Universidade do Minho
Escola de Engenharia

Carolina Lomar da Costa Gonçalves

Implementation of good engineering practices based on a maturity model in agile development contexts

Fevereiro de 2022

Universidade do Minho
Escola de Engenharia

Carolina Lomar da Costa Gonçalves

Implementation of good engineering practices based on a maturity model in agile development contexts

Dissertação de Mestrado
Mestrado em Gestão de Projetos de Engenharia

Trabalho efetuado sob a orientação de:
Professora Doutora Anabela Pereira Tereso
Doutor João Manuel Peixoto Faria

Fevereiro de 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Acknowledgements

I would like to take the opportunity to express my sincere gratitude to the people that supported me, in various ways, during this dissertation and my master's degree.

Firstly, my gratitude for all the guidance, skills, expertise, willingness to offer useful suggestions, and availability to my supervisor Anabela Tereso. Second, to my advisor João Faria, my acknowledgements for his know-how and availability.

To my supervisor at Bosch, André Leite Ferreira, for all the knowledge, guidance, and experience shared with me, for letting me be part of a great team, and always making sure I do my best, a special thanks to you.

My gratefulness to the teams at Bosch that I was proud to being part of, both my main team and internship team, for the friendship, knowledge, and assistance provided. Also, a thanks to my other colleagues at Bosch, that helped me to get more knowledge about my role through their own experience.

Last but not least, I could not forget to be grateful for all the support by my family and friends for all the help, patience, and motivation during this phase.

Thank you all for encouraging me to give everything I could to accomplish another step of my life.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

Implementação de boas práticas de engenharia baseadas num modelo de maturidade em contextos de desenvolvimento ágil

Devido à nova realidade existente em relação ao mercado tecnológico e às mudanças céleres por parte dos requisitos dos clientes, as empresas vêm-se obrigadas a acompanhar esta evolução para conseguirem competir com outras organizações e garantirem a sua presença no mercado. Estes últimos anos demonstram o quanto a sociedade está conectada a nível mundial, o que faz com que as mudanças sejam mais rápidas e imprevisíveis. Este novo mundo é designado por VUCA *(Volatile*, *Uncertain*, Complex e *Ambiguous)* e apresenta vários desafios significativos para indivíduos, equipas e organizações de vários setores que podem ser afetados pelo mesmo. Em resposta às alterações repentinas no mercado de desenvolvimento de software, apareceram as metodologias ágeis, permitindo corresponder à rápida mudança do mercado, devido à sua flexibilidade, planeamento adaptativo e rapidez.

A presente dissertação foi desenvolvida no departamento de engenharia e inovação da empresa *Bosch Car Multimedia*, em Braga, onde as metodologias ágeis, Scrum e *Scaled Agile Framework* (SAFe), já se encontram implementadas na equipa e no projeto. Deste modo, o ambiente empresarial estava preparado para que fossem implementadas boas práticas de engenharia baseadas num modelo de maturidade em contextos de desenvolvimento ágil. O principal objetivo desta investigação consiste em perceber se existe alguma forma de aprimorar a adoção de um modelo de maturidade, neste caso o *Capability Maturity Model Integration* (CMMI), por meio de uma metodologia ágil, Scrum, no contexto de uma equipa de desenvolvimento de software.

Este projeto foi desenvolvido sob a aplicação de uma iteração da metodologia de investigação *Action-Research*, onde o investigador foi integrado na equipa com o papel de *Scrum Master*. Iniciou-se por efetuar uma avaliação do ambiente inicial da equipa, o que permitiu determinar quais as práticas existentes que deveriam ser mantidas, quais deveriam ser aperfeiçoadas e quais deveriam ser criadas de modo a conseguir implementar as boas práticas de engenharia com base no CMMI através do Scrum.

O principal contributo desta dissertação consistiu na aquisição de um maior conhecimento fundamentado sobre modelos de maturidade em contextos ágeis em uma equipa de software, do qual é possível dizer que a literatura existente ainda se encontra em um desenvolvimento inicial sobre o tópico.

Palavras-chave

Agile, CMMI, Gestão de Projetos, Scrum

ABSTRACT

Implementation of good engineering practices based on a maturity model in agile development contexts

Due to the new reality that exists in relation to the technological market and the fast changes in customer requirements, companies are forced to follow this evolution in order to be able to compete with other organizations and ensure their presence in the market. The last few years demonstrate how society is connected worldwide, which makes changes faster and more unpredictable. This new world is called VUCA (Volatile, Uncertain, Complex and Ambiguous) and presents a number of significant challenges for individuals, teams and organizations across industries that may be affected by it. In response to the unexpected changes in the software development market, agile methodologies emerged, which allow to respond to the rapid change in the market, due to their flexibility, adaptive planning, and velocity.

This dissertation was developed in the engineering and innovation department of the Bosch Car Multimedia company, in Braga, where agile methodologies, Scrum and Scaled Agile Framework (SAFe), are already implemented in the team and in the project where this study was carried out. As a result, the business environment was prepared for the deployment of good engineering practices through agile contexts based on a maturity model. The main objective of this research is to understand if there is any way to improve the adoption of a maturity model, in this case the Capability Maturity Model Integration (CMMI), through an agile methodology, Scrum, in the context of a software development team.

This project was developed under the application of an iteration of the Action-Research methodology, where the researcher was integrated in the team with the role of Scrum Master. It started by carrying out an evaluation of the team's initial environment, which allowed to determine which existing practices should be maintained, which should be improved, and which should be added in order to implement good engineering practices based on CMMI through Scrum.

The main contribution of this dissertation consisted in the acquisition of more grounded knowledge about maturity models in agile contexts in a software team, from which it is possible to conclude that the existing literature is still in an early development stage on the topic.

KEYWORDS

Agile, CMMI, Project Management, Scrum Framework

List of Content

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations and Acronyms

CM	Configuration Management
CMMI	Capability Maturity Model Integration
Daily Scrum	DS
EST	Estimating
II	Implementation Infrastructure
MC	Monitor and Control
N.A.	Not Applicable
PA	Practice Area
PAs	Practice Areas
PI	Program Increment
PLAN	Planning
PM	Project Management
PMI	Project Management Institute
PO	Product Owner
PR	Peer Reviews
RDM	Requirements Development and Management
SAFe	Scaled Agile Framework
SM	Scrum Master
TS	Technical Solution
US	User Story

1. INTRODUCTION

This chapter provides the context for the theme, concepts and the current situation of the project that will be addressed throughout the present dissertation. It will begin with the explanation of the background of this research. Additionally, the organization where this research project took place will be presented. Next, the research goals and objectives, motivation and challenges of this dissertation are introduced and the research methodology that was adopted will be explained. At the end, the structure of the document is outlined.

1.1 Background

Over the last thirty years, Project Management (PM) has significantly increased its visibility (Fernandes et al., 2013). During this period society changed rapidly, becoming more deeply connected, evolving into a Volatile, Uncertain, Complex and Ambiguous (VUCA) society, which created a significant number of challenges for individuals, teams and organizations (Bennett & Lemoine, 2014). Aiming to manage business objectives, organizations have been using project management practices, which has been making businesses project oriented. Project Management has emerged as the most common method for integrating organizational activities and motivating groups to attain better levels of productivity and performance (Fernandes et al., 2013). The PM intends to bring an environment where the people incorporated in the project work together to achieve specific objectives (Seymour & Hussein, 2014).

There are currently two primary methodologies for managing software development projects: waterfall and agile. Both offer advantages and disadvantages for stakeholders, depending on the features of the project and the parties involved. The waterfall approach assumes a sequential flow, with the client providing his needs and the product being produced around them. Agile methodologies are better suited for projects where requirements may change fast since they are more adaptable to customer needs and allow clients to examine several iterations of the product (Dima & Maassen, 2018).

Agile methods, such as Scrum, may be used by small companies all the way up to international corporations (Vlaanderen et al., 2011). One of the most popular agile methodologies is Scrum. It gives standard procedures that can be used with a range of techniques or processes. To be able to use Scrum, various adjustments may be required, such as: organization and management format of labour; individuals mindset; client interactions; and office layout (Rola et al., 2016).

The baseline for Scrum is a self-organizing and cross-functional team, which means there is no team leader setting tasks or a way to solve a problem, and therefore all members have the needed abilities to generate value during the Sprints (Schwaber & Sutherland, 2020), which are the minimal iteration cycle for the project execution. Teams must have a shared goal, confidence, respect, teamwork, a swift decision-making mechanism, and the capacity to work with uncertainty in order to be agile (Cockburn & Highsmith, 2001).

1.2 Scrum and CMMI

According to Schwaber and Sutherland (2020), Scrum is built around three pillars: transparency, inspection, and adaptation. Trust must be established on Scrum's pillars, but the existence of transparency, inspection, and adaptation is dependent on the inclusion of five key values, namely commitment, focus, openness, respect, and courage. Notwithstanding, Scrum teams are constantly on the lookout for new opportunities and ideas. They must cultivate and improve the performance of these values in their tasks, performance, and behaviour, in order for the Scrum process to thrive.

Scrum practices include three roles, five events, and three artifacts. To maximize value, have an effective process and obtain all of its advantages, the Scrum team must be aware of and comprehend the eleven elements. The Scrum team is composed of three roles: Product Owner, Scrum Master, and Developers. The events are named as Sprint Planning, Sprint, Daily Scrum, Sprint Review, and Sprint Retrospective. The artifacts are designated by Product Backlog, Sprint Backlog, and Increment (Schwaber & Sutherland, 2020). Scrum helps to ease and simplify developers work by parcelling out the workload into small parts that are performed during sprints. Daily meetings are used to verify progress, adjustments, and develop the product incrementally (Hidalgo, 2019).

As previously mentioned, software development is a fast-paced environment with rapidly changing objectives and needs, requiring the use of agile methodologies. In addition to agile methods, a proper level of maturity is essential to deal with the velocity of change (Rainho & Barreiros, 2019).

Capability Maturity Model Integration (CMMI) is one of the process capability maturity models. It is considered a software process improvement structure for describing and evaluating processes and practices that software-development companies may employ (Staples et al., 2007).

In line with the Software Engineering Institute, the CMMI consists in a process improvement model that collects best practices, assists in the organization and improvement of company activities and in the coordination of multidisciplinary activities, and allows to align the business objectives with the objectives of the process improvement. As a result, its goal is to aid in the improvement of company processes and the capacity to manage the development, purchase, and maintenance of a product or service. In this way, CMMI enables the discovery of ways that assist the organization in evaluating organizational maturity or process capability, which will lead the organization to identify and implement improvement priorities (SEI, 2010).

In the most recent version, 2.0, CMMI has four categories, named, Doing, Managing, Enabling, and Improving. Each category has subcategories that have 10 sub-categories, in total, called Capability Areas. Furthermore, the Capability Areas have a total of 25 Practice Areas, that are the subdivisions of the Capability Areas. Therefore, the CMMI Practice Areas have 6 levels that explain how well the company is implementing each Practice Area, that range from level 0 to 5, where the 0 is the lowest level and 5 is the highest (ISACA-CMMI Institute, 2020).

People believe that CMMI and agile methodologies cannot work together. However, SEI released a paper claiming that the CMMI model and agile methods can coexist, although this has yet to be shown empirically (Pikkarainen, 2009).

1.3 Company Presentation

The Bosch Group, Figure 1, was founded in 1886 by Robert Bosch. It is present in about 60 countries through 440 subsidiaries and regional companies, being a world-leading technology and service supplier, and its major business sectors are mobility solutions, industrial technology, consumer goods, and energy and building technology (Robert Bosch GmbH, 2021).

Figure 1 – Bosch Group Logo.
(Robert Bosch GmbH, 2021)

In Portugal, there are four Bosch Units, namely the Robert Bosch S.A., located in Lisbon, as the commercial headquarters for the Portuguese market, responsible for the central accounting, marketing, training services, and shared human resources and communication services for the entire world; the Bosch Thermotechnology, placed in Aveiro, providing hot water solutions through water heaters; the Bosch Security Systems, located in Ovar, focused on producing security and communications systems, fire alarms, electronic displays and other solutions for other business units in the Bosch Group; and the Bosch Car Multimedia, in Braga, responsible for the automotive electronics division by developing and producing multimedia solutions and automotive sensors and the mobility area with cross-domain computing solutions, chassis systems and automotive aftermarket (Robert Bosch S.A., 2021a).

The Bosch Car Multimedia S.A. is the unit where this dissertation was carried out. It has 3,280 employees, where more than 330 of them are engineers and it is considered one of the most significant employers in Braga area (Robert Bosch S.A., 2021b).

1.4 Research Goals

The motive for creating this project stemmed from the fact that while applying agile methods, good engineering practices are not the main focus. Consequently, this research sought to find a harmonisation model that allowed to verify the relation between the implementation of the Scrum methodology and the good engineering practices defined in 8 of the 25 practice areas up to level 2 of the CMMI model, with the purpose of, not only getting the benefits from both methodologies, but also to improve the team's performance.

To achieve the purpose of this dissertation, a chance to study and answer a research question arose, which served as a foundation for solid, well-established, and effective goals. The research question that is proposed is the following:

Is there any way to enhance the adoption of the CMMI model through agile methodologies in the context of a software development team?

After analysing and evaluating the research question, to define how it can contribute to empirical knowledge, the following goals (G) were considered for this research:

G1 – The development of a control method for implementing good engineering practices in agile and CMMI Level 2 environments;

G2 – The introduction of a team control method – A way to understand if the control method is being fulfilled;

G3 – The identification of synergies between maturity models, CMMI until level 2, and agile methodology, Scrum, for software development teams in the automotive sector.

This dissertation was made during a curricular internship at Bosch Car Multimedia Portugal. This research project was applied in a software team, as their Scrum Master, in order to investigate, throughout observations and interviews, their maturity and main challenges, and also, how agile assisted their work, and to propose improvements using CMMI and Scrum. In discussion with the company representative, it was decided that the level of CMMI to take into consideration was level 2. Therefore, creating a team control method to help them improve their work and performance. And, lastly, identifying the synergies between CMMI and Scrum through a harmonisation method.

1.5 Research Methodology

The research methodology consists of a process that has detailed procedures or techniques, used to detect, select, process, and assess information with respect to a specific theme. The methodology section is the foundation of how the research should be carried out and provides the necessary information to evaluate the validity and reliability of the investigation (Saunders et al., 2019). This methodology topic seeks to help to answer to two questions: *"How was the data collected or generated?"* and *"How was it examined?"*. To have the knowledge about the research methods it is useful to understand how to do research and which tools to use to answer the research question. In order to have an initial approach, the research onion presented by Saunders et al. (2019) helps to bring forward the strategy, method, philosophy, approach, time horizon, techniques and procedures selected.

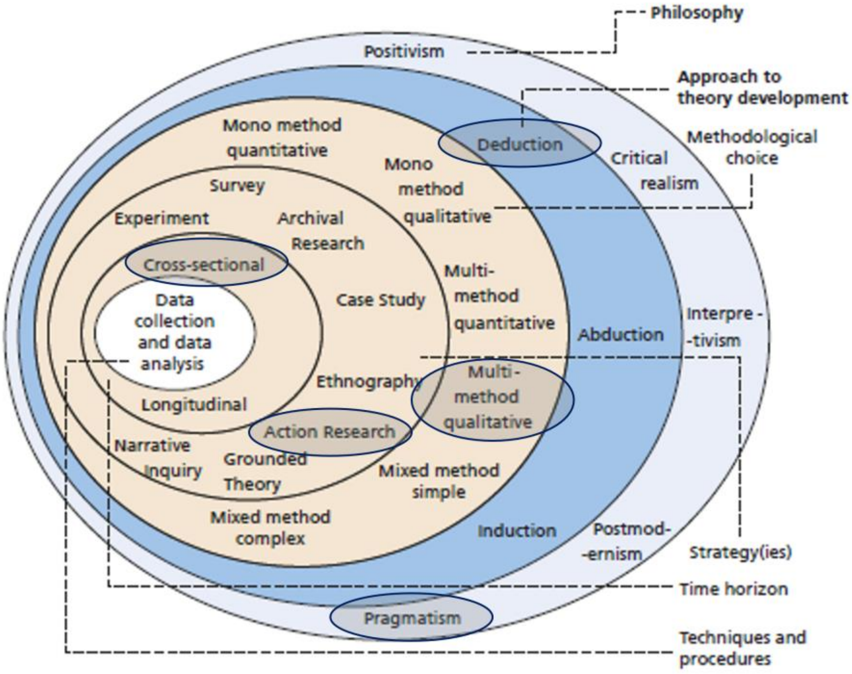


Figure 2 – The *"Research Onion" with the researcher's choices.*
Adapted from Saunders et al. (2019)

The Figure 2 shows the options selected for this dissertation. The research onion is a means of representing data techniques and methods in order to describe the research methodology used (Saunders et al., 2019). The support of the research onion enables the explanation, description, and justification of the possibilities that may be employed to execute the present project. Due to the short time frame during which the research was conducted, the cross-sectional time horizon was used for this study. In terms of strategy, action research was chosen to be used in this dissertation since it is tied to a change process, including interactions, and is based on a real-life scenario. This strategy was chosen in order to meet the goals outlined in the preceding section 1.4 Research Goals. Because the study examined work practices and teams, multi-method qualitative was the best methodological choice for this scientific study. Methods such as observation, interviews, and documental analysis were employed. Given that, this research is based on scientific research, the approach to theory development selected is deduction, which moves from a global into a specific situation to verify the theory's accuracy. The philosophy underlying this research is pragmatism, since the research question should dictate the research philosophy to be used, which is the case, and it also placed a focus on findings and practical solutions (Saunders et al., 2019).

1.6 Structure of the Dissertation

The present dissertation is structured in six chapters.

The first chapter, Introduction, presents the background about the theme, company, research goals, and methodology.

In Chapter 2, Literature Review, the main topics will be described deeply. It begins with the subject Project Management, where Traditional Project Management and Agile Project Management are presented. Then Scrum and SAFe are explained, followed by CMMI and its integration with Agile Methodologies.

The third chapter, designated Methodology: Action Research, incorporates the action research explained for this case, which means that it will be clarified how the action research was implemented and show the adjustments done to fit in this study through its phases. Furthermore, it will be exposing the techniques and procedures for the data collection and analysis.

In chapter 4, Case Presentation, the context in relation to the project, the team and the researcher intervention will be presented. Also, it will be shown the diagnosis phase of the case, made by the researcher, through observation and interviews.

As for chapter 5, Methods and Results, it will expose the subchapter of quality control method developed for the team, focusing on its creation, implementation, and results. Afterwards, the harmonisation method between agile and CMMI will be presented, showing the creation of a method between an agile methodology, Scrum, working together with a maturity model, CMMI.

Last of all, in chapter 6, Conclusions and Future Work, it will be presented the contributions of this research, its limitations and the future work that can be done.

2. LITERATURE REVIEW

This chapter aims to provide the content that will help to understand the main subjects that will be mentioned during the present dissertation. First, to understand better the topics and how it began, the initial area to introduce will be Project Management. Second, after being acquainted with this area, it is important to mention the Agile Project Management which creates a great opportunity to bring the Scrum and SAFe Frameworks. Third, to mention the maturity model of this dissertation, it will be referenced the CMMI. Finally, it is necessary to understand how Agile and CMMI perform together.

2.1 Project Management

Project Management seeks to create an environment so that all the people incorporated in a project work together to achieve a specific objective, thus being possible to deliver a successful project. Even though there are many complexities and uncertainties in a project, history shows many challenging projects that were completed, before the existence of an institute, books, and guides, such as the Pyramids of Giza, the Great Wall of China, and the Coliseum. Despite non-existent documentation of the methods and techniques used, in the 1950s, enterprises started to implement some tools and techniques on their projects. In the 1970s, the technological advance created an opportunity for software organizations to develop project management software (Seymour & Hussein, 2014). In recent years, research activities in relation to projects and project management have been growing due to the increasing complexity and dynamism in the business environment (Bergmann & Karwowski, 2019).

According to previous research, project management standards, tools, techniques, and practices are becoming more widely adopted among several companies and industries. In order to accomplish strategy implementation, business transformation, continuous improvement, and new product creation, the use of project, program, and portfolio management is currently an approach in several organizations (Seymour & Hussein, 2014; Winter et al., 2006).

The Project Management Institute says that project management involves the application of knowledge, skills, tools and techniques in project activities in order to fulfil the requirements and execute the projects efficiently and effectively. The success of a project can be evaluated through time, resources, objectives, and uncertainty (Project Management Institute, 2017). Software Project Management intends to implement techniques for creating, updating or solving a software problem that satisfies the needs of stakeholders. It emphasizes a balanced view of theory and practice, technology and people, customer value and profit, and strategies and tactics (Royce, 1998).

2.1.1 Traditional Project Management

Traditional project management is based on a strict view of human nature, society and human perceptions, knowledge and action, as if humans were completely predictable beings, therefore follows techniques and tools that consist in a sequence of steps. It is an approach adopted for several decades, aimed at plans with little flexibility for change and has a top-down leadership style, that is, it is based on command, control and hierarchy (Bergmann & Karwowski, 2019).

The most common traditional project management method is called the waterfall model. The waterfall model consists in a plan-driven process. It divides the software development process into phases, consisting of a process that is guided by a plan, thence prior to beginning software development, it is necessary to plan and arrange all the process actions first. The steps of the waterfall model correspond to the key software development activities. Prior to beginning software development, the plan and schedule of all process steps must be done (Sommerville, 2016). Figure 3 shows, in a general way, how the waterfall model works.

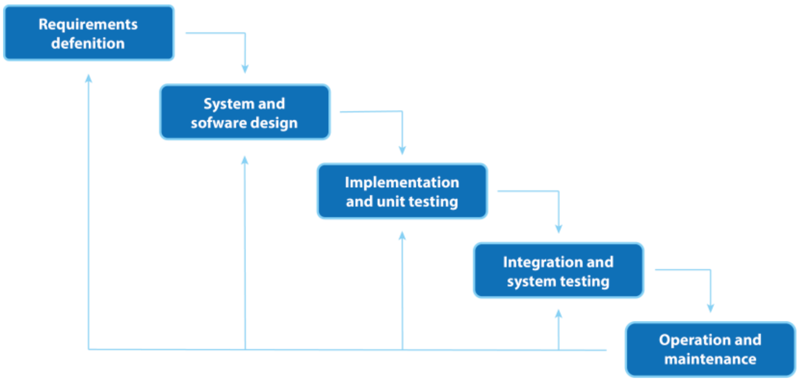

Requirements defenition
System and sofware design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 3 – Waterfall Model.
Adapted from Sommerville (2016)

The first stage, Requirements Analysis and Definition, consists of consulting system users to define system specifications, such as details of services, restrictions and goals. The second one, System and Software Design, intends to acquire a system architecture by identifying which requirements are satisfied by hardware and software. After identifying the requirements that are satisfiable by software, the system and software relationships are described. Implementation and Unit Testing is the third stage. It intends to perform the set or units of programs and verify that the specifications are met. The next stage is the Integration and System Testing, which involves integrating and testing individual program units or programs to confirm that the requirements have been satisfied. After this, the software system is given to the customer once it has been thoroughly tested. The last one, Operation and Maintenance, is the longest phase of the cycle because it contains the installation, the use, and the maintenance of the system. Therefore, in this phase, improvements and new requirements that are identified during the implementation and use of the system will be implemented (Sommerville, 2016).

Due to the dynamic environment, changing requirements and technology in recent decades, traditional project management began to reach its limitations and in response, a new approach to match the set of changes was created, agile project management (Bergmann & Karwowski, 2019).

2.1.2 Agile Project Management

Nowadays, it is necessary for organizations to adapt faster to the digital age, in order to guarantee their existence in the market. Business agility has become imperative. There is a dependence on the ability to rapidly produce new, innovative, and high-quality products and services (Knaster & Leffingwell, 2020).

The Agile Manifesto was founded by a team of software professionals. *"Individuals and interactions over processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation. Responding to change over following a plan"* are the given value-based foundations of these specialists in order to achieve successful software development (Beck et al., 2001). Agile practices and principles can be used regardless of project uncertainties, complexity, volatility and risk (Bergmann & Karwowski, 2019). The agile techniques are attractive and feasible choices for companies. They enable organizations to achieve quality, project budget management, aligned with the organization's business plan, and to deliver value often and constantly, related to the current problems of software development (Campanelli & Parreiras, 2015). Although agile project management has a special focus on software development, it is important to emphasize that it has high potential and can be used in other types of projects (Bergmann & Karwowski, 2019).

Beck et al. (2001) states the 12 agile principles to follow: 1. Focus on satisfying the customer through timely delivery of useful software; 2. Welcome change to give the customer competitive advantage; 3. Deliver working software in short timescales; 4. Business people and developers must work together daily; 5. Give more responsibility to motivated people and the tools to accomplish their goals; 6. Dialogues are the best way to communicate; 7. The main measure of progress is the software being usable; 8. Sustainable development, allowing for the developers to not be overworked; 9. Attention to a high quality software increases agility; 10. The work must be simplified in order to not waste effort and time; 11. Self-organising teams provide better results; and 12. The team must reflect regularly on how to improve and adapt to it.

Some of the most common agile methods are Scrum, Extreme Programming (XP), Kanban, Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), and Crystal. Currently, the Scrum Framework is considered the most adopted agile methodology (Campanelli & Parreiras, 2015). When there are many teams implementing Scrum and working simultaneously in the same project, large-scale agile methodologies were created to respond to this necessity (Layton & Ostermiller, 2017). Therefore, Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS), Nexus, Disciplined Agile Delivery (DaD), and Scrum of Scrums (SoS) are some examples of those methodologies.

2.2 Scrum Framework

Agile methods, such as Scrum, may be used by small companies all the way up to international corporations (Vlaanderen et al., 2011). One of the most popular agile methodologies is Scrum. It gives general procedures that may be used with a variety of techniques or processes (Rola et al., 2016). It is an agile framework that helps in the creation of innovative products and services (Rubin, 2012). Scrum assists organizations to develop adaptive solutions for complex problems in order to generate the highest possible value (Schwaber & Sutherland, 2020). Scrum is not a standardized methodical process that tells managers exactly what to do in order to create a high quality product, it is, instead, a tool to help manage work (Rubin, 2012, p. 13). In line with Ken Schwaber (1997), it is a test oriented and incremental approach to deliver a product or improvement on a product that aims to maximize predictability and control risk. Scrum's baseline is a self-organizing and cross-functional team, that has no team leader settling tasks or finding a solution to a problem, and therefore all members have the abilities required to provide value throughout Sprints (Schwaber & Sutherland, 2020).

Furthermore, Scrum is centred on three main pillars, namely, transparency, inspection, and adaptation. The first one enables stakeholder's connection, either customer, CEO, and individual contributors. It allows everyone to access to the work that is being executed. The second consists of regular examination of the product, processes, practices, and progress, with the intention of creating an opportunity to maximize value. The third and last one seeks continuous improvement (Schwaber & Sutherland, 2020).

According to Schwaber and Sutherland (2020), trust must be built upon Scrum's pillars. However, the occurrence of transparency, inspection and adaptation is dependent on the inclusion of five fundamental values, namely commitment, focus, openness, respect, and courage. Firstly, the team should perform the assignments as a unit, meaning that all members trust each other, know, and believe that they are able to complete the tasks allocated to them. Secondly, focus relates to team members' ability to finish every task they begin, so consideration should be given to the amount of work in progress. Thirdly, it is important to consider members' openness, that is, their willingness to speak in case they need help, and to being open about their hurdles and labour. Fourthly, individuals are expected to respect everyone who is inside the process, their ideas, tough days, and achievements. Lastly, the value of courage, which means individuals should feel at ease to say no, to ask and to attempt new things. Notwithstanding, Scrum teams are constantly on the lookout for new opportunities and ideas. They must cultivate and improve the performance of these values in their tasks, performance, and behaviour in order for the Scrum process to thrive (Schwaber & Sutherland, 2020).

To be able to use Scrum, various adjustments may be required, such as: organization and management format of labour; individuals mindset; client interactions; and office layout (Rola et al., 2016). In order for this empirical framework to work, it is needed to combine the Scrum team with the Scrum cycle which means with specific artifacts and events.

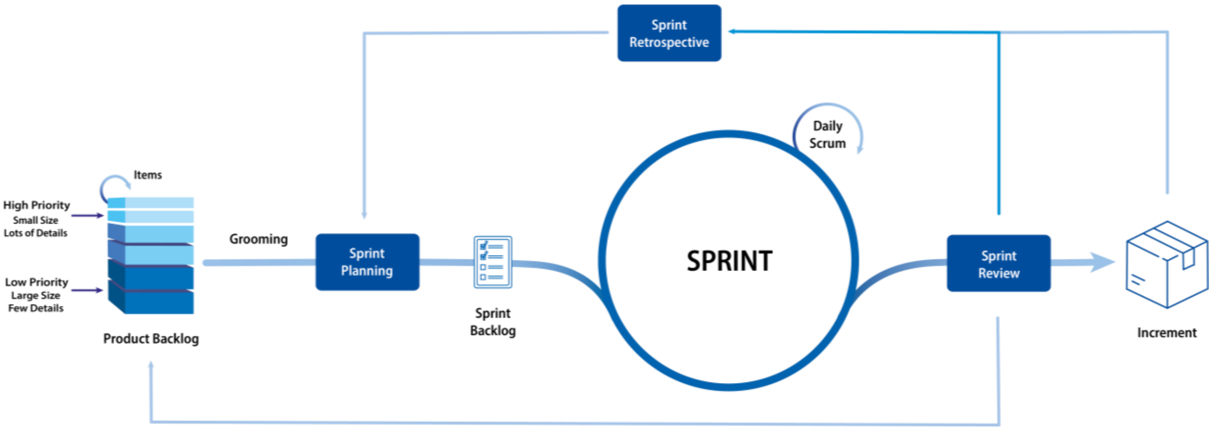


Figure 4 – Scrum Lifecycle.
Adapted from Scrum.org (2021) and Rubin (2013)

The Scrum practices are composed of three roles, five events and three artifacts. The Scrum team must grasp the eleven elements in order to optimize value, have a successful process, and acquire all of the advantages. The roles are Product Owner (PO), Scrum Master (SM) and Developers. The events are, namely, Sprint Planning, Sprint, Daily Scrum, Sprint Review, and Sprint Retrospective. The artifacts are designated by Product Backlog, Sprint Backlog, and Increment (Schwaber & Sutherland, 2020).

2.2.1 Roles of a Scrum Team

The Scrum team is composed of PO, SM and Developers. Since to have a large team can bring issues such as development complexity, or a small team can bring problems of skill lack, the fitting number for the Development Team is seven elements (Lei et al., 2017). Table 1 shows the members of a scrum team and the description of their role.

Table 1 – Scrum Team roles and description.
Adapted from Schwaber and Sutherland (2020)

Role	Description

Product Owner	› Explains the Product Backlog items;
	› Assigns the items' priority;
	› Ensures transparency, visibility and understanding;
	› Maximizes the value of the product.

Scrum Master	› Ensures that the Scrum Framework is being accomplished; › Accountable for the Scrum Team's effectiveness; › Helps the team focus; › Removes obstacles/impediments.
Developers	› Execute the tasks; › Create a plan for the Sprint (Sprint Backlog); › Adapt the plan each day towards the Sprint Goal; › Are accountable as professionals.

In relation to the PO, it is the person who has the contact with the client and brings to Developers the product requirements. This means that the PO explains and manages the Product Backlog, its items and goals, and makes sure the team understands all in order to develop the product at the highest level (Lei et al., 2017). According to Schwaber and Sutherland (2020), the PO is: responsible for creating and clearly communicating the Product Backlog items; responsible for designing and specifically communicate the Product Goal, guaranteeing that the Product Backlog is clear, accessible, and understandable; and acquiring Product Backlog items.

The SM helps the team and organization to understand the theory and practices of Scrum. Additionally, the SM is responsible for assisting the PO, and developing the team and organization. It is responsible for the effectiveness of the Scrum team, since the SM has the role to remove barriers (Schwaber & Sutherland, 2020).

In respect of the development team, this is the one who delivers Increments during Sprints. The Developers create a plan of the Sprint, define the Sprint Backlog, ensure quality via a Definition of Done, and adjust their work to reach the Sprint Goal (Schwaber & Sutherland, 2020).

2.2.2 Events of the Scrum Cycle

A Sprint contains the events, which are a chance to check and develop Scrum artifacts. They are designed to improve transparency, create regularity, and minimize the need of undefined meetings and should be held at the same time. If events are not correctly operated, it is a lost opportunity to evolve

(Schwaber & Sutherland, 2020). Table 2 presents the events of the Scrum methodology and their description.

Table 2 – Scrum events description.
Adapted from Schwaber and Sutherland (2020) and Rad and Turley (2013)

Events	Description
Sprint Planning	› Why is this Sprint valuable? › What can be done in this Sprint? › How will the chosen work get done? › Planning Poker.
Sprint	› Where the ideas become value; › Work needed to achieve the product goal.
Daily Scrum	› Inspect progress toward the Sprint Goal; › Adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.
Sprint Review	› Inspect the outcome of the Sprint; › Determine future adaptations.
Sprint Retrospective	› Plan ways to increase quality and effectiveness for process, people, relations and tools.
Product Backlog Grooming	› Reviews and revises Product Backlog Items; › Provides more and detailed information and estimations to user stories.

In the Sprint Planning the PO and Developers define the goals for a certain sprint by analysing the items that come from the Product Backlog. The Developers generate a goal commitment and a time to develop it during the Sprint Planning, in order to deliver it at the end of the sprint (Rubin, 2012; Schwaber & Sutherland, 2020). User stories are concise and straightforward explanations of functionality used to specify system behaviour. They are designed to individually enable the incremental development through the construction of a small part of the system behaviour. They provide simple information to allow business and technical professionals to comprehend the goal(s), only mentioning the details when the story is ready to be implemented. User stories offer features directly to the end user and get increasingly precise when the acceptance criteria are applied (Leffingwell et al., 2021c).

The Sprint is *"the heartbeat of Scrum"*. This means it is where the work planned and needed at the sprint planning is performed and accomplished at the highest quality. It is from here that the output, in this case the increment, is born. It has a start and an end date and lasts between one and four weeks. The sprint goals cannot change during the sprint (Rubin, 2012; Schwaber & Sutherland, 2020).

The Daily Scrum (DS) is an event that has a period of 15 minutes or less, occurs every day during a sprint and has the purpose to answer to the following questions: *"What did I accomplish since the last DS?"*; *"What do I plan to work on by the next DS?"*; and *"What are the obstacles or impediments that are* *preventing me from making progress?".* This is done to improve communication, to find hindrances, and to encourage fast problem solving and decision making (Rubin, 2012; Schwaber & Sutherland, 2020).

The Sprint Review is one of the *"inspect-and-adapt"* events, where the development team delivers and shows the increment, and aims to examine and adjust the outcome that is being produced. This means that during this event the team shows their output from the sprint to the main stakeholders. If during the Sprint Review there are changes to be made, they need to be aligned in the Product Backlog (Rubin, 2012; Schwaber & Sutherland, 2020).

The Sprint Retrospective is the second of the *"inspect-and-adapt"* event and the last of a sprint. This is where the process, people, relations and tools are examined and adjusted in order to have an improvement for the next sprints (Rubin, 2012; Schwaber & Sutherland, 2020).

Product Backlog Grooming consists in the process of evaluating and modifying the product backlog items, usually by adding detailed information, estimations, and order (Rad & Turley, 2013).

2.2.3 Scrum Artifacts

The Scrum Artifacts are composed of the Product Backlog, the Sprint Backlog, and the Increment, that have as commitment, respectively, the Product Goal, the Sprint Goal, and the Definition of Done. Table 3 depicts the Scrum artifacts, their description, and their commitment.

Table 3 – Scrum artifacts description and commitment.
Adapted from Schwaber and Sutherland (2020)

Artifacts	Description	Commitment
Product Backlog	› List of what is required. › Example: features, bugs fixes.	Product Goal
Sprint Backlog	› Why? → Sprint Goal › What? → Sprint › How? → Increment	Sprint Goal
Increment	› Result of the Scrum process; › Adds value to the product; › Multiple Increments may be created within a Sprint.	Definition of Done

The Product Backlog is the source of work of the Scrum team. This work contains work packages from several sizes, list of requirements, assignments, and improvements to have detailed items (Schwaber & Sutherland, 2020). In relation to the Sprint Backlog, it consists of the list of assignments which means the team explains precisely the task, since the work demanded to the feature tests needed during a Sprint. In this way, the team can trace the work and manage it during the Sprint (Rubin, 2012).

The Increment is inspected during the Sprint Review and consists in the total of all completed items at the end of a Sprint (Schwaber & Sutherland, 2020).

2.3 Scaled Agile Framework

The Scaled Agile Framework (SAFe) merges Agile with Lean. This means that it is built on the Lean-Agile values and principles, which allows to have a better alignment with the customer needs, while also being focused on program and portfolio management and incorporate other agile frameworks such as Scrum, Extreme Programming and Product Development Flow (Knaster & Leffingwell, 2020; Trienekens et al., 2016). SAFe integrates four configurations designated by Essential SAFe, Large Solution SAFe, Portfolio SAFe and Full SAFe. In addition, it has four key levels that correlate between them, namely, the Program, the Team, the Portfolio, and the Large Solution. Starting from small development teams, it is intended that these teams work together, simultaneously, and cooperate, staying aligned in order to add value to the final product (Alqudah & Razali, 2016; Leffingwell et al., 2018).

This methodology allows companies to raise their outcomes and to own business agility which means that great improvements are made in time to market, employee engagement, quality, customer satisfaction (Knaster & Leffingwell, 2020). SAFe, as a philosophy, states that the managers, leaders, and executives of a given organization are the ones who provides success and continuous improvement of Lean-Agile development to the team and to the systems. In order for this to happen, it is needed that the leaders have the knowledge to teach, empower and involve teams and people to achieve their highest capacity (Leffingwell et al., 2018).

2.3.1 SAFe Values and Principles

The Scaled Agile Framework promotes four values as their core, namely, alignment, built-in quality, transparency, and program execution. To keep up with a dynamic environment, constant changes, and teams in different parts of the world, the value Alignment is required. Agile teams are good when they are autonomous, although they need alignment and a strategy based on the company's goals to succeed even further. This is a responsibility that cannot fall on them, no matter how good they are. This core value occurs when everyone works with the same goal. It enables empowerment, autonomy, and decentralized decision-making, allowing the local decision makers to implement better decisions. Throughout the development lifecycle, Built-in Quality guarantees that every component and increment of the solution meets quality requirements. The relevance of internal quality increases with the size of the system; hence, there can be no doubt about the necessity of integrated quality in large-scale systems.

Quality thinking is organized around five areas in SAFe's integrated quality: Flow, Architecture and Design Quality, Code Quality, System Quality and Release Quality. In relation to the value Transparency, which is offered through SAFe practices, is a facilitator of trust. When a business or development can safely rely on another to operate with integrity, trust exists. No one can establish high-performance teams and programs without trust, and no one can create the confidence required to make and keep acceptable pledges without it. SAFe focuses strongly on working systems and business outcomes, because without such focus, the rest of SAFe cannot succeed. This is called Program Execution. Developing complex solutions is hard, so teams will struggle to deliver good solutions, reliably and efficiently. The Agile Release Train (ART) helps them with this, and that is why SAFe starts by implementing Essential SAFe. Value streams deliver value depending on the ARTs and Solution Trains. Alignment, Transparency and Built-in Quality give the team an initial help to focus on execution. This makes the team execute better and better in each Program Increment (PI) (Leffingwell et al., 2021a).

SAFe is based on ten principles, these being economic concepts that inform the roles and practices of SAFe. In order for a company to implement these practices, it is fundamental that the organization first understands the following principles, since practices may differ from case to case (Knaster & Leffingwell, 2020). The principles are: 1. Take an economic view; 2. Apply systems thinking; 3. Assume variability and preserve options; 4. Build incrementally with fast, integrated learning cycles; 5. Base milestones on an objective evaluation of working systems; 6. Visualize and limit WIP, reduce batch sizes, and manage queue lengths; 7. Apply cadence, synchronise with cross-domain planning; 8. Unlock the intrinsic motivation of knowledge workers; 9. Decentralize decision-making; and 10. Organize around value (Knaster & Leffingwell, 2020; Leffingwell et al., 2021b).

2.3.2 Essential SAFe Configuration

For an organization to have a configuration that suits in its own business requirements, SAFe has four configurations, Essential SAFe, Large Solution SAFe, Portfolio SAFe and Full SAFe, that assist the enterprise with solutions and a request of a small number of teams with the purpose of having individuals to produce and deliver.

The Essential SAFe configuration, which is illustrated in Figure 5, is the best configuration for the ones who are starting to implement SAFe, because it is the most basic configuration, also it is considered *"the heart of SAFe"*. This configuration mentions the most critical elements of the framework in order to achieve the advantages that SAFe provides (Leffingwell et al., 2018). These critical elements consist in a minimal set of roles, events, and artifacts (Knaster & Leffingwell, 2020). In Essential SAFe, shown in

Figure 5, the Agile Release Train (ART) is a crucial organizing structure where agile business and technical teams, key stakeholders, and other resources are committed to a critical, continuous solution quest. SAFe is driven by the ART's long-lived, flow-based, self-organizing nature, which provides business agility. The vast majority of trains are virtual entities that span organizational and geographic boundaries (Knaster & Leffingwell, 2020).

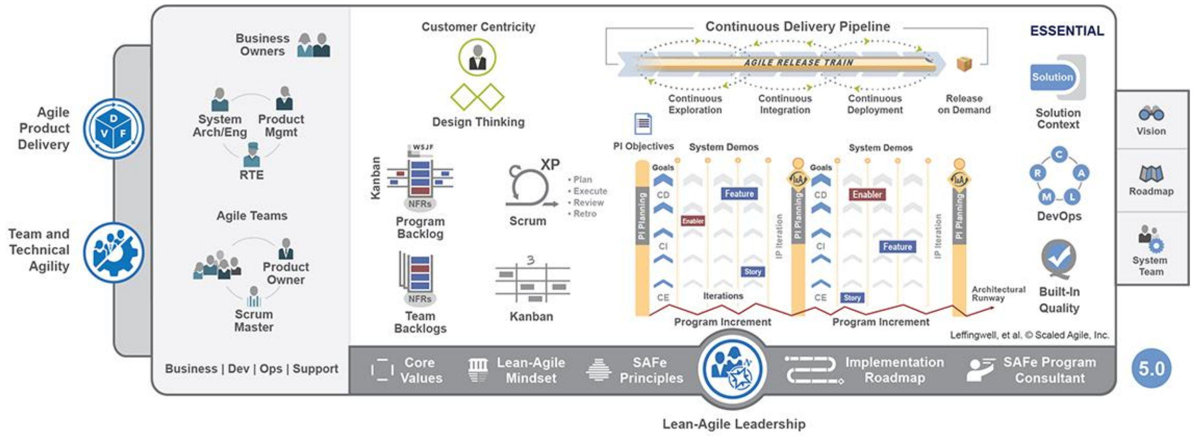


Figure 5 – Essential SAFe Configuration
Leffingwell et al. (2021)

2.4 Capability Maturity Model Integration

In 2018, ISACA-CMMI Institute released a new version of CMMI, called CMMI 2.0, which contains several updates and changes. However, some of the literature used in this dissertation, still contains articles that refer to the 1.2 and 1.3 version.

According to ISACA-CMMI Institute (2020), the Capability Maturity Model Integration (CMMI) was created by industry and CMMI Institute product teams. It combines a set of best practices to help organizations enhance the performance of their core business processes while also providing a light roadmap for establishing, improving, and maintaining capability. As the most significant disadvantages of complex maturity models are the time and the resources required to maintain and keep them current with business, technological, and market demands. The CMMI architecture was created to counteract the disadvantages of complex maturity models, which are the time and resources required to maintain the models updated, and technological and market demands. It aims to be flexible, agile and to evolve as the technology and markets develop. This allows relevant content to be created to keep up with those factors. CMMI works as a guideline that every company can use to apply the best practices, providing a higher level of service and product to the customers. It shows and explains the steps that are needed, in order to improve performance, quality, costs, schedules and functionality (ISACA-CMMI Institute, 2020).

CMMI intends to integrate benefits and lessons learned from different models, therefore, it generates a process improvement model, collects best practices, assists in the organization and improvement of the company's activities and in the coordination of multidisciplinary activities, and allows to align business objectives with process improvement objectives. Therefore, its objective is to help improve the organization's processes and the ability to manage the development, acquisition and maintenance of the product or service. In this way, CMMI enables the identification of approaches that support the company to assess organizational maturity or process capacity, which will lead the organization to establish improvement priorities and implement improvements (CMMI Product Team, 2010). According to CMMI Institute (2020), CMMI is divided in four categories: Doing, Managing, Enabling, and Improving. These are split in a total of 10 sub-categories called Capability Areas and these are divided in 25 Practice Areas overall, as can be seen in Figure 6.

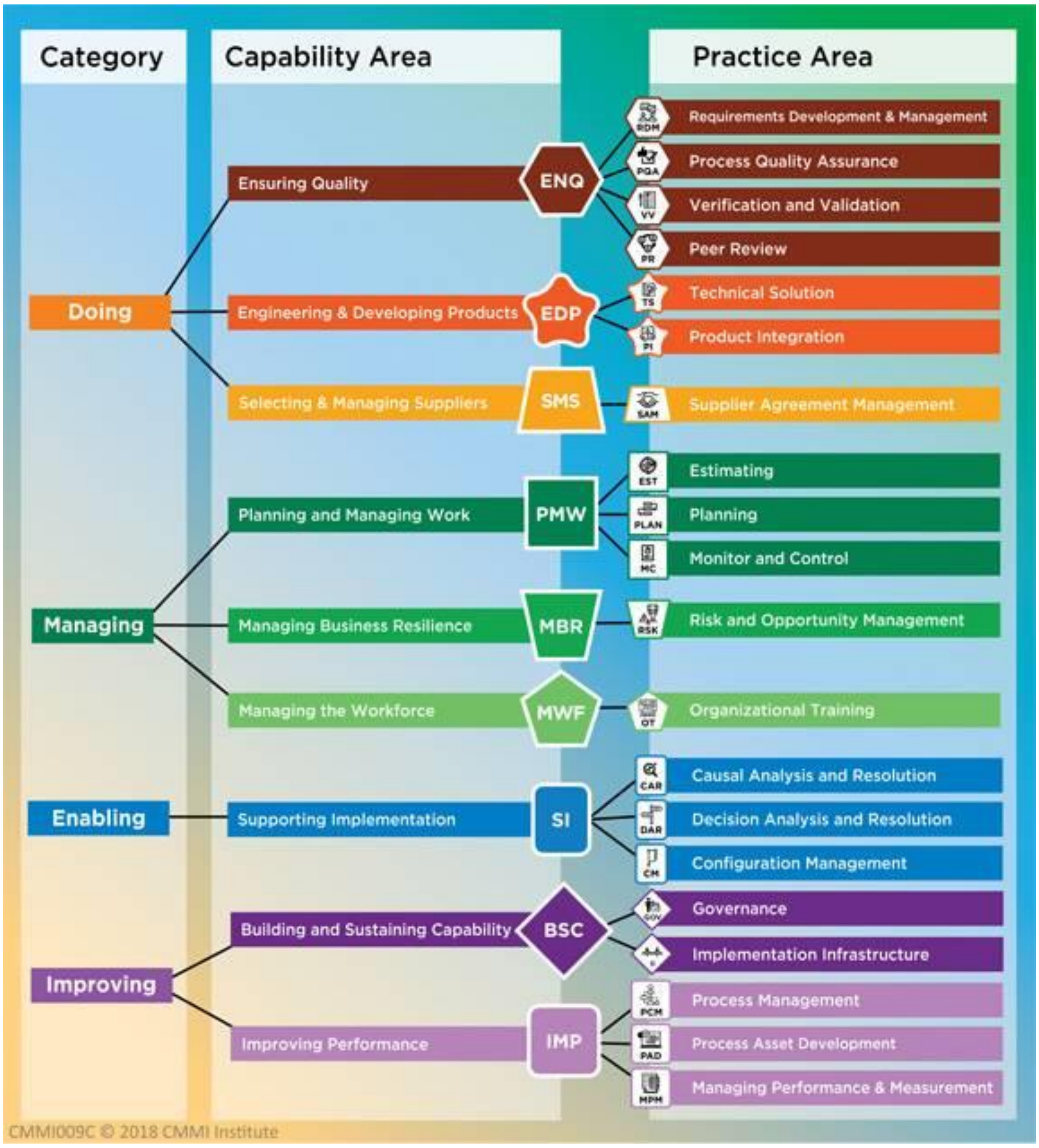
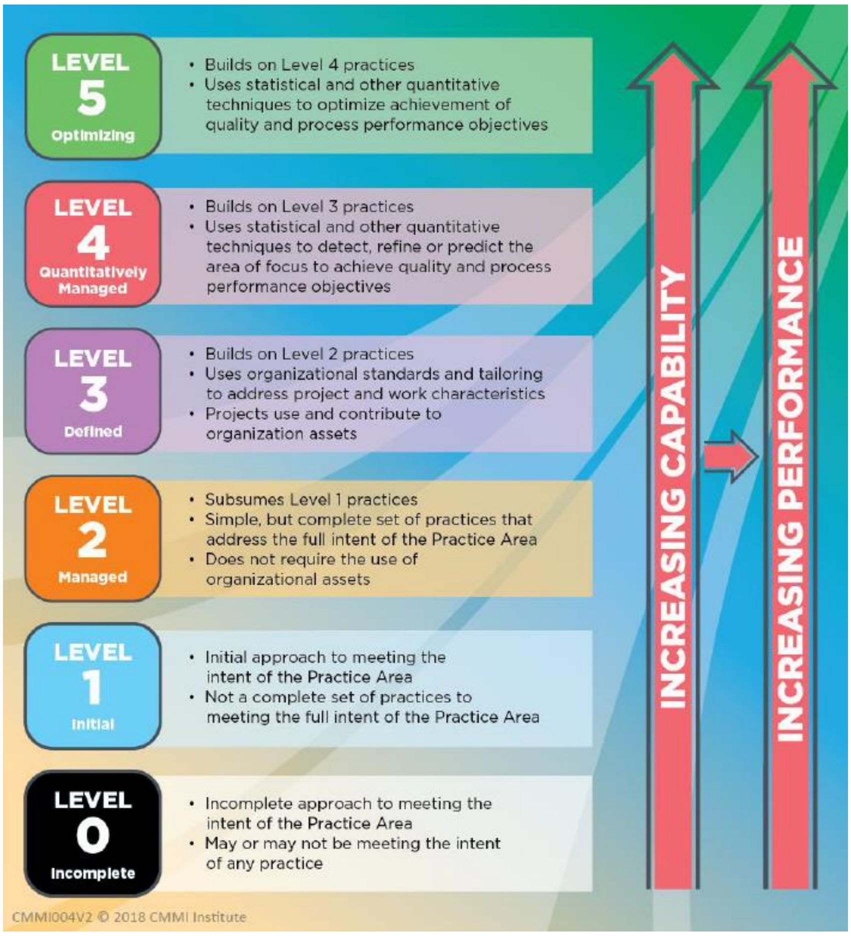

Category
Capability Area
Practice Area
Doing
Managing
Enabling
Improving
Ensuring Quality
ENQ
Engineering & Developing Products
EDP
Selecting & Managing Suppliers
SMS
Planning and Managing Work
PMW
Managing Business Resilience
MBR
Managing the Workforce
MWF
Supporting Implementation
SI
Building and Sustaining Capability
BSC
Improving Performance
IMP
RDM
Requirements Development & Management
PQA
Process Quality Assurance
VV
Verification and Validation
PR
Peer Review
TS
Technical Solution
PI
Product Integration
SAM
Supplier Agreement Management
EST
Estimating
PLAN
Planning
MC
Monitor and Control
RSK
Risk and Opportunity Management
OT
Organizational Training
CAR
Causal Analysis and Resolution
DAR
Decision Analysis and Resolution
CM
Configuration Management
GOV
Governance
II
Implementation Infrastructure
PCM
Process Management
PAD
Process Asset Development
MPM
Managing Performance & Measurement
CMMI009C © 2018 CMMI Institute

Figure 6 – Categories, Capability Areas and Practice Areas of CMMI.
(ISACA-CMMI Institute, 2020)

Categories are developed in this technique as logical groups of linked competence areas that solve difficulties faced in producing or delivering solutions, by using lessons learned from the experience of many organizations, generating small groups of subjects through a list, helping to remember and comprehend them more easily. The categories follow a performance improvement direction, starting with simple activities and progressing through managing them to be more efficient, allowing them to be more effective, and eventually continuously enhancing them to produce greater results. The Doing Category links to Capability Areas that produce and deliver quality solutions; the Managing Category connects to the Capability Areas that plan and manage implementation of solutions; the Enabling Category is associated to Capability Areas that support solution implementation and delivery; and the Improving Category combines Capability Areas that sustain and improve performance. A Practice Area (PA) is defined as a group of practices that outline the key actions required to accomplish a certain goal or value (ISACA-CMMI Institute, 2020).

The Practice Areas (PAs) are evaluated in six evolutionary levels, from 0 to 5, that present how well the company is performing in terms of the PAs (ISACA-CMMI Institute, 2020). In order to better understand these levels, Figure 7 is shown.



LEVEL 5 Optimizing
- Builds on Level 4 practices
- Uses statistical and other quantitative techniques to optimize achievement of quality and process performance objectives

LEVEL 4 Quantitatively Managed
- Builds on Level 3 practices
- Uses statistical and other quantitative techniques to detect, refine or predict the area of focus to achieve quality and process performance objectives

LEVEL 3 Defined
- Builds on Level 2 practices
- Uses organizational standards and tailoring to address project and work characteristics
- Projects use and contribute to organization assets

LEVEL 2 Managed
- Subsumes Level 1 practices
- Simple, but complete set of practices that address the full intent of the Practice Area
- Does not require the use of organizational assets

LEVEL 1 Initial
- Initial approach to meeting the intent of the Practice Area
- Not a complete set of practices to meeting the full intent of the Practice Area

LEVEL 0 Incomplete
- Incomplete approach to meeting the intent of the Practice Area
- May or may not be meeting the intent of any practice

Figure 7 – Description of the evolutionary levels.
(ISACA-CMMI Institute, 2020)

2.5 Capability Maturity Model Integration and Agile Methodologies

This literature review contains articles that refer to the 1.2 and 1.3 versions of the CMMI by the CMMI Product Team.

Scrum techniques do not entirely meet the needs of organizations seeking greater levels of maturity. Although, despite the fact that Scrum does not cover all aspects of project management, it has been determined that it can be made to work with CMMI. The majority of the basics for institutionalizing CMMI project management process areas linked to maturity level 2 are developed without affecting the companies' agility. At the project level, CMMI concentrates on what projects do, rather than the development technique employed, whereas Agile methodologies concentrate on how projects produce products. CMMI and Agile may work together to create synergies that benefit the organization that uses them. Agile development teams are now common in many CMMI-adopting businesses. In contrast, CMMI may be successfully implemented in an Agile environment using an iterative and time-boxed methodology. Agile techniques give software development guidelines that are not included in the CMMI practices. CMMI offers a set of systems engineering techniques that assist large projects adopting agile. In addition, CMMI provides process management and support techniques that aid in the deployment, maintenance, and continuous improvement of an agile methodology (Glazer et al., 2008).

The software industry has been growing in size and competition, so software companies have shifted away from classic software development techniques, toward newer and more adaptable ones that are able to deal with new developments. CMMI does not apply to unpredictable corporate settings, which implies that processes must not only react but also welcome change. Agile software development (ASD) emerged from these needs, with the definition of the Agile Manifesto. ASD focuses primarily on improving the productivity of software development, relationships within the development team, the implied knowledge processes, adaptive planning and focusing on being lightweight. This focus is attained by making the client a part of the development team and working in short iterations of software development. Many big businesses use CMMI to indicate that they are a mature organization, with high degrees of CMMI compliance being linked to improvements in software quality, budget and milestone fulfilment, and customer satisfaction. This topic might be solved by analysing the interrelationships, limitations, and adaptations between agile and CMMI, and attempting to extend agile to meet CMMI. CMMI works as a guideline on what has to be done, whereas agile techniques demonstrate how to implement them (Diaz et al., 2009).

When adopting agile methods in vast projects, the challenge is to maintain small teams connected and coordinated during the project, to assure its success while sticking to agile principles and values. This requires ensuring alignment through a variety of activities, such as, release planning, test planning, continuous integration, and the use of feature-oriented rather than component-oriented teams. Those activities can be challenging for large-scale projects to achieve, from a system engineering view. These alignment and coordination tasks, which are required in bigger and more complex projects, are covered in the CMMI Engineering, Risk Management, and Integrated Project Management process areas. As a result, CMMI provides approaches that can assist in lowering the chance of anything going wrong (Glazer et al., 2008).

Scaling agile methodologies have been gaining recognition in large projects by implementing a top layer and applying clear and early attention to non-functional requirements and product architecture. The software development community is gradually learning how to efficiently adapt current agile methodologies to work on large-scale, complex projects. Some organisations are seeking large-scale enterprise agile deployment. The amount of documentation and implementation assistance available to support this is rather limited. Agile initiatives will not improve in the absence of an organizational environment that supports process development, measurement, feedback, training, and improvement, outlined in the CMMI. The CMMI practices cover institutionalizing and enhancing deployed processes, whether agile or not, which means it promotes a long-term view of process adaptation and improvement across the organization (Glazer et al., 2008).

Marçal et al. (2007) considered only the project management process areas, where for each process area they performed a mapping between the specific practices of the project management process and the Scum ceremonies. During their investigation, they were able to discover gaps and strengths. They estimated the proportion of each process area that was covered by the Scrum practices. Following that, they aggregated the data to provide an overall view.

Policies in big companies require that all components of the organization reach specific levels of maturity. Simultaneously, agile approaches are gaining popularity. This creates a challenge of combining CMMI with agile. It has been demonstrated that projects that use agile methodologies with slight adjustments may reach the level 2 of the CMMI. However, it is unclear from the different reports of successful agile projects how agile techniques contribute to the fulfilment of CMMI process areas, where they need to be changed and where they clash with CMMI goals (Fritzsche & Keil, 2007). Furthermore, Fritzsche and Keil (2014) recommend that research should be undertaken to determine how agile approaches may be modified to achieve certain CMMI levels.

3. METHODOLOGY: ACTION RESEARCH

This chapter intends to explain the methodology used during this dissertation, how it was adopted to the context of development and which techniques and procedures were used to assist the methodology in order to acquire data and achieve results.

3.1 Action Research Phases

The core of an action research is to be on the field, in other words, it starts with the identification of an issue, followed by an attempt to solve it and then continues with the many learnings that can be found doing the same process, repeatedly, until the most suitable solution is found (O'Brien, 1998). The action research seeks to assist in real problems of people in a crisis and generate value and useful knowledge for social sciences. It also intends to help people with their problems by enhancing their self-help skills (Susman & Evered, 1978).

Figure 8 presents the Action-Research Methodology. The first phase, Diagnosing, consists in the identification of an issue. During the second phase, Action Planning, several lines of action are being considered. The third phase, Taking Action, incorporates the choosing path of an action. The next step, Evaluating, intends to analyse the impacts of the action. In the last phase, Specify Learnings, the information from the previous step is used to create knowledge that can be generalized and applied in other areas (O'Brien, 1998).

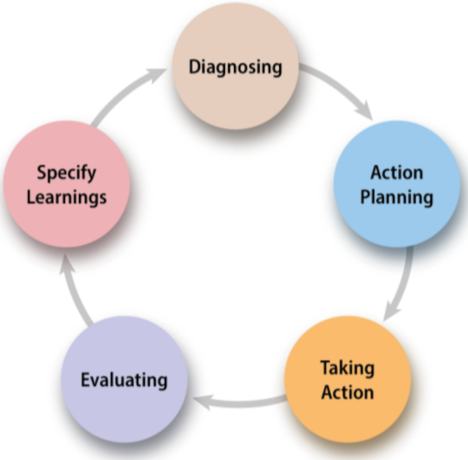


Figure 8 – Action-Research Methodology.
Adapted from O'Brien (1998)

Due to the time constraints associated with the development of this research project, only a single official iteration of the action research cycle could be completed, which did not allow the iterative development of a continuous improvement process. As a result, an adjustment to the action-research methodology was carried out to achieve the dissertation outcomes. There are differences in this new methodology adopted in relation to the original one, the initial (Diagnosing) and final (Specify Learnings) phases are only performed once and the other ones (Action Planning, Taking Action, and Evaluating) are repeated iteratively, as illustrated in Figure 9.

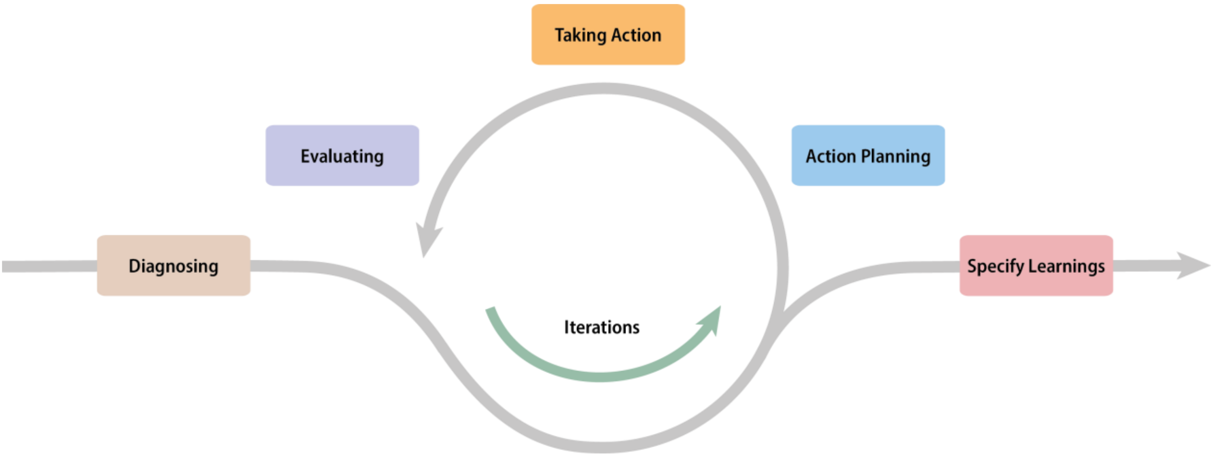


Figure 9 – Adapted Action-Research Methodology.
Adapted from Ribeiro et al. (2018)

As will be discussed in the next chapter, during the Diagnosing phase, a study and examination of the needs of the team and project was carried out. Many CMMI practice areas that needed work were detected. With the interest of improving, controlling, and tracking the refinements of the two practice areas selected, Peer Review and Configuration Management, the PAs were examined independently, and each of that PAs had its own action research cycle.

For the Action Planning phase, it was discussed which was the best way to implement the control process in the team, with the purpose of accomplishing the implementation process and make improvements in the performance of the team. In the third step, Taking Action, it was deployed the process implementation created by the team and the implementation of the control process. In these two phases other methods, tools and activities were allocated in order to meet the needs of the team and the project.

In the Evaluation phase it was analysed whether or not the implemented procedure had issues that required improvements. In case problems existed in the process, it was needed to go to the Action Planning phase of the action research cycle and repeat the cycle. If there were no complications, the information generated in Taking Action would be evaluated. Which means that after this, the focus persisted on a comparison between the initial and final status of the team in the areas evaluated and on the analysis of the fulfilment of the established objectives. And at last, in Specify Learnings, the implementation of the process can be considered successful in the team and, also, can be taken in account that the control process has worked.

It is important to add that most iterations had an informal evaluation from the Developers, since they are the users of the implementations done and the ones that benefit from that. Therefore, when it was needed to do another iteration, the improvements made in the process were done according to the feedback of the team members, which means that their perception was not formally gathered. After one cycle of the action research had been reached, the team's input was formally collected.

3.2 Techniques and procedures for data collection and analysis

Tesfay (2021) describes that continuous improvement arises when the right improvement happens at the right time, accessing to the best combination of tools that allows to get the most advantage out of continuous improvement. Tesfay (2021) also highlights that small and positive improvements must be recognized and encouraged, stating that what is important is to be involved in the process of continuous improvement.

In Development of a Team Quality Control Method subchapter, the methods Plan-Do-Check-Act (PDCA) and the Swimlane Diagram that were used to assist the researcher to create a quality control method for a continuous improvement in specific processes in the team will be mentioned.

The PDCA consists of an iterative management process that is used in corporations to help regulate and improve processes and products (Project Management Institute, 2017). The PDCA is a multi-step methodology for quality improvement. Its first stage is called Plan and aims to create a strategy for improvement. After this step, emerges the need to implement the plan and this one is named Do, and, as the third step, known as Check, the plan's consequences are verified. In the last step, Act, the outcomes are analysed in order to learn and forecast (Swamidass, 2000).

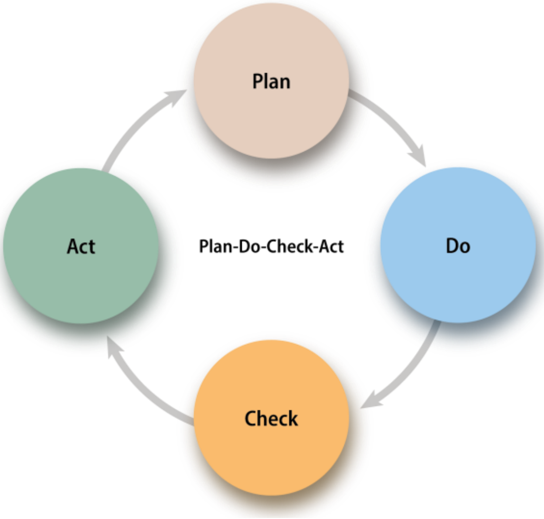
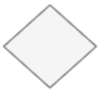


Figure 10 – Plan-Do-Check-Act (PDCA) Cycle.
Adapted from Tesfay (2021)

The Swimlane Diagram is a type of flowchart. A flowchart is a graphical representation of work activities in a sequential order that allows to generate, develop, or provide a result. The Swimlane Diagram shows the workflow, which means, it reveals the set of work activities and resources that occur in each part, demonstrating that the various parts interrelate (Damelio, 2011).

Table 4 shows the symbology used in the Swimlane Diagram of this dissertation.

Table 4 – Flowcharting symbols.
Adapted from Sharp and McDermott (2009)

	Terminator Symbol		Process Symbol		Decision Symbol

During the subchapter Development of a Team Quality Control Method, the Swimlane Diagram and the PDCA will be working together, which means that, usually, the Swimlane Diagram identifies the intervening parts, however, for this particular case, the researcher adapted it to the PDCA. Instead of identifying the intervening parts, the tasks allocated to each phase of the Plan-Do-Check-Act cycle will be identified, as can be shown in Figure 11.

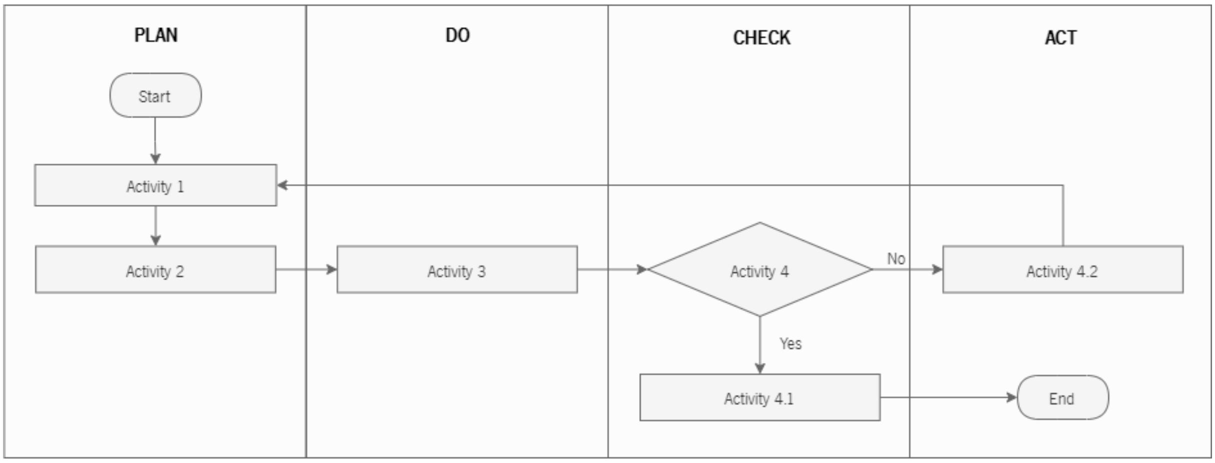


Figure 11 – Swimlane Diagram in conjunction with the PDCA model.
(Author's elaboration)

During this dissertation, a survey was also developed in order to help to generate one of the evidences as result of the project carried out. The survey was applied to seven people, namely, the team members. The intention was to verify if the processes have been successfully implemented and if all the members are informed and aware of those processes. The survey was applied at the end of this project, so it was possible to have an awareness of the status of the knowledge of the team in relation to the processes implemented and strengthen the results of the researcher.

The template of the survey was the following:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.
2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.
3. Is it clear where the versioning process is defined? Yes/No. Justify, please.
4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.
5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.
6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.
7. Is the configuration management system frequently audited? Yes/No. Justify, please.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.
2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.
3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.
4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.
5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Thank you very much for your collaboration! Your contribution is essential.

4. Case Presentation

This chapter seeks to explain the case that was studied during this dissertation. It will start by presenting the project and the team and after that the researcher intervention is explained, this means the role and impact of the researcher in this case. Lastly, it will be mentioned the initial state of the case based on the study of the researcher.

4.1 Presentation of the Project and the Team

The project Ride Care is being carried out at Bosch Car Multimedia Engineering Department. It is an innovation project that is under development in Braga (Portugal), the United States, Germany, Bulgaria, and Mexico. This project is composed by one algorithm development and deployment team located in Braga; three teams responsible for mechanics, hardware, and architecture in the United States; one cloud and embedded software development team in Germany; one embedded software development team in Bulgaria; and one system testing team in Mexico.

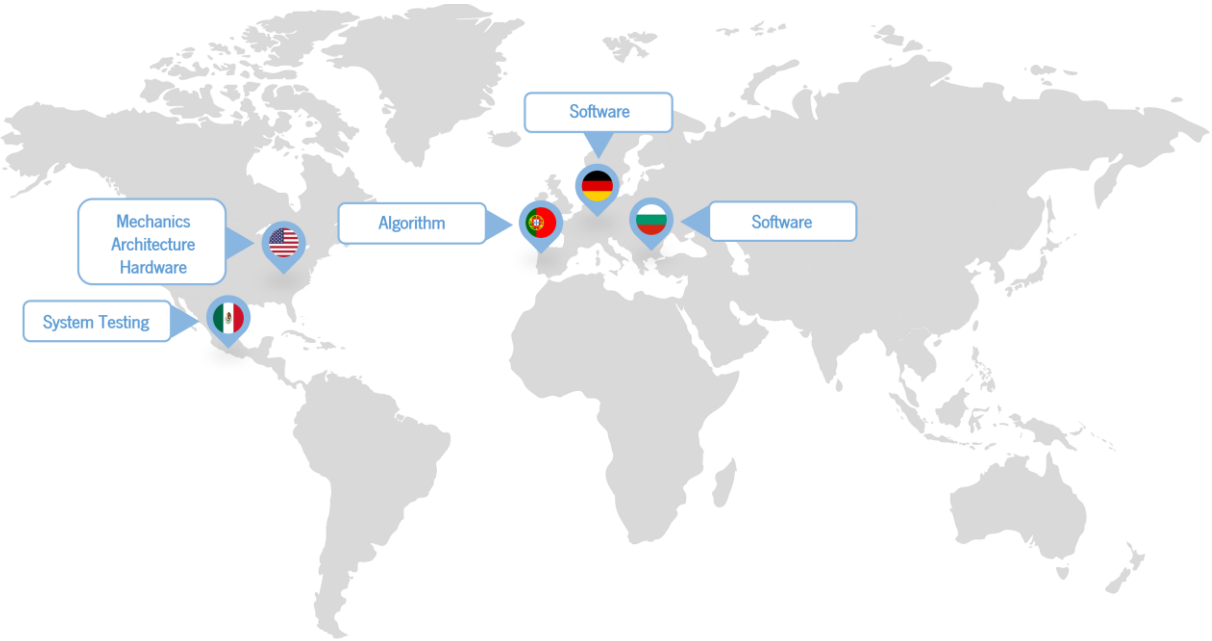
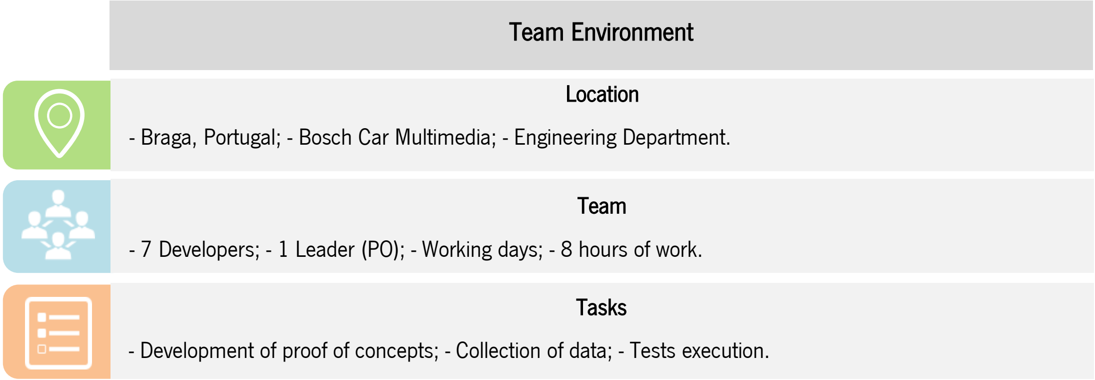


Figure 12 – Name and location of the teams involved in the project.

All the teams interact to create a final product. Braga's team, named Algorithm Team, is formed by one team leader, that also assumes the role of the Product Owner (PO), and seven Developers. The leader organizes and manages the team, and the Developers have knowledge in areas of deep learning, machine learning and computer vision and can be multifunctional. Figure 13, explains the environment where this dissertation was carried out and gives more information about the team.



Team Environment

Location

- Braga, Portugal; - Bosch Car Multimedia; - Engineering Department.

Team

- 7 Developers; - 1 Leader (PO); - Working days; - 8 hours of work.

Tasks

- Development of proof of concepts; - Collection of data; - Tests execution.

Figure 13 – Braga's Team Environment.

Furthermore, the project and the team operate under agile environments, in this case, SAFe and Scrum methodologies. The project is developed through Product Increments, which last two months each and are divided in four sprints of two weeks each, the last of which is to improve the work done during the previous sprints, to prepare and to plan the next Product Increment.

4.2 Intervention of the Researcher

Since this action research is a participatory collaboration, the researcher was incorporated in the Algorithm Team as their Scrum Master, taking up the functions of the previous one, which had left. Thus, it was possible to help the team members and to assist them in improving and supporting in the ceremonies. Getting a new team's commitment is one of the most challenging aspects of integrating them for intervention, so it was decided to make a fluid transition. For this reason, the intervention's core purpose and objective were communicated to all team members from the start, emphasizing that they would be the first to benefit from it.

4.3 Identification of the Initial Status of the Case

In order to implement any improvements in the team, it is needed to identify the team status before applying any method. Therefore, to begin this phase, it was asked to the PO about which of the CMMI PAs were considered the most relevant for this project, taking into consideration the area of the project that the team is responsible for.

Table 5 was created as an output of this process. Therefore, the identified relevant Capability Areas were: Ensuring Quality (ENQ); Engineering and Developing Products (EDP); Planning and Managing Work (PMW); Supporting Implementation (SI); and Sustaining Habit and Persistence (SHP). Regarding the PAs, the following ones were considered: Requirements Development and Management (RDM); Peer

Reviews (PR); Technical Solution (TS); Estimating (EST); Planning (PLAN); Monitor and Control (MC); Configuration Management (CM); and Implementation Infrastructure (II).

Table 5 – Categories, Capability and Practice Area, and Evolutionary Levels.

Categories	Doing						Managing						Enabling		Improving	
Capability Area	ENQ				EDP		PMW						SI		SHP	
Practice Area	RDM		PR		TS		EST		PLAN		MC		CM		II	
Evolutionary Level	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2

To present proposals for improvements and implement them, a diagnosis was made on the stages of implementation of the selected PA, thus the caption was adopted from Software Engineering Institute (2006) and shown in Figure 14.

🟥 Not Implemented 🟧 Partially Implemented 🟨 Largely Implemented 🟩 Fully Implemented ⬜ Not Applicable

Figure 14 – Caption of the Stages of Implementation.

According to the information available in ISACA-CMMI Institute (2020), a table for each PA mentioned in Table 5 was created. Having this, it was possible to begin to collect and analyse information to present the state of the team before applying a control process and to identify the priority of the areas that needed to be worked on. In order to ascertain the initial state of the team, it was decided that a set of interviews would be done to the team leader, since this is the most appropriate person to answer about the state of the team, through weekly meetings, where the PO had to explain, for the PAs, what the team does and the evidences of that and what the team does not do. They will be explained at the end of this chapter. As an output of the meetings, Table 6 was developed, showing the status of the team in relation to the highlighted PAs. All the PAs presented in Table 6 were evaluated as far as the Level 2.

Table 6 – Stages of Implementation of the PAs.

Categories		Doing			Managing			Enabling	Improving
Capability Area		ENQ		EDP	PMW			SI	SHP
Practice Area		RDM	PR	TS	EST	PLAN	MC	CM	II
Evolutionary Level 1	1.1	🟧	🟥	🟩	🟩	🟩	🟩	🟨	🟩
	1.2					🟩	🟩		
Evolutionary Level 2	2.1	🟧	🟥	🟨	🟨	🟨	🟨	🟥	🟨
	2.2	🟧	🟥	🟥	🟩	🟥	🟨	🟨	🟥
	2.3	🟧	🟥	⬜	🟥	🟩	🟧	🟥	
	2.4	🟧	🟥			🟩	🟥	🟥	
	2.5	🟥				⬜		🟥	
	2.6	🟥				🟩		🟥	
	2.7					⬜			
	2.8					⬜			

In the following tables, the acronyms T is for Team and E for Evidence(s) will be used.

Description of the mentioned evidences shown in the following tables:

Docupedia consists in a repository of documentation about Bosch's internal projects. The repository includes project's goals, timelines, and technical documentation.

Code is a set of instructions given to computers to perform a particular task.

OneNote is the tool used to gather all the information about what the team develops for the project is located.

Jira Software is the software used to gather all the information about the project, this means that is the place where the teams report about bugs, epics, user stories, dependencies, people assigned to the previously mentioned features, story points estimation, et cetera.

Epics consist in a work package that can be divided into tasks (user stories), taking into consideration needs or requirements of the consumer.

PI Board Excel contains the epics, user stories, objectives, story points planned and executed, and the team's velocity and capacity.

Story Points Estimation consists in assigning a certain value to a certain user story based on the Fibonacci sequence.

Planning Poker is the activity where the story points are estimated.

Field Operations Meeting is where the project team activities are monitored in order to guarantee that the service is provided to the client.

Requirements Development and Management Practice Area (RDM PA)

The RDM PA intends to assist in the development and update of the needs and expectations of a given solution, that is, it has as outputs the requirements, plans and work products in order to ensure that the interested parties are aligned and, with this, satisfy the needs and expectations of the customers. Evolutionary level 1 is aimed at acquiring and developing requirements, in other words, registering them, while evolutionary level 2 focuses on their maintenance, consequently developing and maintaining requirements updated and registered (ISACA-CMMI Institute, 2020). In order to understand the team state of this PA, Table 7 was created.

Table 7 – Status of Implementation of the CMMI Levels in RDM PA in the team. Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟧	RDM 1.1	Record requirements.	It is registered, but it is not managed.	Docupedia

T	Level	Description	Comments	E
	2			
🟧	RDM 2.1	Elicit stakeholder needs, expectations, constraints, and interfaces or connections.	It is not done.	Docupedia
🟧	RDM 2.2	Transform stakeholder needs, expectations, constraints, and interfaces or connections into prioritized customer requirements.	It is not done.	Docupedia
🟧	RDM 2.3	Develop an understanding with the requirements providers on the meaning of the requirements.	It is not done.	Docupedia
🟧	RDM 2.4	Obtain commitment from project participants that they can implement the requirements.	It is done, but it is not registered.	-
🟥	RDM 2.5	Develop, record, and maintain bidirectional traceability among requirements and activities or work products.	It is not done.	-
🟥	RDM 2.6	Ensure that plans and activities or work products remain consistent with requirements.	It is not done.	-

As can be observed in Table 7, level 1 is partially implemented in the team, regarded that the team registers the requirements, although that information is not updated. In relation to level 2, there is a higher variation between the several practices (from RDM 2.1 to RDM 2.6), where the first four are partially implemented and the last two are not implemented.

Peer Reviews Practice Area (PR PA)

In line with ISACA-CMMI Institute (2020), the PR PA aims to identify solutions to potential problems or defects in the early stages that are detected by producer's peers or subject matter experts. This allows avoiding costs and risks by identifying potential problems in advance. At evolutionary level 1, work products are analysed, and the problems are reported. At level 2, it is intended that there are procedures and support material to start carrying out peer reviews, so that they help with reviewing the work products and in solving the problems that are identified in the peer reviews. For a better understanding of the team state of the PA Peer Reviews, Table 8 was created.

Table 8 – Status of Implementation of the CMMI Levels in PR PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟥	PR 1.1	Perform reviews of work products and record issues.	It is not done.	-
	2			
🟥	PR 2.1	Develop and keep updated procedures and supporting materials used to prepare for and perform peer reviews.	It is not done.	-
🟥	PR 2.2	Select work products to be peer reviewed.	It is not done.	-
🟥	PR 2.3	Prepare and perform peer reviews on selected work products using established procedures.	It is not done.	-
🟥	PR 2.4	Resolved issues identified in peer reviews.	It is not done.	-

Regarding Table 8, none of the levels of this PA, level 1 and 2, are implemented. This means that all of them are registered as not implemented in the team.

Technical Solution Practice Area (TS PA)

The TS PA focuses on the design and construction of the components of the products and the products. This area intends to design and build solutions that meet the requirements established by the customer and, in addition to providing design and economic solutions, it is also a support in the rework reduction. The evolutionary level 1 of TS is dedicated to building solutions that meet customer requirements. At evolutionary level 2, in addition to construction, the design is also carried out, the project is evaluated, the problems solved and a guide for using the solution is also provided (ISACA-CMMI Institute, 2020). Table 9 assists to a better understand of the team status in TS.

Table 9 – Status of Implementation of the CMMI Levels in TS PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟩	TS 1.1	Build a solution to meet requirements.	It is done in engineering practices.	Repository Code
	2			
🟨	TS 2.1	Design and build a solution to meet requirements.	There are some topics done, some not done, and others not applicable.	Docupedia Code
🟥	TS 2.2	Evaluate the design and address identified issues.	It is not done.	-
⬜	TS 2.3	Provide guidance on use of the solution.	It is not applicable.	-

The TS PA has the level 1 as fully implemented in the team. For the level 2, the TS 2.1 is largely implemented, the T.S. 2.2 is not implemented, and the TS 2.3 is not applicable in the team.

Estimating Practice Area (EST PA)

The EST PA consists of forecasting the effort, cost, duration, resources, and size for a given work in order to develop, acquire and deliver solutions. EST allows to plan, make commitments, and reduce uncertainty, so it is possible to access corrective actions early and allows an increased likelihood of meeting goals. Evolutionary level 1 is designed to enable the development of high-level estimates in carrying out the work, while level 2 is reached when aspects such as the development, updating and use of the scope of what was estimated are fulfilled (ISACA-CMMI Institute, 2020). Table 10 explains the status of the team in the EST area.

Table 10 – Status of Implementation of the CMMI Levels in EST PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟩	EST 1.1	Develop high-level estimates to perform the work.	It is done in the Scrum events through Product Backlog Grooming and Sprint Planning.	OneNote Jira Software
	2			
🟨	EST 2.1	Develop, keep updated, and use the scope of what is being estimated.	It is done through the Scrum process in epics and user stories.	Epics Jira Software PI Board Excel
🟩	EST 2.2	Develop and keep updated estimates for the size of the solution.	It is done through Scrum Process in the Planning Poker, Product Backlog Grooming and Sprint Planning.	Jira Software PI Board Excel
🟥	EST 2.3	Based on size estimates, develop and record effort, duration, and cost estimates and their rationale for the solution.	It is not done.	-

The level 1 of the EST practice is fully implemented in the team. Regarding the level 2, the EST 2.1 is largely implemented, the EST 2.2 is fully implemented, and the EST 2.3 is not implemented.

Planning Practice Area (PLAN PA)

According to ISACA-CMMI Institute (2020), the PLAN PA involves using the estimates to develop a plan that describes what is needed to complete the work within the organization's standards, schedule, and budget, define the resources needed, and obtain confirmation from stakeholders. The PLAN comprehensively describes the project's needs, who is responsible for the pieces of work and how stakeholders relate to the project. This allows to optimize costs and increase the probability of achieving the goals. At evolutionary level 1, organizations only create a list of tasks and define who is responsible for them, while at level 2 the plan is constantly updated and deepened, based on organizational data and with greater interaction from the stakeholders. Table 11 presents the initial status of the team.

Table 11 – Status of Implementation of the CMMI Levels in PLAN PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟩	PLAN 1.1	Develop a list of tasks.	It is done through the Scrum Process in the Daily Meeting and the Product Backlog Grooming.	Jira Software OneNote
🟩	PLAN 1.2	Assign people to tasks.	It is done through the Scrum Process and SAFe. Software Jira enables, automatically, to assign people to user stories, tasks, and/or sub-tasks.	Jira Software
	2			
🟨	PLAN 2.1	Develop and keep updated the approach for accomplishing the work.	In order to perform the Scrum events in accordance with what the team executes, a document with guidelines for each Scrum event is available.	Jira Software PI Board Excel

T	Level	Description	Comments	E
2				
🟥	PLAN 2.2	Plan for the knowledge and skills needed to perform the work.	It is not done.	-
🟩	PLAN 2.3	Based on recorded estimates, develop, and keep the budget and schedule updated.	Even though it is not part of team responsibilities, the team ends up being influenced by what happens in the project and these features are defined in the project.	PI Board Excel
🟩	PLAN 2.4	Plan the involvement of identified stakeholders.	It is done through SAFe ceremonies. JIRA allows the stakeholders to stay informed about what is happening in the project.	Jira Software
⬜	PLAN 2.5	Plan transition to operations and support.	It is not applicable in the team.	-
🟩	PLAN 2.6	Ensure plans are feasible by reconciling available and estimated resources.	It is done through the Scrum Process, during the Sprint Planning.	PI Board Excel
⬜	PLAN 2.7	Develop the project plan, ensure consistency among its elements, and keep it updated.	It is not applicable in the team.	-
⬜	PLAN 2.8	Review plans and obtain commitments from affected stakeholders.	It is not applicable in the team.	-

The team has the level 1 of the practice fully implemented. Also, in relation to the level 2, the team has the PLAN 2.1 largely implemented, PLAN 2.2 not implemented, PLAN 2.3, PLAN 2.4, and PLAN 2.6 as fully implemented, and the PLAN 2.5, PLAN 2.7, and PLAN 2.8 as not applicable.

Monitor and Control Practice Area (MC PA)

The MC PA pays attention to progress in order to apply corrective measures when there is a significant deviation from the plan, schedule or budget, in order to increase the probability of achieving the objectives by taking the necessary measures as soon as possible. At evolutionary level 1, completed tasks are recorded and problems identified and resolved. At CMMI level 2, results are compared to estimates, with the intervention of stakeholders, keeping the transition to operations monitored (ISACA-CMMI Institute, 2020). Table 12 shows how the practice areas of the MC are in the initial case.

Table 12 – Status of Implementation of the CMMI Levels in MC PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
1				
🟩	MC 1.1	Record task completions.	It is done through the Scrum Process in the Daily Meetings, Sprint Planning and Sprint Review.	Jira Software
🟩	MC 1.2	Identify and resolve issues.	It is done through the Scrum Process in the Daily Meetings.	Jira Software PI Board Excel

T	Level	Description	Comments	E
2				
🟨	MC 2.1	Track actual results against estimates for size, effort, schedule, resources, knowledge and skills, and budget.	It is done through the Scrum Process in the Sprint Planning, Daily Meetings and Sprint Review.	Story Points Estimation Planning Poker PI Board Excel OneNote
🟨	MC 2.2	Track the involvement of identified stakeholders and commitments.	It is done through the Scrum's Daily Meetings and SAFe's Art Sync.	Docupedia
🟧	MC 2.3	Monitor the transition to operations and support.	It is done by another team in the project, however the team under study supports the other team.	Field Operations Meeting Docupedia
🟥	MC 2.4	Take corrective actions when actual results differ significantly from planned results and manage to closure.	It is not done.	-

The level 1 of the MC practice area is fully implemented in the team. In terms of the level 2, the team has the levels MC 2.1 and MC 2.2 as largely implemented, for the MC 2.3 as partially implemented, and the MC 2.4 as not implemented.

Configuration Management Practice Area (CM PA)

The CM PA establishes and maintains controls over work products, using auditing and verification methods to reduce inefficiencies and lost work and to increase the ability to deliver the most appropriate solution to the customer. At evolutionary level 1, only version control is used, while at level 2 organizations define which items must be checked and which ones do not need to be checked, baselines are defined for item management and the verification system is constantly updated (ISACA-CMMI Institute, 2020). Table 13 presents the completion status of the CM area in the initial phase.

Table 13 – Status of Implementation of the CMMI Levels in CM PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
1				
🟨	CM 1.1	Perform version control.	It is done at a code level, but not at a documental or specification level.	Code
2				
🟥	CM 2.1	Identify items to be placed under configuration management.	It is not done.	-
🟨	CM 2.2	Develop, keep updated, and use a configuration and change management system.	It is done at a code level, but not at a documental or specification level.	Code
🟥	CM 2.3	Develop or release baselines for internal use or for delivery to the customer.	It is not done.	-
🟥	CM 2.4	Manage changes to items under configuration management.	It is not done.	-
🟥	CM 2.5	Develop, keep updated, and use records describing items under configuration management.	It is not done.	-

T	Level	Description	Comments	E
	2			
🟥	CM 2.6	Perform configuration audits to maintain the integrity of configuration baselines, changes, and content of the configuration management system.	It is not done.	-

In this case, the level 1 is largely implemented in the team. For the level 2 only the level CM 2.2 is largely implemented, and the remaining ones are not implemented.

Implementation Infrastructure Practice Area (II PA)

The II PA aims to ensure that the processes of a given organization structure are being used and improved frequently. This supports the ability to achieve goals and objectives in a consistent, efficient and effective way. The evolutionary level 1 of the II intends to execute the processes in a way that meets the intentions of the practices of the evolutionary level 1. On the other hand, the level 2 intends to provide sufficient resources, funding and training for the development and execution processes and implement it, ensuring that these processes are followed and updated (ISACA-CMMI Institute, 2020). The

Table 14 demonstrates the initial state of the II area.

Table 14 – Status of Implementation of the CMMI Levels in II PA in the team.
Adapted from ISACA-CMMI Institute (2020)

T	Level	Description	Comments	E
	1			
🟩	II 1.1	Perform processes that address the intent of the Level 1 practices.	It is done in the entire process that the team performs.	Jira Software Code Repositories Reporting
	2			
🟧	II 2.1	Provide sufficient resources, funding, and training for developing and performing processes.	There are topics that are done, others that are not done, some that are not applicable, and some are done by a different team in the project.	Jira Software
🟥	II 2.2	Develop and keep processes updated, and verify they are being followed.	It is not done.	-

The level 1 of this PA is fully implemented in the team. Regarding the level 2, this one has the II 2.1 partially implemented and the level II 2.2 as not implemented.

5. METHODS AND RESULTS

This chapter aims to present the team quality control method developed, its creation, implementation, and results. Furthermore, it will be shown the harmonisation method between agile (Scrum) and CMMI created.

5.1 Development of a Team Quality Control Method

This subchapter intends to introduce the control method in the team. It starts with the definition of the method based on a general method which is adapted to the chosen practice areas. After this, its implementation and results are presented.

5.1.1 Creation of the Method

After the output of the previous chapter, a new discussion occurred, this time, with both the PO and the team, considering the timeline available for the researcher in the company. It was identified which were the PAs that they thought would bring more benefits to the work of the team and the project, and the ones that were the most valuable for the team improvement, since they were a software development team.

The CMMI PAs selected were Configuration Management and Peer Reviews. Towards verifying that the implementations were performed, iteration cycles were done to ensure that continuous improvement and the Plan-Do-Check-Act (PDCA) methodology was applied. This methodology motivates continuous improvement and, to implement it on the team, it was decided that the *Plan* would be to define the objective needed to achieve a level of completion in a specific PA. Therefore, the *Do* would be defining the measures necessary to meet the goal of each practice area and perform them. After this step, it would be verified if these measures are being implemented and analysed if they are working as planned - this is known as *Check*, and lastly, the *Act* step is where the results of the analysis would be applied upon to improve the next *Plan*, in other words, identify corrective actions and improvements to implement in the next *Plan* or when it starts to become obsolete, restart the process. Also, it will be used a *Swimlane Diagram* in conjunction with the PDCA to help explain, track, and record the process operation of the improvements in the Configuration Management and Peer Reviews PAs.

Before applying these methods in the previously mentioned areas, a general process for the PA was made, based on the PDCA and Swimlane Diagram methods, as can be seen in Figure 15. This General Process intends to demonstrate explicitly the control process of the creation of a PA process in a team. Thus, it is possible to understand the set of necessary activities to achieve the PA process that is intended to be worked on and applied in the team. The method must be well defined and clear, therefore besides the knowledge of the activities, it is required that these activities are described precisely.

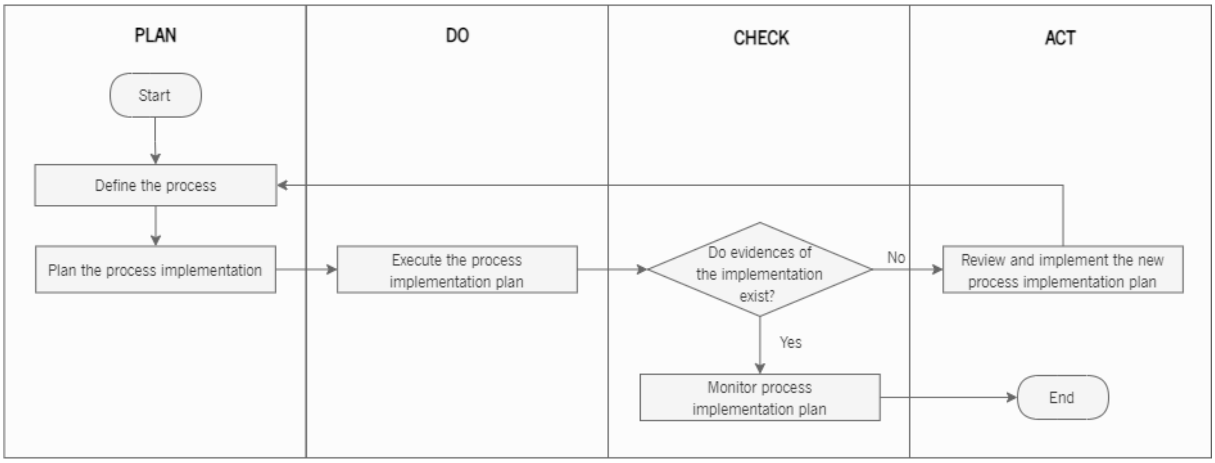

PLAN
DO
CHECK
ACT
Start
Define the process
Plan the process implementation
Execute the process implementation plan
Do evidences of the implementation exist?
No
Review and implement the new process implementation plan
Yes
Monitor process implementation plan
End

Figure 15 – General Process for a Practice Area.

In order to have further detailed information about the steps shown in the Figure 15, the steps *"Define the process"*, *"Plan the process implementation"*, *"Execute the process implementation plan"*, *"Monitor process implementation plan"*, and *"Review and implement the new process implementation plan"* will be explained in the Table 15, Table 16, Table 17, Table 18, and Table 19, respectively.

Table 15 – Information on the implementation of *"Define the process"* step.

"Define the process"	
Responsible	Team Member
Inputs	Practice Area (of the CMMI)
Objectives	Define a process for a Practice Area (of the CMMI).
Outputs	Process defined and updated
Implementation Details	1. Choose a Practice Area (of the CMMI). 2. Define a process for that Practice Area: 2.1. Do a flow diagram of the process; 2.2. Describe all the steps of the process; 2.3. Define the inputs and outputs of the steps. 3. Process approved by the team.

Table 16 – Information on the implementation of *"Plan the process implementation"* step.

"Plan the process implementation"	
Responsible	Process Improvement Responsible
Inputs	Process defined
Objectives	Create a plan for the process implementation.
Outputs	Process implementation plan

"Plan the process implementation"	
Implementation Details	1. Process implementation plan description: 1.1. How the plan will be introduced and implemented; 1.2. Introduce the process implementation plan to the team; 1.3. When the plan will be implemented; 1.4. Who will implement the plan; 1.5. Period of the implementation.

Table 17 – Information on the implementation of *"Implement the process implementation"* step.

"Execute the process implementation"	
Responsible	Team Member(s)
Inputs	Process implementation plan
Objectives	Implement the plan of the process implementation in the team.
Outputs	Process implemented
Implementation Details	1. Implement the process implementation: 1.1. Follow the process implementation plan; 1.2. Implement the process during the timeline proposed.

Table 18 – Information on the implementation of *"Monitor process implementation plan"* step.

"Monitor process implementation plan"	
Responsible	Process Improvement Responsible
Inputs	Process implementation is initiated
Objectives	Monitor and control the plan activities in order to achieve the process implementation plan and have the process implementation applied.
Outputs	Process implementation plan completed
Implementation Details	1. Analyse the process implementation plan: 1.1. Verify that the process implementation plan was complied to; 1.2. Verify that every step of the process implementation was achieved. 2. Designate the process implementation plan as completed.

Table 19 – Information on the implementation of *"Review and implement the new process implementation plan"* step.

"Review and implement the new process implementation plan"	
Responsible	Team
Inputs	Process implementation plan realised
Objectives	Achieve a new process implementation plan. Furthermore, monitor and control the plan activities in order to achieve the process definition, the process implementation plan and have the process implementation applied.
Outputs	New process definition and new process implementation plan created and completed
Implementation Details	Repeat the process creating a new version.

The construction of this general process serves as a foundation to apply a control process in the many existent PAs. Therefore, the first step for each of the PA, CM and PR, will be to explain its purpose and its value to the team. After that, their objectives for the level 1 and level 2 will be mentioned. At last, the control process will be shown and activities clarified.

5.1.2 Implementation of the Method

After having the outputs of the previous section, the implementation phase of the method began. In order to apply this method in the team, it was adapted to each PA selected.

Configuration Management Practice Area (CM PA)

According to CCMI Institute (2020), in order to plan the Configuration Management activities, the control of the work products developed or modified by the project must be carried out. These work products include customer-deliverables and internal work products designated and customer-supplied. Baselines represent an approved version of a work product and provide a clear and accurate understanding for future use. By identifying and managing configuration, change management, and configuration auditing functions of configuration management, it will be feasible to systematically monitor and control changes to baselines and work outputs.

For the CM Practice Area, it was considered as purpose, for the level 1, that the team is required to have records of all the versions of the work products, even if the characteristics of the product or service have been changed. The goal of the level 2 is that the team needs to make control and verification more efficient, constantly update verification processes and establish minimum baselines for the quality of delivery of work products, manage in detail the work products that are under verification and the changes made to them, manage the verification process, and see if this is efficient (CMMI Institute, 2020).

To implement this PA on the team, a Process Control of Configuration Management was created, as can be seen in Figure 16. It began with the definition of the versioning process, where the team created the explanation of the process used to accomplish the steps needed to implement the versioning, implying to have a defined formal process that was of the knowledge of all team members. After this step, it was required to introduce the versioning process to the team, which resulted in establishing a plan of the versioning process implementation. In this case it was decided to use one meeting and one document to explain the process, the steps, and what the team members needed to do when they run into work products that involve versioning. Following this phase, it is time to start implementing the presented plan, which means that the team members will put in practice the implementation, therefore it was used one acceptance criteria dedicated to the versioning in the user story whose deliverable needs to be versioned. In order to verify that the implementation was done, the team members schedule a meeting with the SM showing that the process was done, it stays recorded in the official tool, and the closing of the acceptance criteria is done. In case the defined implementation does not work or when it becomes obsolete, the process must be done again. Due to company confidentiality, the versioning process defined and implemented by the team cannot be shown.

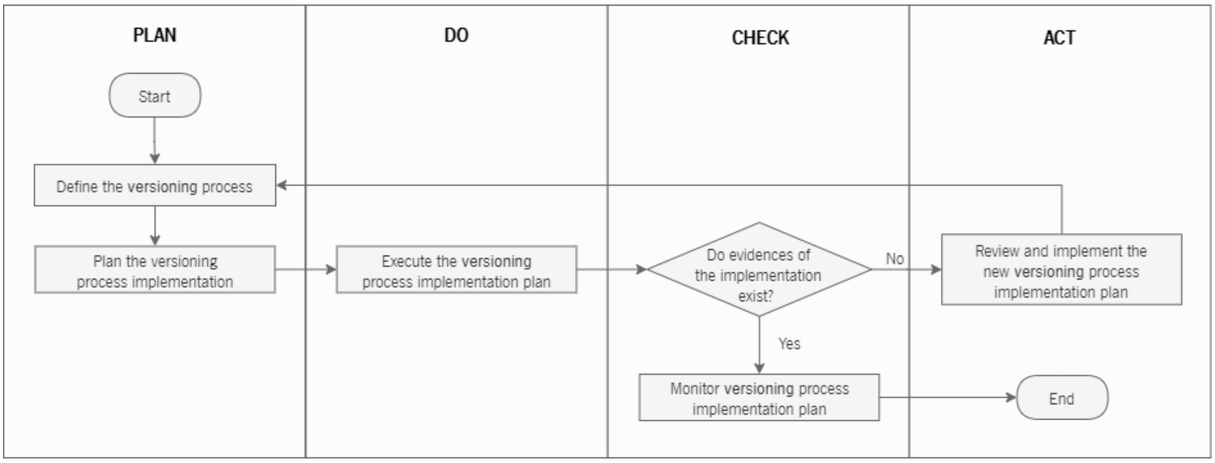

PLAN
DO
CHECK
ACT
Start
Define the versioning process
Plan the versioning process implementation
Execute the versioning process implementation plan
Do evidences of the implementation exist?
No
Yes
Review and implement the new versioning process implementation plan
Monitor versioning process implementation plan
End

Figure 16 – Process Control of Configuration Management.

In order to have further detailed information about the steps shown in the Figure 16, the steps "Define the versioning process", "Plan the versioning process implementation", "Implement the versioning process implementation plan", "Monitor versioning process implementation plan", and "Review and implement the new versioning process implementation plan" will be explained in the Table 20, Table 21, Table 22, Table 23, and Table 24, respectively.

Table 20 – Information on the implementation of *"Define the versioning process"* step.

"Define the versioning process"	
Responsible	Developer(s); Team Member
Inputs	Practice Area of the CMMI
Objectives	Define a versioning process for the Configuration Management Practice Area of the CMMI.
Outputs	Versioning process defined and updated
Implementation Details	1. Choose the Configuration Management Practice Area of the CMMI. 2. Define a versioning process for that PA: 2.1. Do a flow diagram of the versioning process; 2.2. Describe all the steps of the versioning process; 2.3. Define the inputs and outputs of the steps. 3. Versioning process description. 4. Versioning process approved by the PO.

Table 21 – Information on the implementation of *"Plan the versioning process implementation"* step.

"Plan the versioning process implementation"	
Responsible	Scrum Master
Inputs	Versioning process defined

“Plan the versioning process implementation”	
Objectives	Create a plan for the versioning process implementation.
Outputs	Versioning process implementation plan
Implementation Details	1. Versioning process implementation plan description: 1.1. Meeting to introduce the versioning process and the plan of the versioning process implementation; 1.2. Which Program Increment and user story it will need to be implemented on; 1.3. Developer(s) who will implement the versioning process.

Table 22 – Information on the implementation of *“Implement the versioning process implementation”* step.

“Execute the versioning process implementation”	
Responsible	Developer(s); Team member(s)
Inputs	Versioning process implementation plan
Objectives	Implement the plan of the versioning process implementation in the team.
Outputs	Versioning process implemented
Implementation Details	1. Implement the versioning process implementation: 1.1. Follow the versioning process implementation plan; 1.2. Implement the versioning process during the PI and in the specific user story.

Table 23 – Information on the implementation of *“Monitor versioning process implementation plan completed”* step.

“Monitor versioning process implementation plan”	
Responsible	Scrum Master; Process Improvement Responsible
Inputs	Versioning process implementation plan realised
Objectives	To achieve the versioning process implementation plan and have the process implementation applied.
Outputs	Versioning process implementation plan completed
Implementation Details	1. Analyse the versioning process implementation plan: 1.1. Verify that the versioning process implementation plan was complied to; 1.2. Verify that every step of the versioning process implementation was achieved. 2. Designate the versioning process implementation plan as completed.

Table 24 – Information on the implementation of *“Review and implement the new versioning process implementation plan”* step.

“Review and implement the new versioning process implementation plan”	
Responsible	Team
Inputs	New versioning process and new versioning process implementation plan realised
Objectives	Achieve a new versioning process implementation plan and have the versioning process implementation applied.
Outputs	New versioning process created and versioning process implementation plan created and completed
Implementation Details	Repeat the process creating a new version.

Peer Reviews Practice Area (PR PA)

In accordance to CMMI Institute (2020), the Peer Reviews help identify problems and eliminate flaws in outputs, through inspections, structured walkthroughs, audits, and other review procedures. Peer reviews should be done in stages as work products are created; they should be organized and not be confused with management or status reviews. Peer evaluation should be focused on the work product rather than the person who did it. When flaws are discovered during peer review, the author must be informed so that they can be fixed. The most effective peer reviews are made by participants that are impartial and open in their assessments. Peer review data must be protected against misuse, such as punishing or rewarding employees, publicly criticizing or applauding performance, or breaking data security and privacy regulations.

For the PR Practice Area, the main goal considered in order to achieve the level 1 of this Practice Area was that the team needs to review the work that has been done and, in case of issues, register them. To achieve the level 2, the objective is that the team needs to define, streamline, and facilitate processes on how to prepare peer reviews and use their results constructively, in order to increase quality and reduce costs (CMMI Institute, 2020).

A Process Control of Peer Reviews was developed to execute this PA on the team, as shown in Figure 17. It began with the description of the peer reviewing process, in which team members provided an explanation of the procedures required to conduct peer reviews, suggesting the existence of a specified formal process that was known by all team members. Following this, it was necessary to present the peer reviewing process to the team, which led to the development of a strategy for implementing the peer reviewing process. In this situation, it was decided to use a single meeting and a single document to describe the process, the steps, and what the team members should do if they come across work items that require peer review. Following this phase, it is time to start implementing the plan presented, which means that the team members will put in practice the implementation. In order to verify that the implementation was done, the team members schedule a meeting with the SM showing that the process was accomplished and it stays recorded in the official tool. In case the defined implementation does not work or when it becomes obsolete the process must be done again. Due to company confidentiality, the peer reviewing process defined and implemented by the team cannot be shown.

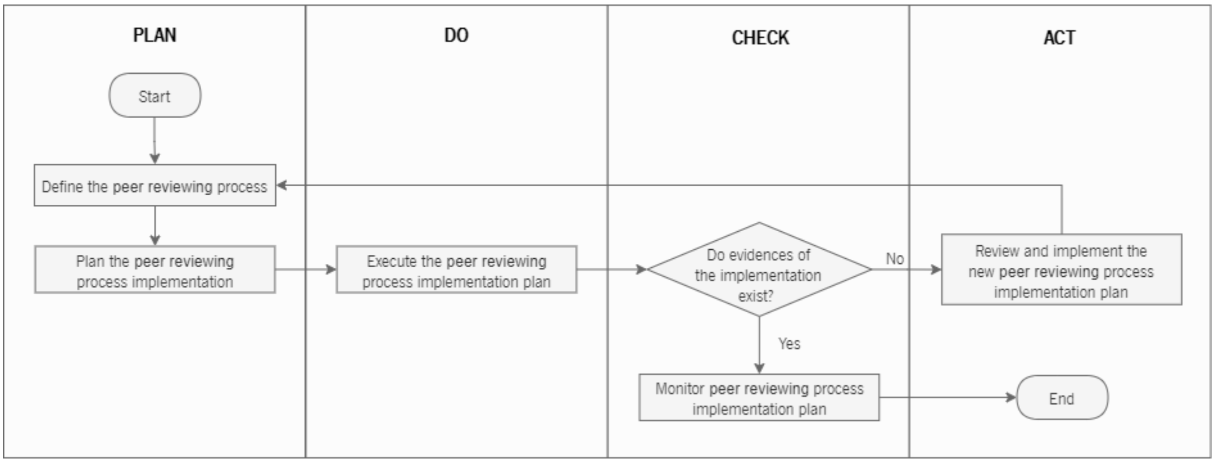

PLAN
DO
CHECK
ACT
Start
Define the peer reviewing process
Plan the peer reviewing process implementation
Execute the peer reviewing process implementation plan
Do evidences of the implementation exist?
No
Yes
Review and implement the new peer reviewing process implementation plan
Monitor peer reviewing process implementation plan
End

Figure 17 – Process Control of Peer Reviews.

In order to have further detailed information about the steps shown in the Figure 17, the steps *"Define the peer reviewing process"*, *"Plan the peer reviewing process implementation"*, *"Implement the peer reviewing process implementation plan"*, *"Monitor peer reviewing process implementation plan"*, and *"Review and implement the new peer reviewing process implementation plan"* will be explained in the Table 25, Table 26, Table 27, Table 28, and Table 29, respectively.

Table 25 – Information on the implementation of *"Define the peer reviewing process"* step.

"Define the peer reviewing process"	
Responsible	Developer(s)
Inputs	Peer Reviewing Practice Area of the CMMI
Objectives	Define a peer reviewing process.
Outputs	Peer reviewing process defined and updated
Implementation Details	1. Choose the Peer Reviews Practice Area of the CMMI. 2. Define a peer reviewing process for that PA: 2.1. Do a flow diagram of the process; 2.2. Describe all the steps of the process; 2.3. Define the inputs and outputs of the steps. 3. Peer reviewing process description. 4. Peer reviewing process approved by the PO.

Table 26 – Information on the implementation of *"Plan the peer reviewing process implementation"* step.

"Plan the peer reviewing process implementation"	
Responsible	Scrum Master
Inputs	Peer reviewing process defined
Objectives	Create a plan for the peer reviewing process implementation.
Outputs	Peer reviewing process implementation plan

"Plan the peer reviewing process implementation"	
Implementation Details	1. Peer reviewing process implementation plan description: 1.1. Meeting to introduce the peer reviewing process and the plan of the peer reviewing process implementation; 1.2. Which Program Increment and user story it will need to be implemented on; 1.3. Developer(s) who will implement the peer reviewing process.

Table 27 – Information on the implementation of *"Execute the peer reviewing process implementation"* step.

"Execute the peer reviewing process implementation"	
Responsible	Developer(s)
Inputs	Peer reviewing process implementation plan
Objectives	Implement the plan of the peer reviewing process implementation in the team.
Outputs	Peer reviewing process implemented
Implementation Details	1. Implement the peer reviewing process implementation: 1.1. Follow the peer reviewing process implementation plan; 1.2. Implement the peer reviewing process during the PI and in the specific user story.

Table 28 – Information on the implementation of *"Monitor peer reviewing process implementation plan completed"* step.

"Monitor peer reviewing process implementation plan"	
Responsible	Scrum Master
Inputs	Peer reviewing process implementation plan realised
Objectives	To achieve the peer reviewing process implementation plan and have the process implementation applied.
Outputs	Peer reviewing process implementation plan completed
Implementation Details	1. Analyse the peer reviewing process implementation plan: 1.1. Verify that the peer reviewing process implementation plan was complied to; 1.2. Verify that every step of the peer reviewing process implementation was achieved. 2. Designate the peer reviewing process implementation plan as completed.

Table 29 – Information on the implementation of *"Review and implement the new peer reviewing process implementation plan"* step.

"Review and implement the new peer reviewing process implementation plan"	
Responsible	Team
Inputs	New peer reviewing process and new peer reviewing process implementation plan realised
Objectives	Achieve a new peer reviewing process implementation plan and have the peer reviewing process implementation applied.
Outputs	New peer reviewing process created and peer reviewing process implementation plan created and completed
Implementation Details	Repeat the process creating a new version.

5.1.3 Results of the Implementation of the Method

During this section the results of the implementation of the previously presented method will be shown and supported through evidences, demonstrates that the method was implemented.

First Evidence

The first evidence considered the status of the team after the implementation of these processes, that is, the status of the implementation of the CMMI Levels in the PAs, CM and PR (Final Status – FS), compared with the Initial Status (IS). Furthermore, the evidence Bitbucket consists in a platform that provides a wide range of services, from code management (e.g., code versioning and peer review) to continuous integration and continuous delivery functionalities.

Therefore, after the new process was implemented in the team, the CM PA had the changes shown in Table 30.

Table 30 – Initial and Final Status of Implementation of the CMMI Levels in CM PA in the team.
Adapted from ISACA-CMMI Institute (2020)

IS	FS	Level	Description	E
		1		
🟨	🟩	CM 1.1	Perform version control.	Bitbucket
		2		
🟥	🟩	CM 2.1	Identify items to be placed under configuration management.	Bitbucket Docupedia
🟨	🟩	CM 2.2	Develop, keep updated, and use a configuration and change management system.	Bitbucket Docupedia
🟥	🟩	CM 2.3	Develop or release baselines for internal use or for delivery to the customer.	Bitbucket
🟥	🟩	CM 2.4	Manage changes to items under configuration management.	Bitbucket
🟥	🟩	CM 2.5	Develop, keep updated, and use records describing items under configuration management.	Bitbucket
🟥	🟥	CM 2.6	Perform configuration audits to maintain the integrity of configuration baselines, changes, and content of the configuration management system.	-

The team completed the implementation of the level 1 of the CM PA. The level 1 was already largely implemented, thus the team completed its implementation. As for the level 2, the team did not achieve its completion, although it completed the implementation of CM 2.2, which was previously only largely implemented, and the levels CM 2.1, CM 2.3, CM 2.4, and CM 2.5, which were not implemented, have been fully implemented. CM 2.6 suffered no changes, continuing to be not implemented.

In respect to the PR PA, the Table 31 shows the modifications in the levels that happened with the implementation of the process in the team.

Table 31 – Initial and Final Status of Implementation of the CMMI Levels in PR PA in the team.
Adapted from ISACA-CMMI Institute (2020)

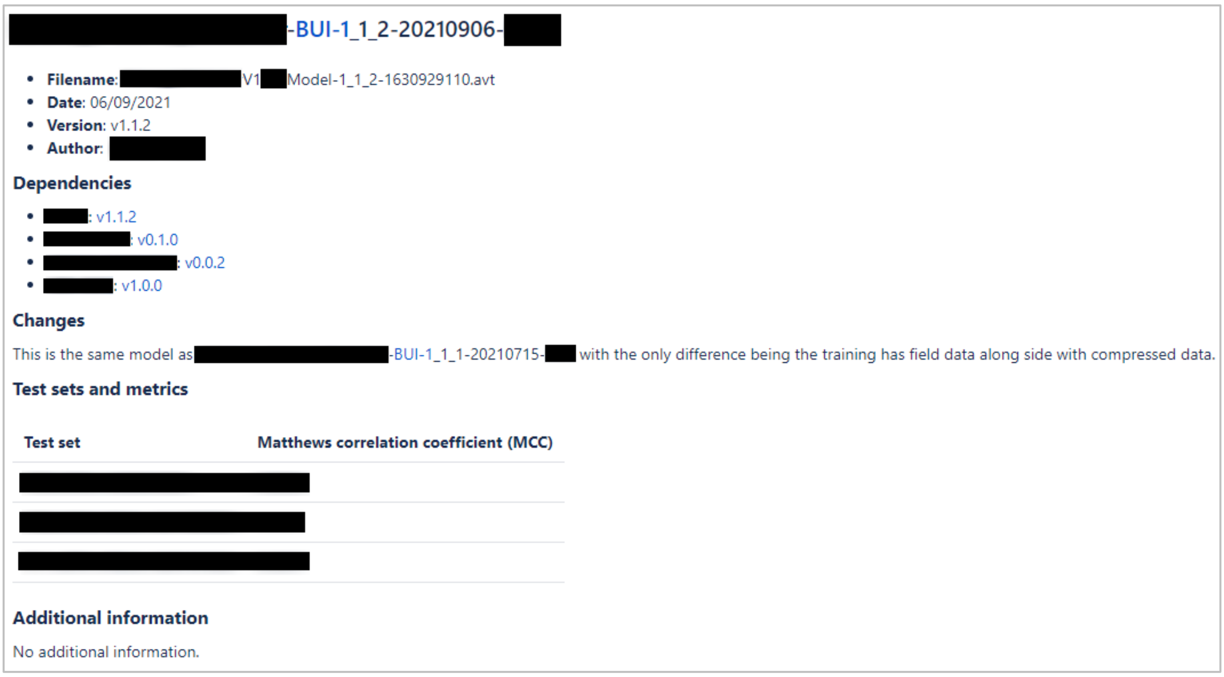
IS	FS	Level	Description	E
		1		
🟥	🟩	PR 1.1	Perform reviews of work products and record issues.	Bitbucket
		2		
🟥	🟩	PR 2.1	Develop and keep updated procedures and supporting materials used to prepare for and perform peer reviews.	Bitbucket Docupedia
🟥	🟩	PR 2.2	Select work products to be peer reviewed.	Bitbucket Docupedia
🟥	🟩	PR 2.3	Prepare and perform peer reviews on selected work products using established procedures.	Bitbucket
🟥	🟩	PR 2.4	Resolved issues identified in peer reviews.	Bitbucket

This PA, PR, had the most significant improvement, since all the levels were not implemented and, after the process implementation, all the levels went from not implemented to fully implemented. Therefore, this PA is implemented in the team up to level 2.

Second Evidence

The second evidence comes from a report from Bitbucket tool that shows the implementation of the CM and PR PA implemented in the team. The Docupedia report could not be retrieved due to company confidentiality restrictions.

Figure 18 shows several of the levels that are implemented, such as, CM 1.1, CM 2.2, CM 2.3, CM 2.4, and CM 2.5, and Table 32 summarizes this information for a better display of this evidence.



[redacted]-BUI-1_1_2-20210906-[redacted]

- **Filename:**[redacted]V1[redacted]Model-1_1_2-1630929110.avt
- **Date:** 06/09/2021
- **Version:** v1.1.2
- **Author:** [redacted]

Dependencies

- [redacted]: v1.1.2
- [redacted]: v0.1.0
- [redacted]: v0.0.2
- [redacted]: v1.0.0

Changes

This is the same model as[redacted]-BUI-1_1_1-20210715-[redacted] with the only difference being the training has field data along side with compressed data.

Test sets and metrics

Test set	Matthews correlation coefficient (MCC)
[redacted]	
[redacted]	
[redacted]	

Additional information

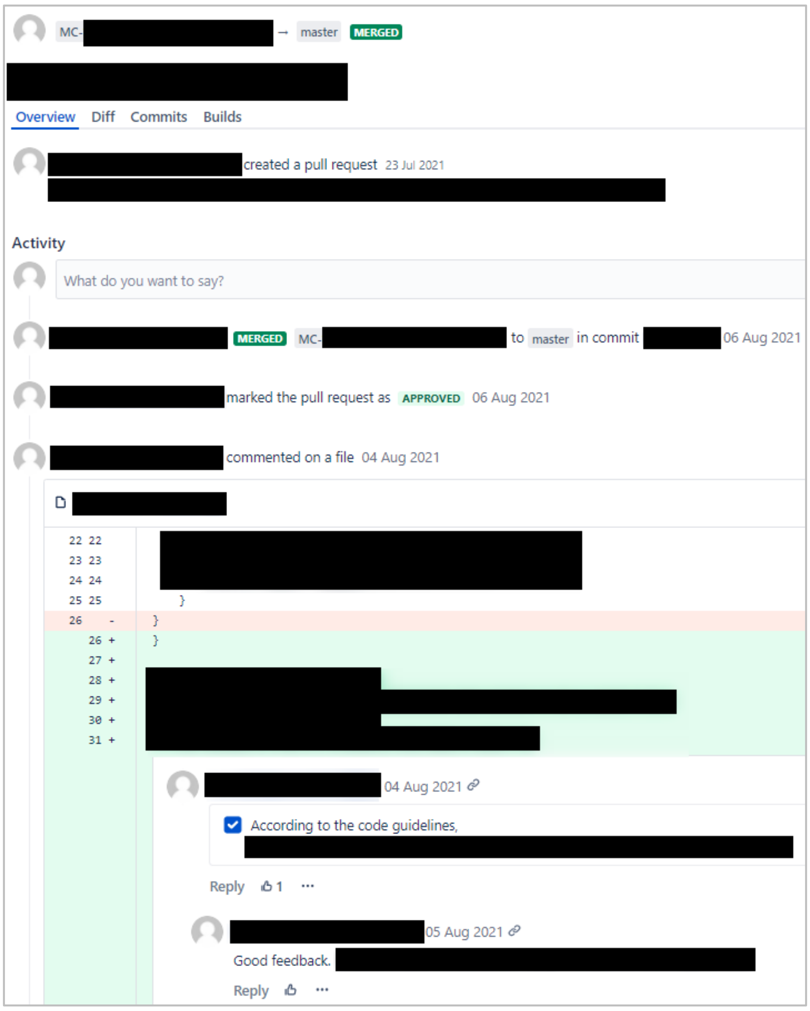
No additional information.

Figure 18 – Report of the implementation of the CM PA in the Bitbucket.
(Details from Bitbucket could not be shown due to company confidentiality.)

Table 32 – Outline of CM PA taking into consideration the report of Figure 18.

CM Levels	Process	Evidence 2	Comments
CM 1.1	Process Control of Configuration Management	Bitbucket (Figure 18)	Figure 18 shows one of the work products that is under versioning control, it is kept updated, and has its control versions, e.g., *"BUI-1_1_2-20210906"*, *"Version: v1.1.2"*, and where it is *"Dependencies"*.
CM 2.1	Process Control of Configuration Management	Docupedia Bitbucket (Figure 18)	The tool that helps the team to identify which are the items to be placed under configuration management is the Docupedia. Furthermore, the Figure 18 shows some characteristics, e.g.: author, date, and version.
CM 2.2	Process Control of Configuration Management	Docupedia Bitbucket (Figure 18)	The management system is developed and kept up to date in the Docupedia. That system is then used through the Bitbucket, as shown in Figure 18.
CM 2.3	Process Control of Configuration Management	Bitbucket (Figure 18)	The development or release baselines for internal use is identified in Figure 18, e.g., "*BUI*".
CM 2.4	Process Control of Configuration Management	Bitbucket (Figure 18)	The management of changes to items under configuration management are registered in Figure 18, e.g., "*Changes*".
CM 2.5	Process Control of Configuration Management	Bitbucket (Figure 18)	As Figure 18 shows, the configuration items are recorded in order to recover previous versions and to know the items' new status and the changes implemented. E.g., versions, dependencies, and changes.
CM 2.6	Process Control of Configuration Management	-	This level was not implemented in the team.

Figure 19 demonstrates most of the implemented levels, such as PR 1.1, PR 2.3, and PR 2.4, and Table 33 aggregates this information for a more comprehensive presentation of this evidence.



MC-[redacted] → master MERGED

[redacted]

Overview Diff Commits Builds

[redacted] created a pull request 23 Jul 2021

[redacted]

Activity

What do you want to say?

[redacted] MERGED MC-[redacted] to master in commit [redacted] 06 Aug 2021

[redacted] marked the pull request as APPROVED 06 Aug 2021

[redacted] commented on a file 04 Aug 2021

[redacted]

```
[redacted]
    }
-  }
+  }
[redacted]
```

[redacted] 04 Aug 2021

☑ According to the code guidelines, [redacted]

Reply 👍 1 ···

[redacted] 05 Aug 2021

Good feedback. [redacted]

Reply 👍 ···

Figure 19 – Report of the implementation of the PR PA in the Bitbucket.
(Details from Bitbucket could not be shown due to company confidentiality.)

Table 33 – Outline of PR PA taking into consideration the report of Figure 19.

PR Levels	Process	Evidence 2	Comments
PR 1.1	Process Control of Peer Reviews	Bitbucket (Figure 19)	As it is shown in Figure 19, reviews of the work products are performed and the issues are recorded.
PR 2.1	Process Control of Peer Reviews	Bitbucket Docupedia	This part was present in Docupedia and in another part of the Bitbucket that described procedures and gives support materials to be used when doing peer reviewing.
PR 2.2	Process Control of Peer Reviews	Bitbucket Docupedia	The selection of the work product was demonstrated in Docupedia. However, this report shows a work product that was selected to be peer reviewed and the process itself.
PR 2.3	Process Control of Peer Reviews	Bitbucket (Figure 19)	As the first comment in Figure 19 shows, *"According to the code guidelines"*, the team members prepare and perform peer reviews on the selected work products using predetermined procedures.
PR 2.4	Process Control of Peer Reviews	Bitbucket (Figure 19)	The second comment in Figure 19 demonstrates that when it was identified the issue, it was discussed and solved, therefore the team members resolve issues identified in peer reviews.

Third Evidence

The last evidence is the result from a survey done to the elements of the team where the study was developed and the process implemented, which can be found in Appendix II: Survey's Answers. The questions taken into consideration were the ones mentioned in the subchapter Techniques and procedures for data collection and analysis. In some of the answers, the Developers responded with "N.A." due to the fact that they did not interact with the process or that part of the process.

The Developers' answers to the CM PA were the following:

Question 1 – Is versioning control implemented on the work developed by the team?: *"Yes, by using GIT for versioning and OneNote for tracking. However, we have already a new tool to keep track of our results over time, that will improve the readability."*, *"Yes, but it can be improved (more examples, documentation, quick steps, etc.)."*, *"Yes, a version system has been implemented to models, datasets and repositories."*, *"Yes. All work is versioned via GIT."*, *"Yes. Either by Bitbucket (software) or Docupedia (documentation)."*, *"Yes, using tags in Bitbucket."*, and *"Yes."*.

Researcher Comments: These answers provide evidence that the Developers are aware that the versioning control was implemented and that they are familiar with the tools used in the level CM 1.1. The answers that mentioned *"GIT"* are referring to Bitbucket.

Question 2 – Do you know which are the items that go under configuration management (versioning)?: *"Yes. Code and model weights."*, *"Yes, they are models, datasets and repositories."*, *"Yes. Code, team libraries and models."*, *"Yes. Bitbucket and Docupedia."*, *"Yes, Code, datasets and Generated artifacts"*, *"Yes. Docupedia"* and *"No"*.

Researcher Comments: Most of the elements of the team are aware of the items that go under configuration management, also, some of them have a notion of where it is located. One of the team's members is not aware of where it is located or which are the items that go under configuration management.

Question 3 – Is it clear where the versioning process is defined?: *"Yes, we have code, data and model versioning. It is used to keep track of the changes and results delivered each sprint. Tools used for the effect: GIT."*, *"Yes, but it can be improved (more examples, documentation, quick steps, etc.)."*, *"Yes, it is available in the Docupedia, OneNote and README files of the respective repository."*, *"Yes. On a OneNote page."*, *"Yes. For software standard practices are used (master branch paired with development branch). For model weights, the process is defined in Docupedia. The process for versioning documents*

I don't think is very clear. When done through Docupedia, a new version is created at each edit.”, *“Yes, in Docupedia.”*, and *“Yes. Docupedia.”*.

Researcher Comments: As seen in the Developers’ answers, it is clear where the versioning process is defined and they are familiar with the process and the tools used for versioning process.

Question 4 – Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit?: *“Yes, we define the baseline tags internally specially for algorithm deployment, which in our case is tracked under sdd_weights.”*, *“Yes, but some normalization is needed.”*, *“Yes and Yes, the tagging system is well documented in the Docupedia.”*, *“Yes. There are instructions for the evolution of the version number, based on the changes made.”*, *“Yes. Internal build and released have different tag naming schemes.”*, *“Yes, tags containing BUI for internal; REL for release.”*, and *“Yes. Docupedia/Bitbucket.”*.

Researcher Comments: According to the Developers' statements, they know the baselines for internal use and the members responsible for it develop and release them.

Question 5 – Are the changes of the items placed under configuration management properly managed/updated?: *“Yes, even though it is not yet the ideal solution. As of now it is a little cumbersome, but we have already identified the issues and are fixing them.”*, *“Yes.”*, *“Yes, every change from last version is documented.”*, *“No. A lot of changes are not reflected in the version number.”*, “ *Yes. A log of the major changes is kept between versions.”*, *“Yes, through Bitbucket.”*, and *“Yes. Bitbucket.”*.

Researcher Comments: Most of the team considers that changes of the items placed under configuration management are properly kept up to date. One of the elements believes that not all changes are reflected in the configuration management.

Question 6 – Do you know where the changes and dependencies of the items under configuration management are described?: *“Yes, our dependencies are stored in a requirements file for each project.”*, *“No.”*, *“Yes, the respective README file contains all the information regarding changes.”*, *“No. I don't know the dependencies for any of the items except for the models where it is described in the readme.”*, *“Yes. For weights there is a readme file that tracks the major changes and dependencies. For code, each commit states the changes and dependencies.”*, *“Yes, Readme's in Bitbucket.”*, *“Yes. Bitbucket.”*.

Researcher Comments: The majority of the members know where the changes and dependencies of the items under configuration management are described, except two members that think they do not know where the changes and dependencies are described.

Question 7 – Is the configuration management system frequently audited?: *"Yes, we constantly iterate over the system, which requires us to constantly audit the code."*, *"N.A."*, *"No, the system isn't audited."*, *"No. It was never audited that I'm aware."*, *"Yes. The versioning system was recently overhauled to account for more recent needs. Both for software and for model weights."*, *"No. It is not."*, and *"Yes. Docupedia."*.

Researcher Comments: Three of the team members know that the configuration management system is not frequently audited, while three other elements think that the system is frequently audited. One of them decided to write that is not applicable to them.

In what concerns to the Questions 1, 2, 3, 4, 5, 6, they are associated to the levels CM 1.1, CM 2.1, CM 2.2, CM 2.3, CM 2.4, and CM 2.5, respectively, and all of them have been implemented. The Question 7 is related to the CM 2.6, and this one has not been implemented. Table 34 shows the summary of the answers of the survey to the CM PA.

Table 34 – Results from the survey done to the elements of the team for the CM PA.

Level	Question	Yes	No	N.A.
1				
CM 1.1	1. Is versioning control implemented on the work developed by the team?	7	-	-
2				
CM 2.1	2. Do you know which are the items that go under configuration management (versioning)?	6	1	-
CM 2.2	3. Is it clear where the versioning process is defined?	7	-	-
CM 2.3	4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit?	7	-	-
CM 2.4	5. Are the changes of the items placed under configuration management properly managed/updated?	6	1	-
CM 2.5	6. Do you know where the changes and dependencies of the items under configuration management are described?	5	2	-
CM 2.6	7. Is the configuration management system frequently audited?	3	3	1

In relation to the PR PA, the team elements' answered:

Question 1 – Are the work products reviewed? And are issues recorded when they are found?: *"Yes, but it can be improved."*, *"Yes, corrections, comments and suggestions are done on the bitbucket."*, *"Yes. New code is now reviewed by at least one team member before it goes to the master branch. Any issue found is reported in the review process."*, *"Yes, to both. Peer reviews are performed on Bitbucket. This platform allows to record improvements/issues on the work to be reviewed."*, *"Yes. Bitbucket."*, *"N.A."* and *"N.A."*.

Researcher Comments: The team members to which this question was applicable to answered that the work products are reviewed and the issues found are recorded. The remaining elements wrote that is not applicable to them.

Question 2 – Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated?: *"Yes, the procedure is on Docupedia."*, *"Yes, procedure is documented and up to date in the Docupedia."*, *"Yes. Docupedia."*, *"Yes. It's currently documented in Docupedia."*, *"N.A."*, *"N.A."* and *"N.A."*.

Researcher Comments: Almost of the team is aware that the procedures and supporting materials used to prepare and perform peer reviews are developed and kept updated and where that information is available. The other members answered that is not applicable to them.

Question 3 – Are the work products that undergo peer review properly selected?: *"Yes, on Bitbucket."*, *"Yes, Bitbucket handles it automatically."*, *"Yes. Docupedia."*, *"Yes. The major code repositories that directly interact with production are peer review."*, *"N.A."*, *"N.A."* and *"N.A."*.

Researcher Comments: As can be seen, the answers to the Question 3 are confirming that the work products that undergo peer review are properly selected and the members know where the process is made (Bitbucket) and where it is explained which are the work products to be selected (Docupedia). The other members answered that is not applicable to them. The other members responded that is not applicable to them.

Question 4 – Do you follow the established procedures when reviewing work products?: *"Yes. on Bitbucket."*, *"Yes, every step of the procedure is followed accordingly."*, *"Yes. Docupedia/Bitbucket."*, *"Yes. The review procedures are followed by all parties involved. Ticket creation, code development, code review, discussion, merge."*, *"Yes."*, *"N.A."* and *"N.A."*.

Researcher Comments: Most of the members do indeed follow the established procedures when reviewing work products. The remaining elements responded that is not applicable to them.

Question 5 – Does the team try to solve the issues identified in peer reviews?: *"Yes, as the issues can originate from improvements found to reasoning mistakes; it is always important to address them."*, *"Yes, any issue identified must be fixed before merging to master branch."*, *"Yes. Bitbucket."*, *"Yes. Every time an issue is raised the author either need to fix it or justify the decision not to fix it before the final merge."*, *"Yes."*, *"N.A."*, and *"N.A."*.

Researcher Comments: The team elements do try to solve the issues identified in the peer reviews, most of the time, they do in fact fix the issues, since it can lead to more problems when it merges with the master. The other members answered that is not applicable to them.

Regarding the Questions 1, 2, 3, 4, and 5, they are connected with the levels PR 1.1, PR 2.1, PR 2.2, PR 2.3, and PR 2.4, respectively, and all have been implemented. The overview of survey responses to the PR PA is shown in Table 35.

Table 35 – Results from the survey done to the elements of the team for the PR PA.

Level	Question	Yes	No	N.A.
1				
PR 1.1	1. Are the work products reviewed? And are issues recorded when they are found?	5	-	2
2				
PR 2.1	2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated?	4	-	3
PR 2.2	3. Are the work products that undergo peer review properly selected?	4	-	3
PR 2.3	4. Do you follow the established procedures when reviewing work products?	5	-	2
PR 2.4	5. Does the team try to solve the issues identified in peer reviews?	5	-	2

Evidences Overview

To summarize the evidences, the Table 36 was created, where it can be shown which evidence and which part of each evidence correspond to which level of the two PAs, CM and PR, of CMMI.

Table 36 – Compilation of the evidences.

Evolutionary Levels		Process	Evidence 1	Evidence 2	Evidence 3
CM PA	CM 1.1	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 1
	CM 2.1	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 2
	CM 2.2	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 3
	CM 2.3	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 4
	CM 2.4	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 5
	CM 2.5	Process Control of Configuration Management	Implemented	Figure 18	CM: Question 6
	CM 2.6	Process Control of Configuration Management	Not Implemented	-	CM: Question 7
PR PA	PR 1.1	Process Control of Peer Reviews	Implemented	Figure 19	PR: Question 1
	PR 2.1	Process Control of Peer Reviews	Implemented	-	PR: Question 2
	PR 2.2	Process Control of Peer Reviews	Implemented	-	PR: Question 3
	PR 2.3	Process Control of Peer Reviews	Implemented	Figure 19	PR: Question 4
	PR 2.4	Process Control of Peer Reviews	Implemented	Figure 19	PR: Question 5

In conclusion, is important to highlight that the team evolved in their engineering practices, and, in addition, they accomplished a higher evolutionary level in CM and PR PAs in an agile development context, in this case using practices of some of the Scrum ceremonies.

5.2 Harmonisation Method between Agile and CMMI

This subchapter aims to identify synergies between maturity models, in this case, CMMI up to level 2, and an agile methodology, Scrum, for a software development team in the automotive sector.

Due to the information provided in Identification of the Initial Status of the Case subchapter, it was possible to create a harmonisation method between agile and CMMI methodologies. However, it is important to emphasise that it can be complex to harmonise agile and CMMI, because there is no perfect match between agile and CMMI, as mentioned in the Capability Maturity Model Integration and Agile Methodologies subchapter. This means that, however hard it may be, agile methodologies can be adjusted to the CMMI methodology up to a certain level (level 2) and certain practice areas and levels. Furthermore, the Practice Areas considered for the harmonisation method were the areas mentioned in the Table 5 of the subchapter Identification of the Initial Status of the Case, which means to the RDM, PR, TS, EST, PLAN, MC, CM, and II PAs.

Therefore, take into consideration the Figure 20 to understand the caption for the output of the harmonisation method between agile and CMMI.

🟥 It does not assist 🟧 It assists very little 🟩 It assists substantially

Figure 20 – Stages of Implementation's Caption.

The method presented in Table 37 was created according to the information in the subchapter mentioned before and the Appendix I: Practice Areas Selected, through the researcher's analysis.

Table 37 – Harmonisation Method Between Scrum and CMMI.

Categories		Doing						Managing						Enabling		Improving	
Capability Area		ENQ				EDP		PMW						SI		SHP	
Practice Area		RDM		PR		TS		EST		PLAN		MC		CM		II	
Levels		1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Scrum	Product Backlog Grooming	🟧	🟧	🟥	🟥	🟥	🟥	🟩	🟧	🟩	🟧	🟥	🟥	🟥	🟥	🟥	🟥
	Sprint Planning	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟧	🟩	🟧	🟧	🟥	🟥	🟥	🟥	🟥
	Daily Meeting	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟧	🟥	🟩	🟧	🟥	🟥	🟥	🟥
	Sprint Review	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟧	🟧	🟥	🟥	🟥	🟥
	Sprint Retrospective	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

The Practice Areas PR, TS, CM and II are considered not assisted by Scrum methodology, since there are no Scrum ceremonies that are supporting these areas. However, the team can and, in this case, does perform those Practice Areas, but not due to the Scrum context. Nevertheless, in the PR and CM PAs, a strategy was adopted that consisted on having an acceptance criteria dedicated to peer reviews or configuration management in the user stories that involved peer reviewing or versioning in order to guarantee that the team was applying the process of those PAs.

Through the researcher's analysis, for the remaining PAs, RDM, EST, PLAN, and MC, there are some example activities that can be covered by some ceremonies of the Scrum methodology. Therefore, due to the information provided in the chapter Identification of the Initial Status of the Case and the Appendix I: Practice Areas Selected, it was considered the following outcomes for each PA mentioned above.

In relation to the RDM PA, it was considered that it supports very little throughout the Product Backlog Grooming ceremony of the Scrum, in both levels, due to this being the only event that was considered as associated to the example activities needed to complete the levels of this PA as referred in Table 38 of the Appendix I: Practice Areas Selected. The other ceremonies do not have any aspect that is dedicated to work the goals or example activities of this PA.

With regard to the EST PA, its stage of implementation was *"It assists substantially"* throughout the Product Backlog Grooming and Sprint Planning events of the Scrum methodology for the first level, considering that the example activity of the level 1 is done, in the team, during those ceremonies. The level 2 is assisted very little by the same ceremonies, because it is not in all the example activities, therefore the stage of implementation that was more adequate was *"It assists very little"*. The remaining ceremonies do not contribute toward the purposes or example activities of this PA.

In the matter of the PLAN PA, the implementation status for the Product Backlog Grooming and the Sprint Planning for this PA was the *"It assists substantially"* to the level 1, because those are the ceremonies needed, in the team, to complete the example activities mentioned in the Appendix I: Practice Areas Selected – Table 42 related to the PLAN PA. For the same level, it was considered that the ceremony Daily Meeting supports very little this PA. In relation to the level 2, the Product Backlog Grooming and the Sprint Planning events assist very little the PLAN PA. The subsequent ceremonies lack any feature dedicated to implementing the objectives or example activities of this PA.

Regarding the MC PA, the Sprint Planning and the Sprint Review of the Scrum methodology are supporting very little the level 1 of this PA, since they only cover some of the example activities revealed in Appendix I: Practice Areas Selected – Table 43. In addition, Daily Meeting is the ceremony that assists substantially this level of the MC PA. Related to the level 2, the ceremonies Sprint Planning and Sprint Review assist very little the example activities mentioned in Table 43 of the Appendix I: Practice Areas Selected. The remaining ceremonies cannot correspond to any of the characteristics toward the purposes or example activities of this PA.

In summary, even though there are synergies between the two models, it is possible to argue that Scrum ceremonies per se cannot ensure entirely the completion of the RDM, PR, TS, EST, PLAN, MC, CM, and II PAs of the CMMI model. Therefore, the Scrum ceremonies do not overlap with the engineering practices based on the maturity model CMMI. Thus, in order to fully complete the PAs levels, teams should find other models or strategies that can work alongside with Scrum to completely implement the CMMI's PAs.

6. Conclusions and Future Work

During this chapter the final considerations and the future work will be presented. It will start with the final considerations considering the identified problem, the goals considered, the outcomes obtained, and the main limitations faced in this study. Lastly, activities to be performed in the future are outlined.

6.1 Conclusions

During the execution of this project, several topics, in particular agile methodologies and CMMI, were examined in order to use action research to understand how a software development team uses and adapts to them. Following that, some recommendations were provided based on the theory and procedures observed, which helped to create advantages to the team under investigation.

As stated in the first chapter, the research question considered in this study was *"Is there any way to enhance the adoption of the CMMI model through agile methodologies in the context of a software development team?"*, with the purpose of achieving to the proposed goals:

G1 – The development of a control method for implementing good engineering practices in agile and CMMI Level 2 environments;

G2 – The introduction of a team control method – A way to understand if the control method is being fulfilled;

G3 – The identification of synergies between maturity models, CMMI until level 2, and agile methodology, Scrum, for software development teams in the automotive sector.

During this study it was possible to answer to the research question, where it was shown that there are ways to enhance the adoption of the CMMI model through agile methodologies, in this case Scrum, in the context of a software development team. As mentioned before, agile methodologies cannot cover all the levels and practice areas of the CMMI, but they can assist teams to adopt the CMMI model. This means that the Scrum ceremonies cannot overlap the good engineering practices based on the maturity model CMMI, however they can support some example activities of the Requirements Development and Management, Estimating, Planning, and Monitor and Control Practice Areas.

One of the main goals of this dissertation was to study the behaviour of good engineering practices based on the maturity model in agile development contexts, which means that it was needed to investigate if the use of agile methods could help teams to achieve good engineering practices and if it could work jointly with a maturity model. An agile environment was already present in the team and the

Scrum framework was successfully implemented. However, aspects such as good engineering practices were insufficient in efficiency and productivity.

During this project, a team quality control method that supported the team during the implementation of the PR and CM PAs was created, which means that it assisted them in other ways to improve their work and their compliance with the good engineering practices. Also, when in action, the strategy that was used, employed the acceptance criteria of a user story, oriented to the PR and CM PAs, that is associated to an agile environment, in this case, to the team that was using the Scrum methodology.

This interactive involvement of the researcher as the team's Scrum Master provided the opportunity to facilitate the adjustments needed, enabling the areas of improvement to be easier to be implemented. The team provided a positive environment to the adjustment of their work to the new processes. Their continuous feedback was significant to the development of the present research.

With the output taken from the subchapter Identification of the Initial Status of the Case, the comments done to the example activities of the PAs of the CMMI model enable to identify synergies between some of the PAs and the ceremonies of the Scrum methodology, which facilitated the creation of the harmonisation method between Scrum and CMMI methodologies. This method was built according to the performance of the team under study.

Several challenges arose during this research. For instance, firstly, the study only covers the team where the study was carried out, which implies that there are practice areas that would be out of reach to work due to not being the responsibility of the team, but of the project. And, lastly, due to the restricted schedule, it was necessary to reduce the number of desired practice areas to work on and prioritize those that would bring the greatest benefit to the team and, in relation to measuring the improvement of the team's efficiency, it was not possible to analyse due to the limited time for the development of this research.

6.2 Future Work

As it could be seen during the previous chapters, there is always room to improve, and the continuous improvement is an advantage. As future work, it is proposed that, as a continuation of the presented project, the study should be done on other software development teams in the practice areas that were not covered in this project. Also, it would be a challenge to study this subject in agile frameworks at upper levels, such as SAFe and upper levels of CMMI.

REFERENCES

Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, *6*(6), 828–837. https://doi.org/10.18517/ijaseit.6.6.1374

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. https://agilemanifesto.org/

Bennett, N., & Lemoine, G. J. (2014). *What VUCA Really Means for You*. https://hbr.org/2014/01/what-vuca-really-means-for-you

Bergmann, T., & Karwowski, W. (2019). Agile Project Management and Project Success A Literature Review. *Advances in Intelligent Systems and Computing*, *783*, 405–414. https://doi.org/10.1007/978-3-319-94709-9_39

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring - A systematic literature review. *Journal of Systems and Software*, *110*, 85–100. https://doi.org/10.1016/j.jss.2015.08.035

CMMI Institute. (2020). *CMMI Model V2.0 - Overview*. ISACA-CCMI Institute.

CMMI Product Team. (2010). *CMMI® for Development, Version 1.3*. Software Engineering Institute.

Cockburn, A., & Highsmith, J. (2001). Agile Software Development: The People Factor. *Computer*, *34*, 131–133.

Damelio, R. (2011). *The Basics of Process Mapping* (2nd ed.). Taylor & Francis Group.

Diaz, J., Garbajosa, J., & Calvo-Manzano, J. A. (2009). Mapping CMMI Level 2 to Scrum Practices: An Experience Report. *Communications in Computer and Information Science*, *42*, 93–104. https://doi.org/10.1007/978-3-642-04133-4_8

Fernandes, G., Ward, S., & Araújo, M. (2013). Identifying useful project management practices: A mixed methodology approach. *International Journal of Information Systems and Project Management*, *1*(4), 5–21. https://doi.org/10.12821/ijispm010401

Fritzsche, M., & Keil, P. (2007). Agile Methods and CMMI: Compatibility or Conflict? *E-Informatica Software Engineering Journal*, *1*(1).

Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). CMMI® or Agile: Why Not Embrace Both! In *Software Engineering Institut* (Issue November). http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8533

ISACA-CMMI Institute. (2020). *CMMI Model V2.0 - Overview*.

Knaster, R., & Leffingwell, D. (2020). *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework* (1st ed.). Addison-Wesley.

Layton, M. C., & Ostermiller, S. J. (2017). *Agile Project Management For Dummies* (2nd ed.). John Wiley & Sons, Inc.

Leffingwell, D., Jemilo, D., & Oren, I. (2021a). *Scaled Agile Framework - Core Values*. https://www.scaledagileframework.com/safe-core-values/

Leffingwell, D., Jemilo, D., & Oren, I. (2021b). *Scaled Agile Framework - Principles*. https://www.scaledagileframework.com/safe-lean-agile-principles/

Leffingwell, D., Jemilo, D., & Oren, I. (2021c). *Story - Scaled Agile Framework*. https://www.scaledagileframework.com/story/

Leffingwell, D., Knaster, R., Oren, I., & Jemilo, D. (2018). *SAFe 4.5 Reference Guide Scaled Agile Framework for Lean Software and Systems Engineering*. Addison-Wesley Professional.

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, *43*, 59–67. https://doi.org/10.1016/j.rcim.2015.12.001

Marçal, A. S. C., De Freitas, B. C. C., Furtado Soares, F. S., & Belchior, A. D. (2007). Mapping CMMI project management process areas to SCRUM practices. *Proceedings - International Conference on Software Engineering*, 13–22. https://doi.org/10.1109/SEW.2007.102

O'Brien, R. (1998). An overview of the methodological approach of action Research. Faculty of Information Studies. *University of Toronto*. http://www.web.ca/~robrien/papers/arfinal.html

Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (Sixth). Project Management Institute, Inc.

Rad, N. K., & Turley, F. (2013). *The Scrum Master Training Manual*. Management Plaza. http://mgmtplaza.com/latestversion

Ribeiro, F., Ferreira, A. L., Tereso, A., & Perrotta, D. (2018). Development of a grooming process for an agile software team in the automotive domain. *Advances in Intelligent Systems and Computing*, *745*, 887–896. https://doi.org/10.1007/978-3-319-77703-0_86

Robert Bosch GmbH. (2021). *Company | Bosch Global*. https://www.bosch.com/company/

Robert Bosch S.A. (2021a). *A nossa empresa | Bosch em Portugal*. https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal/

Robert Bosch S.A. (2021b). *Braga | Bosch em Portugal*. https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal/braga/

Rola, P., Kuchta, D., & Kopczyk, D. (2016). Conceptual model of working space for Agile (Scrum) project team. *Journal of Systems and Software*, *118*, 49–63. https://doi.org/10.1016/j.jss.2016.04.071

Royce, W. (1998). *Software Project Management: A Unified Framework* (6th ed.). Addison-Wesley.

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process* (First). Addison-Wesley.

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2019). Research Methods for Business Students. In *Qualitative Market Research: An International Journal* (Eighth). Pearson Education Limited.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org. https://scrumguides.org/scrum-guide.html

Scrum.org. (2021). *What is Scrum?* https://www.scrum.org/resources/what-is-scrum

Seymour, T., & Hussein, S. (2014). The History Of Project Management. *International Journal of Management & Information Systems (IJMIS)*, *18*(4), 233. https://doi.org/10.19030/ijmis.v18i4.8820

Sharp, A., & McDermott, P. (2009). Workflow Modeling Tools for Process Improvement and Application Development. In I. Artech House (Ed.), *Encyclopedia of Database Systems* (Second). https://doi.org/10.1007/978-0-387-39940-9_818

Software Engineering Institute. (2006). *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document. September.*

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education Limited.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. (2007). An exploratory study of why organizations do not adopt CMMI. *Journal of Systems and Software*, *80*, 883–895. https://doi.org/10.1016/j.jss.2006.09.008

Susman, G. I., & Evered, R. D. (1978). An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly*, *23*(4), 582. https://doi.org/10.2307/2392581

Swamidass, P. M. (2000). PLAN-DO-CHECK ACT (PDCA). In *Encyclopedia of Production and Manufacturing Management* (p. 523). Springer, Boston, MA. https://doi.org/https://doi.org/10.1007/1-4020-0612-8_689

Tesfay, Y. Y. (2021). Developing Structured Procedural and Methodological Engineering Designs. In *Developing Structured Procedural and Methodological Engineering Designs*. https://doi.org/10.1007/978-3-030-68402-0_8

Trienekens, O., Turetken Igor, & Stojanov Jos J. M. (2016). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and*

Process.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology*, *53*, 58–70. https://doi.org/10.1016/j.infsof.2010.08.004

Winter, M., Smith, C., Morris, P., & Cicmil, S. (2006). Directions for future research in project management: The main findings of a UK government-funded research network. *International Journal of Project Management*, *24*(8), 638–649. https://doi.org/10.1016/j.ijproman.2006.08.009

Appendix I: Practice Areas Selected

Requirements Development and Management Practice Area:

Table 38 – RDM PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Requirements Development and Management (RDM)	
RDM 1.1: Record requirements	
Example Activities	Comments
Record the requirements.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Example Work Products	Comments
Recorded requirements	It is done.
RDM 2.1: Elicit stakeholder needs, expectations, constraints, and interfaces or connections	
Example Activities	Comments
Elicit stakeholder needs, expectations, constraints, and interfaces or connections.	It is done in the project, although it is not done in the team.
Example Work Products	Comments
List of stakeholder needs, expectations, constraints	It is done in the project, although it is not done in the team.
List of interfaces or connections	It is done in the project, although it is not done in the team.
RDM 2.2: Transform stakeholder needs, expectations, constraints, and interfaces or connections into prioritized customer requirements	
Example Activities	Comments
Translate stakeholder needs, expectations, constraints, and interfaces or connections into recorded customer requirements.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Develop, record, and keep updated a prioritization of customer requirements.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Example Work Products	Comments
Prioritized customer requirements	It is not done.
Customer constraints	It is done on an ad hoc basis.
RDM 2.3: Develop an understanding with the requirements providers on the meaning of the requirements	
Example Activities	Comments
Develop criteria for identifying appropriate requirements providers.	It is not done.
Develop criteria for the evaluation and acceptance of requirements.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Analyze requirements to ensure that established criteria are met.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Reach an understanding of and obtain commitments to requirements with the requirements providers and the project participants.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Record needed changes to requirements.	It is not done in the team, although it should.
Example Work Products	Comments
List of appropriate requirements providers	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Criteria for evaluation and acceptance of requirements	It is not done in the team, although it should.
Results of analyses against criteria	It is not done in the team, although it should.
Recorded changes to requirements	It is not done in the team, although it should.

Requirements Development and Management (RDM)	
RDM 2.3: Develop an understanding with the requirements providers on the meaning of the requirements	
Example Work Products	Comments
Set of approved requirements	It is done on an ad hoc basis.
RDM 2.4: Obtain commitment from project participants that they can implement the requirements	
Example Activities	Comments
Assess the impact of requirements on existing commitments.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Negotiate and record commitments.	It is done on an ad hoc basis.
Example Work Products	Comments
Impact assessment	It is done on an ad hoc basis.
Recorded commitments that requirements can be met	It is done through SAFe in the PI Confidence Vote.
RDM 2.5: Develop, record, and maintain bidirectional traceability among requirements and activities or work products	
Example Activities	Comments
Develop, record, and keep updated bidirectional requirements traceability.	It is not done.
Example Work Products	Comments
Records of bidirectional requirements traceability	It is not done.
RDM 2.6: Ensure that plans and activities or work products remain consistent with requirements	
Example Activities	Comments
Review plans, activities, and projects for consistency with requirements and changes made to them.	It is not done.
Record inconsistencies and their sources.	It is not done.
Initiate and record any necessary corrective actions and communicate results to affected stakeholders.	It is done on an ad hoc basis through the Scrum process in the Product Backlog Grooming.
Example Work Products	Comments
Records of inconsistencies between requirements, plans, and work products	It is done on an ad hoc basis.
Corrective actions	It is done on an ad hoc basis.

Peer Reviews Practice Area:

Table 39 – PR PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Peer Reviews (PR)	
PR 1.1: Perform reviews of work products and record issues	
Example Activities	**Comments**
Review work products to identify issues.	It is not done.
Record results.	It is not done.
Example Work Products	**Comments**
List of issues from work product reviews	It is not done.
PR 2.1: Develop and keep updated procedures and supporting materials used to prepare for and perform peer reviews	
Example Activities	**Comments**
Record and keep updated peer review procedures.	It is not done.
Record and keep updated related supporting materials.	It is not done.
Example Work Products	**Comments**
Procedures for preparing for and performing peer reviews	It is not done.
Supporting materials	It is not done.
PR 2.2: Select work products to be peer reviewed	
Example Activities	**Comments**
Evaluate the criticality of the work product.	It is not done.
Determine and record the review type to use.	It is not done.
Example Work Products	**Comments**
Work product selection criteria	It is not done.
List of selected work products	It is not done.
PR 2.3: Prepare and perform peer reviews on selected work products using established procedures	
Example Activities	**Comments**
Develop schedule.	It is not done.
Follow procedures.	It is not done.
Record results from the peer reviews and the data from the process.	It is not done.
Communicate results to affected stakeholders.	It is not done.
Example Work Products	**Comments**
List of work products	It is not done.
Schedule	It is not done.
Peer review results	It is not done.
PR 2.4: Resolved issues identified in peer reviews	
Example Activities	**Comments**
Resolve issues.	It is not done.
Record resolutions and results and communicate to affected stakeholders.	It is not done.
Example Work Products	**Comments**
Resolution of issues	It is not done.
Results	It is not done.

Technical Solution Practice Area:

Table 40 – TS PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Technical Solution (TS)	
TS 1.1: Build a solution to meet requirements	
Example Activities	Comments
Build a solution.	It is done.
Example Work Products	Comments
Product or service	It is done.
TS 2.1: Design and build a solution to meet requirements	
Example Activities	Comments
Define the architecture.	It is done.
Identify, develop, or acquire effective design methods or tools for the solution.	It is done on an ad hoc basis.
Evaluate commercial off-the-shelf (COTS) products.	Not applicable.
Develop a preliminary design.	It is not done.
Develop a detailed design.	It is not done.
Track requirements against design to ensure they are satisfied.	It is not done.
Build the solution.	It is done.
Example Work Products	Comments
Architecture	It is done.
Component design	It is done.
Completed solution	It is done.
TS 2.2: Evaluate the design and address identified issues	
Example Activities	Comments
Determine what types of reviews to perform.	It is not done.
Identify review participants.	It is not done.
Send draft designs to reviewers.	It is not done.
Conduct a technical review.	It is not done.
Record decisions, issues, and concerns.	It is not done.
Identify potential fixes.	It is not done.
Communicate issues and decisions to affected stakeholders.	It is not done.
Update the design to address identified issues.	It is not done.
Review the solution.	It is not done.
Revise the component as necessary.	It is not done.
Example Work Products	Comments
Design evaluation issues	It is not done.
Design review meeting minutes	It is not done.
Updated design	It is not done.
Updated solution	It is not done.
TS 2.3: Provide guidance on use of the solution	
Example Activities	Comments
Develop and provide guidance materials.	Not applicable.
Example Work Products	Comments
Guidance material	Not applicable.

Estimating Practice Area:

Table 41 – EST PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Estimating (EST)	
EST 1.1: Develop high-level estimates to perform the work	
Example Activities	**Comments**
Review needs and assumptions and determine high-level estimates with stakeholders.	It is done in the Scrum events, in this case, Product Backlog Grooming and Sprint Planning.
Example Work Products	**Comments**
Rough order of magnitude estimate	It is done, in terms of complexity, using Epics and the Planning Poker in relation to the User Stories.
EST 2.1: Develop, keep updated, and use the scope of what is being estimated	
Example Activities	**Comments**
Review requirements and objectives with stakeholders to determine scope.	It is done through the SAFe events, in this particular case, in the PI Planning.
Example Work Products	**Comments**
List of tasks and activities or Work Breakdown Structure (WBS)	It is done through the Scrum process, in epics and user stories.
List of needed resources	It is not done.
Work flow diagram	It is done through the Scrum process.
EST 2.2: Develop and keep updated estimates for the size of the solution	
Example Activities	**Comments**
Use applicable methods to estimate the size and complexity of solutions and tasks.	It is done through Scrum Process in the Planning Poker, Product Backlog Grooming and Sprint Planning.
Example Work Products	**Comments**
Size estimate	It is done through Scrum Process in the Planning Poker, Product Backlog Grooming and Sprint Planning.
EST 2.3: Based on size estimates, develop and record effort, duration, and cost estimates and their rationale for the solution	
Example Activities	**Comments**
Collect and use historical data to develop, calibrate, or recalibrate models or methods to transform size and complexity into effort, duration, and cost estimates.	It is not done.
Describe and record the rationale for the estimates of effort, duration, and cost for the solution.	It is not done.
Include estimates of supporting infrastructure needs.	It is not done.
Example Work Products	**Comments**
Effort estimate	It is not done.
Duration estimate	It is not done.
Cost estimate	It is not done.
Estimating rationale	It is not done.

Planning Practice Area:

Table 42 – PLAN PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Planning (PLAN)	
PLAN 1.1: Develop a list of tasks	
Example Activities	**Comments**
Develop a task list.	It is done through Scrum Events, for instance, Product Backlog Grooming and Sprint Planning. Example: Jira Software and OneNote.
Review the task list with affected stakeholders.	-
Revise the list as needed.	It is done through Scrum Events, for instance, Product Backlog Grooming and Sprint Planning. Example: Jira Software.
Example Work Products	**Comments**
Task list	It is done. Example: Jira Software.
PLAN 1.2: Assign people to tasks	
Example Activities	**Comments**
Assign an individual who is responsible for each task.	It is done through Scrum Events, for instance, Product Backlog Grooming and Sprint Planning, it is discussed the user stories and who will be responsible for them.
Assign any additional people to the task.	It is done through Scrum events in the Product Backlog Grooming and the Sprint Planning allows to assign the story points of the user stories and the assistance needed.
Review assignments with the assigned individuals.	It is done through Scrum events in the Product Backlog Grooming and the Sprint Planning.
Record assignments in task list.	At team level: Scrum events, such as, the Daily Meeting and Product Backlog Grooming. At project level: Comments on JIRA issue between teams.
Example Work Products	**Comments**
Task list with assignments	It is done in the Jira Software allows the team elements assign to the user stories, tasks and sub-tasks.
PLAN 2.1: Develop and keep updated the approach for accomplishing the work	
Example Activities	**Comments**
Identify the objectives of the project.	It is done. Example: Docupedia.
Identify the approach to be used to achieve objectives.	It is done on an ad hoc basis.
Identify requirements.	-
Record business considerations.	-
Define and record the project lifecycle.	It is done through Scrum events.
Identify major resource needs and constraints.	It is done. Example: PI Excel.
Identify stakeholders.	It is available in the project, but it is not in the team.
Record agreements with stakeholders.	-
Identify risks.	It is not done in the team, although it should.
Identify safety and security approaches.	-
Review the project approach with affected stakeholders and obtain agreement.	At team level: It is done through Scrum process in the Sprint Planning, through the Jira Software. At project level: It is done through Scrum process in the PI Planning.
Revise the approach as necessary.	At team level: It is done through Scrum process in the Product Backlog Grooming and the Sprint Planning. At project level: It is done through SAFe in the ART Sync ceremony.

Planning (PLAN)	
PLAN 2.1: Develop and keep updated the approach for accomplishing the work	
Example Work Products	**Comments**
Recorded approach for accomplishing the objectives	-
Recorded project lifecycle	-
PLAN 2.2: Plan for the knowledge and skills needed to perform the work	
Example Activities	**Comments**
Identify the knowledge and skills needed to perform the work.	It is not done in the team, although it should.
Determine the gaps between the knowledge and skills needed versus those held by the currently assigned people.	It is not done in the team, although it should.
Select methods for providing needed knowledge and skills.	It is not done in the team, although it should.
Incorporate selected methods into the project plan.	It is not done in the team, although it should.
Example Work Products	**Comments**
Inventory of skill needs	It is not done in the team, although it should.
PLAN 2.3: Based on recorded estimates, develop, and keep the budget and schedule updated	
Example Activities	**Comments**
Identify major milestones.	It is done. Example: Roadmap.
Identify schedule assumptions.	-
Identify constraints.	It is done through user stories, epics or features.
Identify task dependencies.	It is done. Example: Roadmap PI.
Identify resources.	It is done on an ad hoc basis.
Identify and analyze risks.	It is done in the project, but it is not done in the team.
Based on estimates, develop the budget and schedule, and keep them updated.	It is done on an ad hoc basis.
Establish corrective action criteria.	It is not available.
Example Work Products	**Comments**
Budget	Out of scope.
Schedule	It is done. Example: PI Planning Excel.
Resource plan	It is available in the team resource, this means human and labour resources.
Budget and schedule risks	It is not available.
PLAN 2.4: Plan the involvement of identified stakeholders	
Example Activities	**Comments**
Develop a list of stakeholders.	It is done. Example: Docupedia.
Identify the involvement of each stakeholder.	It is done. Example: Teams' setup.
Record when the involvement is required.	It is done.
Example Work Products	**Comments**
Stakeholder involvement plan	It is done through the SAFe and Scrum Process.
Responsible, Accountable, Support, Consulted, Informed (RASCI) table	It is done on an ad hoc basis.
PLAN 2.5: Plan transition to operations and support	
Example Activities	**Comments**
Determine transition scope and objectives.	The team is not responsible for these features.
Determine transition requirements and criteria.	The team is not responsible for these features.
Determine approach to transition.	The team is not responsible for these features.
Develop schedule for transition.	The team is not responsible for these features.

Planning (PLAN)	
PLAN 2.5: Plan transition to operations and support	
Example Activities	**Comments**
Determine transition responsibilities and resources including post-transition support.	The team is not responsible for these features.
Determine operations and support training needs.	The team is not responsible for these features.
Example Work Products	**Comments**
Plans for transition to operations and support	It is available in the project, but it is not in the team.
PLAN 2.6: Ensure plans are feasible by reconciling available and estimated resources	
Example Activities	**Comments**
Perform resource leveling to adjust scheduling of tasks and resources.	It is done through Scrum Process in the Sprint Planning.
Ensure commitments are supported by adequate personnel or other required resources.	At team level: It is done through Scrum Process in the Planning Poker. At project level: It is done through SAFe in the Confidence Voting ceremony.
Negotiate commitments with affected stakeholders.	At team level: It is done through Scrum Process in the Sprint Planning. At project level: It is done through SAFe in the PI Planning ceremony.
Example Work Products	**Comments**
Revised plan and commitments	It is done. Example: Jira Software and PI Excel.
PLAN 2.7: Develop the project plan, ensure consistency among its elements, and keep it updated	
Example Activities	**Comments**
Record the project plan.	It is available in the project, but it is not in the team.
Review the project plan with affected stakeholders.	It is available in the project, but it is not in the team.
Revise the project plan as necessary.	It is available in the project, but it is not in the team.
Example Work Products	**Comments**
Overall project plan	It is available in the project, but it is not in the team.
PLAN 2.8: Review plans and obtain commitments from affected stakeholders	
Example Activities	**Comments**
Ensure individuals are involved in reviewing the work they are responsible for and the inputs that initiate the work.	It is available in the project, but it is not in the team.
Record commitments.	It is available in the project, but it is not in the team.
Review and approve project commitments.	It is available in the project, but it is not in the team.
Example Work Products	**Comments**
Results of plan reviews	It is available in the project, but it is not in the team.
Recorded commitments	It is available in the project, but it is not in the team.

Monitor and Control Practice Area:

Table 43 – MC PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Monitor and Control (MC)	
MC 1.1: Record task completions.	
Example Activities	Comments
Record task completion.	It is done through Scrum Process in the Daily Meeting.
Review updated task list with affected stakeholders.	It is done through Scrum Process in the Daily Meeting, Sprint Planning and Review.
Example Work Products	Comments
Task list	-
MC 1.2: Identify and resolve issues.	
Example Activities	Comments
Record the issue in the issue and action item list.	It is done through Scrum Process in the Daily Meeting.
Assign responsibility for resolving the issue or action item.	It is done through Scrum Process in the Sprint Planning and Daily Meetings.
Assign a due date.	It is done through Scrum process where the tasks and the user stories have established due dates.
Track issues and action items to closure.	It is done through Scrum Process in the Daily Meeting.
Example Work Products	Comments
Issues and action item list	-
MC 2.1: Track actual results against estimates for size, effort, schedule, resources, knowledge and skills, and budget	
Example Activities	Comments
Track actual results to plans and estimates.	It is done through Scrum Process in the Sprint Review (story points planned vs story points performed).
Monitor resources provided and used.	It is done at the level of human and material resources and structure.
Monitor the knowledge and skills of workgroup members.	It is not done.
Monitor commitments against those identified in the project plan.	It is done through Scrum Process in the Sprint Review. It is done through SAFe in the ART Sync.
Record significant differences in planned vs. actual values.	It is done through Scrum Process in the Sprint Review (story points planned vs story points performed).
Monitor progress against schedule.	It is done through Scrum Process in the Sprint Review.
Monitor expended effort and costs.	It is not done.
Example Work Products	Comments
Records of actuals versus estimates	It is done story points planned vs story points performed.
Records of significant deviations	It is done through Scrum Process in the Daily Meetings.
Records of status reviews	It is done through Scrum Process in the Sprint Review.
Corrective actions	It is not done.
Cost performance reports	It is not done.
Schedule performance reports	It is not done.
MC 2.2: Track the involvement of identified stakeholders and commitments	
Example Activities	Comments
Periodically review and record the status of stakeholder involvement.	It is done in the Daily Meeting through the methodology Scrum and the ART Meeting through SAFe.
Identify and record significant stakeholder issues.	It is done in the Daily Meeting through the methodology Scrum and the ART Meeting through SAFe.

Monitor and Control (MC)	
MC 2.2: Track the involvement of identified stakeholders and commitments	
Example Activities	**Comments**
Develop recommendations and coordinate actions to resolve issues.	It is done in the Daily Meeting through the methodology Scrum and the ART Meeting through SAFe.
Example Work Products	**Comments**
Records of stakeholder involvement	It is done through SAFe in the PI System Demo Slim.
Agendas and schedules for collaborative activities	It is done. Example: Docupedia.
Recommendations for resolving stakeholder issues	It is done in the project, although it is not done in the team.
Recorded issues	It is not done.
MC 2.3: Monitor the transition to operations and support	
Example Activities	**Comments**
Monitor the capabilities of operations and support to receive, store, use, and maintain new or modified solutions.	It is done on an ad hoc basis.
Monitor training delivery to stakeholders involved in receiving, storing, using, and maintaining solutions.	It is not the responsibility of this team. It is done by another team.
Review and analyze the results of transition activities.	It is done together with another team.
Example Work Products	**Comments**
Status reports of transition activities	It is not the responsibility of this team. It is done by another team.
Transition readiness report	It is not the responsibility of this team. It is done by another team.
Records of transition support reviews	It is not the responsibility of this team. It is done by another team.
Lessons learned report	It is not the responsibility of this team. It is done by another team.
MC 2.4: Take corrective actions when actual results differ significantly from planned results and manage to closure	
Example Activities	**Comments**
Collect issues for analysis.	It is not done.
Analyze issues to decide if corrective action is needed.	It is not done.
Take corrective action on identified issues.	It is not done.
Manage corrective actions to closure.	It is not done.
Example Work Products	**Comments**
List of issues requiring corrective action	It is not done.

Configuration Management Practice Area:

Table 44 – CM PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Configuration Management (CM)	
CM 1.1: Perform version control	
Example Activities	**Comments**
List the work products to be placed under version control and keep it updated.	It is done at a code level, but not at a documental or specification level.
Control versions.	It is done at a code level, but not at a documental or specification level.
Example Work Products	**Comments**
List of work products and their versions	It is not done.
CM 2.1: Identify items to be placed under configuration management	
Example Activities	**Comments**
Assign unique identifiers to configuration items.	It is not done.
Describe the important characteristics for each configuration item.	It is not done.
Specify when each item is placed under configuration management.	It is not done.
Example Work Products	**Comments**
Identified configuration items	It is not done.
CM 2.2: Develop, keep updated, and use a configuration and change management system	
Example Activities	**Comments**
Describe how the items and changes to them are controlled, used, and managed throughout the solution lifecycle.	It is not done.
Establish methods to manage multiple levels of control.	It is done at the code level.
Provide access control to ensure authorized access to the configuration management system.	It is done at the code level, in Bitbucket.
Store and retrieve configuration items in the configuration management system.	It is done at the code level.
Preserve the contents of the configuration management system.	It is done at the code level.
Update the configuration management system as necessary.	It is done at the code level, in Bitbucket.
Example Work Products	**Comments**
Configuration management system	It is done at the code level, in Bitbucket.
Change management system	It is not done.
Updated configuration items	It is not done.
CM 2.3: Develop or release baselines for internal use or for delivery to the customer	
Example Activities	**Comments**
Obtain authorization or approval before developing or releasing baselines of configuration items.	It is not done.
Develop or release baselines only from configuration items in the configuration management system.	It is not done.
Record the set of configuration items contained in a baseline.	It is not done.
Make the current set of baselines available.	It is not done.
Example Work Products	**Comments**
Authorization	It is not done.

Configuration Management (CM)	
CM 2.3: Develop or release baselines for internal use or for delivery to the customer	
Example Work Products	Comments
Baselines	It is not done.
CM 2.4: Manage changes to items under configuration management	
Example Activities	Comments
Initiate and record change requests.	It is not done.
Analyze the impact of change requests.	It is not done.
Categorize and prioritize change requests.	It is not done.
Review and get agreement on change requests to be addressed in the next baseline with affected stakeholders.	It is not done.
Track the status of change requests to closure.	It is not done.
Incorporate changes in a manner that maintains integrity.	It is not done.
Perform reviews or testing to ensure changes have not caused unintended impacts.	It is not done.
Record changes to configuration items and rationale.	It is not done.
Example Work Products	Comments
Change requests	It is not done.
Results of change impact analysis	It is not done.
Approval board records	It is not done.
Revision history of configuration items	It is not done.
Results of reviews or tests for unintended impacts	It is not done.
Revised work products and baselines	It is not done.
CM 2.5: Develop, keep updated, and use records describing items under configuration management	
Example Activities	Comments
Record configuration management actions in sufficient detail so the content and status of each configuration item is known and previous versions can be recovered.	It is not done.
Ensure that affected stakeholders have access to and knowledge of the configuration status of configuration items.	It is not done.
Specify the differences among previous, related, and latest versions of baselines.	It is not done.
Identify the version of configuration items that constitute a specific baseline.	It is not done.
Revise the status and history, e.g., changes, of each configuration item as necessary.	It is not done.
Example Work Products	Comments
Revision history or change log of configuration items	It is not done.
Change request records	It is not done.
Status of configuration items	It is not done.
Differences between baselines	It is not done.
CM 2.6: Perform configuration audits to maintain the integrity of configuration baselines, changes, and content of the configuration management system	
Example Activities	Comments
Assess the integrity of baselines and generate action items to address identified issues.	It is not done.
Confirm integrity of configuration management records.	It is not done.
Review the structure and integrity of items in the configuration management system.	It is not done.
Record action items and track them to closure.	It is not done.

<table>
<tr><th colspan="2">Configuration Management (CM)</th></tr>
<tr><th colspan="2">CM 2.6: Perform configuration audits to maintain the integrity of configuration baselines, changes, and content of the configuration management system</th></tr>
<tr><th>Example Work Products</th><th>Comments</th></tr>
<tr><td>Configuration audit or review results</td><td>It is not done.</td></tr>
<tr><td>Action items</td><td>It is not done.</td></tr>
</table>

Implementation Infrastructure Practice Area:

Table 45 – II PA Levels of the CMMI.
Adapted from ISACA-CMMI Institute (2020)

Implementation Infrastructure (II)	
II 1.1: Perform processes that address the intent of the Level 1 practices	
Example Activities	**Comments**
Perform processes.	It is done.
Example Work Products	**Comments**
Outputs of processes	It is done.
II 2.1: Provide sufficient resources, funding, and training for developing and performing processes	
Example Activities	**Comments**
Identify needed resources.	Not applicable.
Determine budget.	Not applicable.
Provide funding.	Not applicable.
Develop or buy tools.	Not applicable.
Develop training materials.	It is not done.
Provide training.	It is not done at the team level, but it takes place at the organizational level on the team members.
Example Work Products	**Comments**
Budget for resources	Not applicable.
Training materials	It is not done.
List of needed people, roles, and skills	It is not done.
Tools	It is done.
Training records	It is not done.
II 2.2: Develop and keep processes updated, and verify they are being followed	
Example Activities	**Comments**
Identify process purpose.	It is not done.
Determine process description format.	It is not done.
Describe and record processes.	It is not done.
Perform processes.	It is not done.
Verify that processes are being followed.	It is not done.
Review and update recorded processes with affected stakeholders.	It is not done.
Communicate and make recorded processes available.	It is not done.
Example Work Products	**Comments**
Recorded processes	It is not done.
Process verification results	It is not done.

APPENDIX II: SURVEY'S ANSWERS

Answers to the survey:

Developer 1:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes. All work is versioned via GIT.

2. Do you know which are the items that go under configuration management (versioning)? [CM 2.1] Yes/No. Justify, please.

Yes. Code, team libraries and models.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes. On a OneNote page.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes. There are instructions for the evolution of the version number, based on the changes made.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

No. A lot of changes are not reflected in the version number.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

No. I don't know the dependencies for any of the items except for the models where it is described in the readme.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

No. It was never audited that I'm aware.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

N.A.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

N.A.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

N.A.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

N.A.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

N.A.

Thank you very much for your collaboration! Your contribution is essential.

Developer 2:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes, but it can be improved (more examples, documentation, quick steps, etc.)

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

No

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes, but it can be improved (more examples, documentation, quick steps, etc.)

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes, but some normalization is needed

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

No

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

N.A.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

Yes, but it can be improved

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

N.A.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

N.A.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

Yes

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Yes

Thank you very much for your collaboration! Your contribution is essential.

Developer 3:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes. Either by Bitbucket (software) or Docupedia (documentation).

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

Yes. Bitbucket and Docupedia.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes. For software standard practices are used (master branch paired with development branch). For model weights, the process is defined in Docupedia. The process for versioning documents I don't think is very clear. When done through Docupedia, a new version is created at each edit.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes. Internal build and released have different tag naming schemes.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes. A log of the major changes is kept between versions.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

Yes. For weights there is a readme file that tracks the major changes and dependencies. For code, each commit states the changes and dependencies.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

Yes. The versioning system was recently overhauled to account for more recent needs. Both for software and for model weights.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

Yes. New code is now reviewed by at least one team member before it goes to the master branch. Any issue found is reported in the review process.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

Yes. It's currently documented in Docupedia.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

Yes. The major code repositories that directly interact with production are peer review.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

Yes. The review procedures are followed by all parties involved. Ticket creation, code development, code review, discussion, merge.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Yes. Every time an issue is raised the author either need to fix it or justify the decision not to fix it before the final merge.

Thank you very much for your collaboration! Your contribution is essential.

Developer 4:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes.

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

Yes. Docupedia.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes. Docupedia.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes. Docupedia/Bitbucket.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes. Bitbucket.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

Yes. Bitbucket.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

Yes. Docupedia.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

Yes. Bitbucket.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

Yes. Docupedia.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

Yes. Docupedia.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

Yes. Docupedia/Bitbucket.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Yes. Bitbucket.

Thank you very much for your collaboration! Your contribution is essential.

Developer 5:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes, a version system has been implemented to models, datasets and repositories.

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

Yes, they are models, datasets and repositories.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes, it is available in the Docupedia, OneNote and README files of the respective repository.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes and Yes, the tagging system is well documented in the Docupedia.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes, every change from last version is documented.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

Yes, the respective README file contains all the information regarding changes.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

No, the system isn't audited.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

Yes, corrections, comments and suggestions are done on the Bitbucket.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

Yes, procedure is documented and up to date in the Docupedia.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

Yes, Bitbucket handles it automatically.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

Yes, every step of the procedure is followed accordingly.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Yes, any issue identified must be fixed before merging to master branch.

Thank you very much for your collaboration! Your contribution is essential.

Developer 6:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes, using tags in Bitbucket.

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

Yes, Code, datasets and Generated artifacts.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes, in Docupedia.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes, tags containing BUI for internal; REL for release.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes, through Bitbucket.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

Yes, Readme's in Bitbucket.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

No. It is not.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

Yes, to both. Peer reviews are performed on Bitbucket. This platform allows to record improvements/issues on the work to be reviewed.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

Yes, the procedure is on Docupedia.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

Yes, on Bitbucket.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

Yes, on Bitbucket.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

Yes, as the issues can originate from improvements found to reasoning mistakes; it is always important to address them.

Thank you very much for your collaboration! Your contribution is essential.

Developer 7:

Configuration Management and Peer Reviews Survey

This survey was developed in order to gather evidence of the results of the implemented processes, versioning and peer reviewing, taking into consideration the Configuration Management and Peer Reviews Practice Area of the CMMI.

All answers will be confidential. The results of this research will be used to carry out the dissertation of the researcher, without ever mentioning information that could compromise the organization or the members involved. Therefore, to those who are answering this survey, no aspect about the work developed by the team needs to be mentioned, just the opinion about the implemented processes.

In case you did not have contact with the processes mentioned, please put N.A. (Not Applicable) in your answer.

It is expected that the respondent gives a full answer about the subject referred in the question.

I. Configuration Management Practice Area (Versioning)

1. Is versioning control implemented on the work developed by the team? Yes/No. Justify, please.

Yes, by using GIT for versioning and OneNote for tracking. However, we have already a new tool to keep track of our results over time, that will improve the readability.

2. Do you know which are the items that go under configuration management (versioning)? Yes/No. Justify, please.

Yes. Code and model weights.

3. Is it clear where the versioning process is defined? Yes/No. Justify, please.

Yes, we have code, data and model versioning. It is used to keep track of the changes and results delivered each sprint. Tools used for the effect: GIT.

4. Do you develop, release, or know the baselines (tags) for internal use? Are the baselines for internal use explicit? Yes/No. Justify, please.

Yes, we define the baseline tags internally specially for algorithm deployment, which in our case is tracked under sdd_weights.

5. Are the changes of the items placed under configuration management properly managed/updated? Yes/No. Justify, please.

Yes, even though it is not yet the ideal solution. As of now it is a little cumbersome, but we have already identified the issues and are fixing them.

6. Do you know where the changes and dependencies of the items under configuration management are described? Yes/No. Justify, please.

Yes, our dependencies are stored in a requirements file for each project.

7. Is the configuration management system frequently audited? Yes/No. Justify, please.

Yes, we constantly iterate over the system, which requires us to constantly audit the code.

II. Peer Reviews Practice Area

1. Are the work products reviewed? And are issues recorded when they are found? Yes/No. Justify, please.

N.A.

2. Are the procedures and supporting materials used to prepare and perform peer reviews developed and kept updated? Yes/No. Justify, please.

N.A.

3. Are the work products that undergo peer review properly selected? Yes/No. Justify, please.

N.A.

4. Do you follow the established procedures when reviewing work products? Yes/No. Justify, please.

N.A.

5. Does the team try to solve the issues identified in peer reviews? Yes/No. Justify, please.

N.A.

Thank you very much for your collaboration! Your contribution is essential.