantages.

The higher percentage of companies in the study that believe that CMMI and Agile can coexist simultaneously (80%/4 companies) mean that nowadays companies did not see CMMI and Agile as much as opposites but more like complementary methodologies.

The company that consider that both cannot coexist harmoniously is a company with 1 to 2 years of experience with CMMI level 2 and that does not implement CMMI model for all projects and does not currently integrate these two methodologies, CMMI and Agile. They claim that CMMI is about processes and project in general but more focused on closed scope projects and in another hand Agile is more suitable for projects of fixed teams with open costs and scope.

Each maturity level process areas were mapped with agile best practices in terms of levels of support.

Maturity Level 2 – Managed

Table 5 represents the map between each process area from CMMI level 2 and agile best practices.

Table 5: CMMI Level 2 Process Areas mapped to Agile best practices

Unsupported U | Partially supported PS | Supported S | Largely supported LS

	CM				MA				PMC			
	U	PS	S	LS	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	40%			60%	20%		20%	60%		20%		80%
Customer Collaboration	40%		20%	40%	60%		20%	20%	20%		40%	40%
Iterative Development / Short Iterations	40%		20%	40%	20%		20%	60%	20%			80%
Retrospectives	40%			60%			60%	40%		20%		80%
Sprint / Iteration Review	40%			60%			60%	40%			40%	60%
Sprint / Iteration Planning	20%		20%	60%	20%		40%	40%			20%	80%
	PP				**PPQA**				**REQM**			
	U	PS	S	LS	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	20%	20%		60%	60%			40%	20%	20%	40%	20%
Customer Collaboration	20%		20%	60%	60%			40%		20%	20%	60%

Iterative Development / Short Iterations			20%	80%	20%	20%	20%	40%	20%		20%	60%
Retrospectives		20%	20%	60%	20%			80%	20%	20%	40%	20%
Sprint / Iteration Review				100%	20%	20%	20%	40%	20%		20%	60%
Sprint / Iteration Planning				100%	20%	20%	20%	40%		20%	60%	20%

	SAM			
	U	**PS**	**S**	**LS**
Daily Meetings / Daily Standup	100%			
Customer Collaboration	80%		20%	
Iterative Development / Short Iterations	80%		20%	
Retrospectives	80%		20%	
Sprint / Iteration Review	80%		20%	
Sprint / Iteration Planning	60%		20%	20%

For Configuration Management (CM) the percentage of responses were 53% Largely Supported, 10% Supported and 37% Unsupported.

For Measurement and Analysis (MA) the percentage of responses were 43% Largely Supported, 37% Supported and 20% Unsupported.

For Project Monitoring and Control (PMC) the percentage of responses were 70% Largely Supported, 17% Supported, 7% Partially Supported and 7% Unsupported.

For Process and Product Quality Assurance (PPQA) the percentage of responses were 47% Largely Supported, 10% Supported, 10% Partially Supported and 33% Unsupported.

For Requirements Management (REQM) the percentage of responses were 40% Largely Supported, 33% Supported, 13% Partially Supported and 13% Unsupported.

For Supplier Agreement Management (SAM) the percentage of responses were 3% Largely Supported, 17% Supported, and 80% Unsupported.

For CMMI level 2, in general, the bigger percentage of responses, 48%, stands for Largely Supported. Concluding that CMMI level 2 process areas are Largely Supported with Agile best practices.

However, between Process areas in the same CMMI level, we found different levels of support. The PA Project Planning (PP) is the one that grants higher support followed by Project Monitoring and Control (PMC).

Supplier Agreement Management (SAM) have 80% not supported what means that this process area cannot support on agile practices to fulfill its goals.

Maturity Level 3 – Defined

Table 6 represents the map between each process area from CMMI level 3 and agile best practices.

Table 6: CMMI Level 3 Process Areas mapped to Agile best practices

Unsupported U | Partially supported PS | Supported S | Largely supported LS

	DAR				IPM				OPD			
	U	PS	S	LS	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	40%	20%	20%	20%	40%	40%		20%	80%		20%	
Customer Collaboration	60%	20%		20%	40%	20%	20%	20%	80%		20%	
Iterative Development / Short Iterations	40%	20%	20%	20%		80%		20%	80%		20%	
Retrospectives	20%	40%	20%	20%	20%	40%	20%	20%	60%		40%	
Sprint / Iteration Review	20%	20%	40%	20%		60%	20%	20%	80%		20%	
Sprint / Iteration Planning	40%		40%	20%		60%	20%	20%	80%		20%	
	OPF				**OT**				**PI**			
	U	PS	S	LS	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	60%	20%	20%		80%		20%		40%		20%	40%
Customer Collaboration	60%	20%	20%		80%		20%		40%		20%	40%
Iterative Development / Short Iterations	60%	20%	20%		80%		20%			20%	40%	40%

Retrospectives	40%		40%	20%	60%		40%		80%			20%
Sprint / Iteration Review	40%	20%	40%		80%		20%		20%		40%	40%
Sprint / Iteration Planning	40%	20%	40%		60%	20%	20%		20%	20%	20%	40%
	RD				**RSKM**				**TS**			
	U	**PS**	**S**	**LS**	**U**	**PS**	**S**	**LS**	**U**	**PS**	**S**	**LS**
Daily Meetings / Daily Standup	20%	40%		40%	20%	20%	20%	40%	20%	40%		40%
Customer Collaboration		20%	20%	60%	40%	20%	20%	20%	40%	20%		40%
Iterative Development / Short Iterations		40%	20%	40%	40%		40%	20%	20%	20%	20%	40%
Retrospectives	40%		20%	40%	40%		20%	40%	80%			20%
Sprint / Iteration Review		20%		80%	40%	20%	20%	20%	40%		20%	40%
Sprint / Iteration Planning	20%	20%	20%	40%	20%	40%	20%	20%	20%	20%		60%
	VAL				**VER**							
	U	**PS**	**S**	**LS**	**U**	**PS**	**S**	**LS**				
Daily Meetings / Daily Standup	60%			40%	60%			40%				
Customer Collaboration		20%	20%	60%	40%			60%				
Iterative Development / Short Iterations	20%	20%	20%	40%	40%		20%	40%				
Retrospectives	60%			40%	60%			40%				
Sprint / Iteration Review	40%			60%	40%			60%				
Sprint / Iteration Planning	40%	20%		40%	40%	20%		40%				

In general, for CMMI level 3 process areas the bigger percentage of responses, 56%, stands for Supported.

For CMMI level 3, in general, the bigger percentage of responses, 56%, stands for Supported. Concluding that CMMI level 3 process areas are Supported with Agile best practices.

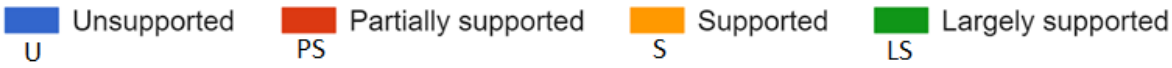
However, between Process areas in the same CMMI level, we found different levels of support. The PA Requirements Development (RD) is the one that grants higher support followed by Validation (VAL) and Verification (VER).

Organizational Process Definition (OPD) have 77% not supported what means that this process area cannot support on agile practices to fulfill its goals.

Maturity Level 4 – Quantitatively Managed

Table 7 represents the map between each process area from CMMI level 2 and agile best practices.

Table 7: CMMI Level 4 Process Areas mapped to Agile best practices



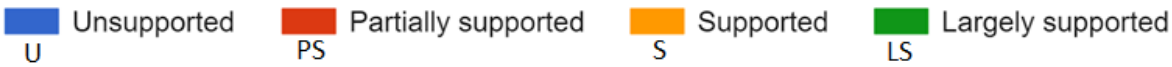
	OPM				QPM			
	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	80%		20%		60%			40%
Customer Collaboration	80%			20%	60%		20%	20%
Iterative Development / Short Iterations	80%		20%		60%			40%
Retrospectives	80%			20%	60%			40%
Sprint / Iteration Review	80%		20%		60%		20%	20%
Sprint / Iteration Planning	80%			20%	60%			40%

In general, for CMMI level 4 process areas the bigger percentage of responses, 70%, stands for Unsupported.

Maturity Level 5 – Optimizing

Table 8 represents the map between each process area from CMMI level 2 and agile best practices.

Table 8: CMMI Level 5 Process Areas mapped to Agile best practices



	CAR				OPP			
	U	PS	S	LS	U	PS	S	LS
Daily Meetings / Daily Standup	40%	20%	40%		80%	20%		
Customer Collaboration	60%	20%	20%		80%	20%		
Iterative Development / Short Iterations	40%	20%	40%		80%		20%	
Retrospectives	20%	40%	20%	20%	60%	20%		20%
Sprint / Iteration Review	40%	40%	20%		80%		20%	
Sprint / Iteration Planning	60%	20%	20%		80%		20%	

In general, for CMMI level 5 process areas, the bigger percentage of responses, 60%, stands for Unsupported.

In general, the Agile best practices that are commonly more supported among the process areas are Daily Meetings / Daily Standup, Retrospectives, Sprint/Iteration Review and Sprint/Iteration Planning.

Table 9 represents the general summary of all Maturity levels support with Agile Best Practices. As we can see higher maturity levels process areas lower integration with agile.

Table 9: Summary of the compatibility between Agile best practices and CMMI Process Areas

Maturity Level	Process Area	Support with Agile Best Practices
Maturity Level 2 Managed	Configuration Management (CM)	Largely Supported
	Measurement and Analysis (MA)	Largely Supported
	Project Monitoring and Control (PMC)	Largely Supported
	Project Planning (PP)	Largely Supported
	Process and Product Quality Assurance (PPQA)	Largely Supported
	Requirements Management (REQM)	Largely Supported
	Supplier Agreement Management (SAM)	Unsupported
Maturity Level 3 Defined	Decision Analysis and Resolution (DAR)	Unsupported
	Integrated Project Management (IPM)	Partially Supported
	Organizational Process Definition (OPD)	Unsupported
	Organizational Process Focus (OPF)	Unsupported

	Organizational Training (OT)	Unsupported
	Product Integration (PI)	Supported
	Requirements Development (RD)	Largely Supported
	Risk Management (RSKM)	Supported
	Technical Solution (TS)	Supported
	Validation (VAL)	Supported
	Verification (VER)	Supported
Maturity Level 4 Quantitatively Managed	Organizational Performance Management (OPM)	Unsupported
	Quantitative Project Management (QPM)	Unsupported
Maturity Level 5 Optimizing	Causal Analysis and Resolution (CAR)	Unsupported
	Organizational Process Performance (OPP)	Unsupported

5. CONCLUSIONS

5.1. SYNTHESIS OF WORK DEVELOPED

The questions that this study intends to answer were identified early in the process of research into this subject and all of them are answered in this study context.

Regarding question 1 (Q1) that question the importance for companies in achieving higher CMMI maturity levels, we can conclude that companies with lower levels of CMMI didn't have that will because their main objective is market recognition and ability to compete for project where CMMI certification is a requirement. Achieving the lower levels of CMMI is enough for achieving that goals and for that reason companies did not see benefits that make it worthwhile to reach higher levels.

However, higher maturity level companies continue to think important to achieve higher levels because of the benefits they think the certification have are much more about process improvements and product quality than recognition. As company 5 alleged, since CMMI Level 3 they identify that there is a greater predictability, less rework, better customer satisfaction and more project control.

Concerning question 2 (Q2), if companies consider compatible and valuable the coexistence between CMMI objectives and Agile best practices, we found out with this study that 80% of companies considered these two methodologies as harmoniously coexisting. That means that many of the myths and fears regarding its coexistence are now dissipated. The only company that did not think these methodologies can coexist and help each other, is a company with CMMI level 2 for 1 to 2 years, and maybe for that reason they did not explore that synergies already.

Question 3 objective is to understand what the points of greater and lesser compatibility between these two methodologies are. We can see that higher levels of CMMI correspond to less compatibility with agile best practices while lower levels represent much greater compatibility with Agile best practices.

Agile methodologies are associated commonly to informal and lightweight documentation that do not emphasize process definition or measurement to the degree that models such as the CMMI do. However, the literature has proved that CMMI model can be applied in a lightweight manner without incurring in excessive documentation.

Companies pursue to be certified in CMMI to get some visibility and differentiate themselves from competitors as well as improve their processes and consequently achieving higher quality in the final product.

The seek for those well-known methodologies as Agile and CMMI is firstly done more to follow market trends than to bring great changes to the organization. But consequently, if those methodologies were understood and followed in the best way, great things can be achieved. The purpose of this study was exactly to write another page in how to take the best advantage of all methodologies and new trends to the benefit of the company.

The major conclusion is that agile methodologies provide many good engineering practices, and together with CMMI, both approaches can achieve very positive synergies.

5.2. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

The major limitation of this work is the fact that the sampling was restrictive because there are not many companies in Portugal that fulfill the needs of this study with respect to the use of CMMI for development and agile methodologies simultaneously. We still manage to find two companies for CMMI maturity level 2 and 3. And one company representative of CMMI maturity level 5. However, there is no company in Portugal with CMMI level 4 and for that reason the survey was not conducted in any Level 4 company.

For future works it is interesting to try to understand exactly how the agile practices help in the achievement of each one of the specific goals for each CMMI maturity level.

BIBLIOGRAPHY

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from http://agilemanifesto.org/

Caralli, R., Knight, M., & Montgomery, A. (2012). Maturity models 101: a primer for applying maturity models to smart grid security, resilience, and interoperability, (November), 1–10. Retrieved from http://resources.sei.cmu.edu/asset_files/WhitePaper/2012_019_001_58920.pdf

Darrin, M. A. G., & Devereux, W. S. (2017). The Agile Manifesto, design thinking and systems engineering. *11th Annual IEEE International Systems Conference, SysCon 2017*. https://doi.org/10.1109/SYSCON.2017.7934765

Gandomani, T. J., & Zulzalil, Hh. (2013). Compatibility of Agile Software Development Methods and CMMI. *Indian Journal of Science and Technology Compatibility*, *6*(8), 5090–5094. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=09746846&AN=90027917&h=9o8cesnkV6MvKX3vZLgPKIss1IPqE12BpV+AY2UlbJWdXr4AyS5WZCFJiSIJsBYsG1bEbh/RSo0+IQ7UpY2c5w==&crl=c

Glazer, H., Boehm, B., & Turner, R. (2010). Love and Marriage: CMMI and Agile Need Each Other.

Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). CMMI ® or Agile : Why Not Embrace Both ! *(CMU/SEI-2008-TN-003). Software Engineering Institut*, (November), 48. https://doi.org/10.1109/AGILE.2006.30

Goksen, Y., Cevik, E., & Avunduk, H. (2015). A Case Analysis on the Focus on the Maturity Models and Information Technologies. *Procedia Economics and Finance*, *19*(15), 208–216. https://doi.org/10.1016/S2212-5671(15)00022-2

ICEAA. (2017). CMMI and Agile Development: A Binary Choice?, (Cmmi).

Institute, C. (2017). What Is CMMI? Retrieved from http://cmmiinstitute.com/capability-maturity-model-integration/

INSTITUTE, C. (n.d.). Published Appraisal Results. Retrieved from https://sas.cmmiinstitute.com/pars/pars.aspx

INSTITUTE, C. (2017). Discover Which CMMI® INSTITUTE MODEL Will Take Your Organization to the Next Level.

Jeff Dalton, Ross Timmerman, Laura Adkins, Kirk Botula, Neil Potter, Dan Torrens, … Margaret Tanner Glover. (2016). A Guide to Scrum and CMMI: Improving Agile Performance with CMMI, 130. Retrieved from http://cmmiinstitute.com/sites/default/files/Scrum_XP_Profile_Final.pdf

Majumdar, A., Ashiqe-Ur-Rouf, M., Islam, M. N., & Arefeen, S. (2011). Capability Maturity Model Integration. *International Journal of Computer and Information Technology*, *2*(1), 68–74. https://doi.org/10.1007/978-3-319-06106-1_13

Matalonga, S. (2012). Agile vs CMMI: Reasons why the vs is sticking for the time being. *Agile Process and Product Management*, *13*(15).

Paper, C., Garbajosa, J., & Calvo-manzano, J. A. (2009). Mapping CMMI Level 2 to Scrum Practices: An Experience Report, *42*(September). https://doi.org/10.1007/978-3-642-04133-4

Paul E. McMahon. (2010). Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement. (SEI Series in Software Engineering, Ed.). Addison Wesley.

Proença, D., & Borbinha, J. (2016). Maturity Models for Information Systems - A State of the Art. *Procedia Computer Science*, *100*(2), 1042–1049. https://doi.org/10.1016/j.procs.2016.09.279

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide™. Retrieved from http://www.scrumguides.org/scrum-guide.html

Scrum.org. (2018). WHAT IS SCRUM? Retrieved from https://www.scrum.org/resources/what-is-scrum?gclid=EAIaIQobChMIsLSqp5uN2QIVwrgbCh2RnA2fEAAYAiAAEgIZUvD_BwE

Selleri Silva, F., Soares, F. S. F., Peres, A. L., Azevedo, I. M. De, Vasconcelos, A. P. L. F., Kamei, F. K., & Meira, S. R. D. L. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, *58*, 20–43. https://doi.org/10.1016/j.infsof.2014.09.012

Soares, F. S. F., & de Lemos Meira, S. R. (2013). An Agile Maturity Model for Software Development Organizations. *ICSEA 2013: The Eighth International Conference on Software Engineering Advances*, (c), 324–328. Retrieved from http://www.thinkmind.org/download.php?articleid=icsea_2013_11_30_10152

Software Engineering Institute. (2010). CMMI for Development, Version 1.3. *Carnegie Mellon University*, (November), 482. https://doi.org/CMU/SEI-2010-TR-033 ESC-TR-2010-033

Souza, T. F. de, & Gomes, C. F. S. (2015). Estudo bibliométrico dos principais modelos de maturidade em gerenciamento de projetos. *Perspectiva Em Gestão E Conhecimento*, *5*(Número Especial), 5–26. Retrieved from http://periodicos.ufpb.br/ojs2/index.php/pgc

Staples, M., & Niazi, M. (2008). Systematic review of organizational motivations for adopting CMM-based SPI. *Information and Software Technology*, *50*(7–8), 605–620. https://doi.org/10.1016/j.infsof.2007.07.003

Sutherland, J., Jakobsen, C. R., & Johnson, K. (2007). Scrum and CMMI level 5: The magic potion for code warriors. *Proceedings - AGILE 2007*, 272–277. https://doi.org/10.1109/AGILE.2007.52

VERSIONONE.COM. (2017). 11th State of Agile Report VERSIONONE.COM #StateOfAgile. *11th State of Agile Report*. https://doi.org/10.1093/jicru/ndl025

VERSIONONE.COM, & COLLAB.NET. (2018). 12th State of Agile Report COLLAB.NET VERSIONONE.COM #StateOfAgile. *12th State of Agile Report*.

Vijayasarathy, L. R., & Butler, C. W. (2016). Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software*, *33*(5), 86–94. https://doi.org/10.1109/MS.2015.26

Webb, C. (2016). Navigating the Agile Landscape. *Blog Deloitte*. Retrieved from http://blog.deloitte.com.au/navigating-the-agile-landscape/

Wells, D. (1999a). The Rules of Extreme Programming. Retrieved from http://www.extremeprogramming.org/rules.html

Wells, D. (1999b). When should Extreme Programming be Used? Retrieved from http://www.extremeprogramming.org/when.html

Young, D. (2010). Software Development Methodologies. *Association of Modern Technologies Proffesionals*, (August 2013). Retrieved from http://www.itinfo.am/eng/software-development-methodologies/

ANNEXES

A1. SURVEY QUESTIONS

Integração CMMI & Agile

O presente questionário está integrado no âmbito da tese de mestrado em Gestão dos Sistemas e Tecnologias de Informação da NOVA Information Management School, da universidade Nova de Lisboa.

O objetivo principal deste estudo é compreender como é que empresas com certificação CMMI-DEV integram esta metodologia com as metodologias ágeis que implementem nos seus projetos.

O tempo de preenchimento deste inquérito é de aproximadamente 30 minutos e toda a informação recolhida será anónima e utilizada exclusivamente no âmbito deste estudo.

Agradeço desde já o tempo dispensado na colaboração com este estudo. Sem a sua ajuda não seria possível!

Mafalda Morais
916258510
mafaldasmorais@gmail.com
SkypeID: anamafaldamorais
LinkedIn: https://www.linkedin.com/in/mafaldamorais/

Para mais informações sobre o mestrado: http://www.novaims.unl.pt/mgi-gsti

*Obrigatório

1. **Endereço de e-mail** *

_____

A Empresa

2. **Qual a dimensão da empresa?** *
   Marcar apenas uma oval.
   - ◯ 0 - 10 funcionários
   - ◯ 10 - 50 funcionários
   - ◯ 50 - 250 funcionários
   - ◯ mais de 250 funcionários

3. **Possui certificação CMMI?** *
   Marcar apenas uma oval.
   - ◯ Sim *Ir para a pergunta 7.*
   - ◯ Não *Ir para a pergunta 3.*

Metodologias

4. **Consideram vir a implementar CMMI no futuro?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

5. **Selecione de entre as seguintes quais consideram vantagens do CMMI** *
Marque todas que se aplicam.

- ☐ Melhoria de processos
- ☐ Maior rigor na execução de tarefas diárias
- ☐ Garantia do cumprimento de prazos e custos
- ☐ Melhoria contínua
- ☐ Reconhecimento do mercado
- ☐ Outro: _____

6. **Selecione de entre as seguintes quais consideram desvantagens do CMMI** *
Marque todas que se aplicam.

- ☐ Dificuldade de implementação
- ☐ Mais trabalho
- ☐ Investimento elevado
- ☐ Rigidez do modelo
- ☐ Outro: _____

7. **Utilizam alguma metodologia ágil?** *
Marcar apenas uma oval.

- ◯ Sim *Ir para a pergunta 21.*
- ◯ Não *Ir para a pergunta 17.*

Metodologias (CMMI)

8. **Qual o nível de maturidade CMMI da sua empresa?** *
Marcar apenas uma oval.

- ◯ Level 2 - Managed
- ◯ Level 3 - Defined
- ◯ Level 4 - Quantitatively Managed
- ◯ Level 5 - Optimizing

9. **Qual é a representação?** *
Marcar apenas uma oval.

- ◯ Staged
- ◯ Continuous

10. **Há quanto tempo obtiveram a certificação CMMI?** *
Marcar apenas uma oval.

- ◯ Menos de 1 ano
- ◯ Entre 1 e 2 anos
- ◯ Entre 2 e 3 anos
- ◯ Entre 3 e 4 anos
- ◯ Entre 4 e 5 anos
- ◯ Mais de 5 anos

11. **O que motivou o interesse em obter esta certificação?** *

_____

_____

_____

_____

_____

12. **Consideram importante atingir níveis mais elevados de CMMI (Se Aplicável)?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não
- ◯ N/A

13. **Selecione de entre as seguintes quais consideram vantagens do CMMI** *
Marque todas que se aplicam.

- ☐ Melhoria de processos
- ☐ Maior rigor na execução de tarefas diárias
- ☐ Garantia do cumprimento de prazos e custos
- ☐ Melhoria contínua
- ☐ Reconhecimento do mercado
- ☐ Outro: _____

14. **Selecione de entre as seguintes quais consideram desvantagens do CMMI** *
Marque todas que se aplicam.

- ☐ Dificuldade de implementação
- ☐ Mais trabalho
- ☐ Investimento elevado
- ☐ Rigidez do modelo
- ☐ Outro: _____

24. **Qual(ais) metodologias ágeis costumam utilizar?** *
Marque todas que se aplicam.

- [ ] Adaptive software development (ASD)
- [ ] Crystal
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Extreme Programming (XP)
- [ ] Feature-Driven Development (FDD)
- [ ] Kanban
- [ ] Lean Software Development
- [ ] Rapid application development (RAD)
- [ ] Scrum
- [ ] Outro: _____

25. **Quais foram as razões para escolha dessa(s) metodologia(s)?** *
Marque todas que se aplicam.

- [ ] A mais usada no mercado
- [ ] Conhecida de alguém na organização
- [ ] Recomendada por parceiro ou cliente
- [ ] Mais vantagens
- [ ] Fácil Implementação
- [ ] Outro: _____

26. **Já utilizavam Agile antes de obterem a certificação CMMI?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

27. **Consideram que os objectivos do CMMI convivem em harmonia com as práticas Agile?** *
Marcar apenas uma oval.

- ◯ Sim *Ir para a pergunta 29.*
- ◯ Não *Ir para a pergunta 27.*

CMMI & Agile Não Integrados

28. **Porque motivo não integram as duas metodologias no mesmo projeto?** *

_____

_____

_____

_____

_____

15. **O que mudou desde a introdução deste modelo na empresa?** *

16. **Todos os projetos seguem o modelo CMMI?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

17. **Utilizam alguma metodologia ágil?** *
Marcar apenas uma oval.

- ◯ Sim *Ir para a pergunta 23.*
- ◯ Não *Ir para a pergunta 17.*

Agile

18. **Consideram usar metodologias ágeis em futuros projetos?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não
- ◯ Talvez

19. **Justifique a resposta anterior** *

20. **Consideram que os objectivos do CMMI convivem em harmonia com as práticas Agile?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não
- ◯ Talvez

21. **Justifique a resposta anterior** *

Pare de preencher este formulário.

Agile

22. **Qual(ais) metodologias ágeis costumam utilizar?** *
Marque todas que se aplicam.

- [ ] Adaptive software development (ASD)
- [ ] Crystal
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Extreme Programming (XP)
- [ ] Feature-Driven Development (FDD)
- [ ] Kanban
- [ ] Lean Software Development
- [ ] Rapid application development (RAD)
- [ ] Scrum
- [ ] Outro:

23. **Quais foram as razões para escolha dessa(s) metodologia(s)?** *
Marque todas que se aplicam.

- [ ] A mais usada no mercado
- [ ] Conhecida de alguém na organização
- [ ] Recomendada por parceiro ou cliente
- [ ] Mais vantagens
- [ ] Fácil Implementação
- [ ] Outro:

Pare de preencher este formulário.

Agile & CMMI

24. **Qual(ais) metodologias ágeis costumam utilizar?** *
Marque todas que se aplicam.

- [ ] Adaptive software development (ASD)
- [ ] Crystal
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Extreme Programming (XP)
- [ ] Feature-Driven Development (FDD)
- [ ] Kanban
- [ ] Lean Software Development
- [ ] Rapid application development (RAD)
- [ ] Scrum
- [ ] Outro: _____

25. **Quais foram as razões para escolha dessa(s) metodologia(s)?** *
Marque todas que se aplicam.

- [ ] A mais usada no mercado
- [ ] Conhecida de alguém na organização
- [ ] Recomendada por parceiro ou cliente
- [ ] Mais vantagens
- [ ] Fácil Implementação
- [ ] Outro: _____

26. **Já utilizavam Agile antes de obterem a certificação CMMI?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

27. **Consideram que os objectivos do CMMI convivem em harmonia com as práticas Agile?** *
Marcar apenas uma oval.

- ◯ Sim *Ir para a pergunta 29.*
- ◯ Não *Ir para a pergunta 27.*

CMMI & Agile Não Integrados

28. **Porque motivo não integram as duas metodologias no mesmo projeto?** *

_____

_____

_____

_____

_____

29. **Consideram integrar as duas metodologias no futuro?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não
- ◯ Talvez

Ir para a pergunta 29.

CMMI Process Areas & Agile Best Practices - Maturity Level 2 - Managed

Pretendemos fazer um mapeamento relativamente à compatibilidade entre as áreas processuais do CMMI com as práticas mais usadas no Agile.

Assim sendo para cada área processual do nível 2 de CMMI assinale a opção que considere mais adequada ou a que mais se adequa à realidade da sua empresa relativamente à integração da prática Agile com a área processual em questão.

30. **Configuration Management (CM)** *
The purpose of CM is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup	◯	◯	◯	◯
Customer Collaboration	◯	◯	◯	◯
Iterative Development / Short Iterations	◯	◯	◯	◯
Retrospectives	◯	◯	◯	◯
Sprint/Iteration Review	◯	◯	◯	◯
Sprint/Iteration Planning	◯	◯	◯	◯

31. **Measurement and Analysis (MA)** *
The purpose of Measurement and Analysis is to develop and sustain a measurement capability used to support management information needs.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup	◯	◯	◯	◯
Customer Collaboration	◯	◯	◯	◯
Iterative Development / Short Iterations	◯	◯	◯	◯
Retrospectives	◯	◯	◯	◯
Sprint/Iteration Review	◯	◯	◯	◯
Sprint/Iteration Planning	◯	◯	◯	◯

32. **Project Monitoring and Control (PMC)** *

The purpose of PMC is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

33. **Project Planning (PP)** *

The purpose of PP is to establish and maintain plans that define project activities.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

34. **Process and Product Quality Assurance (PPQA)** *

The purpose of PPQA is to provide staff and management with objective insight into processes and associated work products.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

38. **Integrated Project Management (IPM)** *

The purpose of IPM is to establish and manage the project and the involvement of relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

39. **Organizational Process Definition (OPD)** *

The purpose of OPD is to establish and maintain a usable set of organizational process assets, work environment standards, and rules and guidelines for teams.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

40. **Organizational Process Focus (OPF)** *

The purpose of OPF is to plan, implement, and deploy organizational process improvements based on a thorough understanding of current strengths and weaknesses of the organization's processes and process assets.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

35. **Requirements Management (REQM)** *

The purpose of REQM is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

36. **Supplier Agreement Management (SAM)** *

The purpose of SAM is to manage the acquisition of products and services from suppliers.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

CMMI Process Areas & Agile Best Practices - Maturity Level 3 - Defined

Pretendemos fazer um mapeamento relativamente à compatibilidade entre as áreas processuais do CMMI com as práticas mais usadas no Agile.

Assim sendo para cada área processual do nível 3 de CMMI assinale a opção que considere mais adequada ou a que mais se adequa à realidade da sua empresa relativamente à integração da prática Agile com a área processual em questão.

37. **Decision Analysis and Resolution (DAR)** *

The purpose of DAR is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

47. **Verification (VER)** *

The purpose of VER is to ensure that selected work products meet their specified requirements.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

CMMI Process Areas & Agile Best Practices - Maturity Level 4 - Quantitatively Managed

Pretendemos fazer um mapeamento relativamente à compatibilidade entre as áreas processuais do CMMI com as práticas mais usadas no Agile.

Assim sendo para cada área processual do nível 4 de CMMI assinale a opção que considere mais adequada ou a que mais se adequa à realidade da sua empresa relativamente à integração da prática Agile com a área processual em questão.

48. **Organizational Performance Management (OPM)** *

The purpose of OPM is to proactively manage the organization's performance to meet its business objectives.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

49. **Quantitative Project Management (QPM)** *

The purpose of QPM is to quantitatively manage the project to achieve the project's established quality and process performance objectives.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

41. **Organizational Training (OT)** *

The purpose of OT is to develop skills and knowledge of people so they can perform their roles effectively and efficiently.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

42. **Product Integration (PI)** *

The purpose of PI is to assemble the product from the product components, ensure that the product, as integrated, behaves properly (i.e., possesses the required functionality and quality attributes), and deliver the product.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

43. **Requirements Development (RD)** *

The purpose of RD is to elicit, analyze, and establish customer, product, and product component requirements.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

44. **Risk Management (RSKM)** *

The purpose of RSKM is to identify potential problems before they occur so that risk handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

45. **Technical Solution (TS)** *

The purpose of TS is to select, design, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product related lifecycle processes either singly or in combination as appropriate.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

46. **Validation (VAL)** *

The purpose of VAL is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

CMMI Process Areas & Agile Best Practices - Maturity Level 5 - Optimizing

Pretendemos fazer um mapeamento relativamente à compatibilidade entre as áreas processuais do CMMI com as práticas mais usadas no Agile.

Assim sendo para cada área processual do nível 5 de CMMI assinale a opção que considere mais adequada ou a que mais se adequa à realidade da sua empresa relativamente à integração da prática Agile com a área processual em questão.

50. **Causal Analysis and Resolution (CAR)** *

The purpose of CAR is to identify causes of selected outcomes and act to improve process performance.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

51. **Organizational Process Performance (OPP)** *

The purpose of OPP is to establish and maintain a quantitative understanding of the performance of selected processes in the organization's set of standard processes in support of achieving quality and process performance objectives, and to provide process performance data, baselines, and models to quantitatively manage the organization's projects.
Marcar apenas uma oval por linha.

	Unsupported	Partially supported	Supported	Largely supported
Daily Meetings / Daily Standup				
Customer Collaboration				
Iterative Development / Short Iterations				
Retrospectives				
Sprint/Iteration Review				
Sprint/Iteration Planning				

Uma cópia das suas respostas será enviada para o endereço de e-mail fornecido