Universidade do Minho
Escola de Engenharia

Helena Isabel Lopes Macedo

Development of a Continuous Improvement Process for Agile Software Development Teams

Master Dissertation

Integrated Master's Program in Industrial Engineering and Management

Work done under the guidance of

Professor José Dinís-Carvalho

October, 2019

Universidade do Minho
Escola de Engenharia

Helena Isabel Lopes Macedo

Development of a Continuous Improvement Process for Agile Software Development Teams

Master Dissertation

Integrated Master's Program in Industrial Engineering and Management

Work done under the guidance of

Professor José Dinís-Carvalho

October, 2019

AKNOWLEDGEMENTS

Foremost, I would like to express my deepest gratitude to my parents and my sister, for the huge support, love, motivation and for always believing in me during this research work. Without them I would've never made it. Thank you for all the sacrifices and for letting me follow my dreams, specially to my mom, this work is for her.

I would also like to thank my supervisor José Dinís-Carvalho for all the inspiration and privilege of working close to him. It was a great honor to work and study under his guidance. Thank you for all the opportunities, learnings and for always having trust in me. A special thanks to my supervisor at Bosch, André Leite Ferreira, for giving me the golden opportunity to make part of such a big project, and also helping me do a lot of Research and I came to know so many new things that I did not know before. Thank you for trusting me and my work since day one.

To the teams I had the honor to work with, Cloud, A&V, Algo, MobiBUS and DevOps Team, for letting me be part of them. Their commitment, involvement and proactivity were essential for the development of this dissertation project. Specially to Cloud team, Ariana and Ângela, I'll take with me high standards of teamwork, humor and support because of you guys. Also, to all my academic colleagues who have accompanied me over these years, Sara, Clara, Cris, Marcela, João Barrote, Rui Brito, Melissa, Mariana, Rafaela, I am really thankful for all the learnings and moments shared.

To my best friends for walking by my side all these years and for turning all my days better days. Lili, Margarida, Cathy, Marta, Ana, Moisés, I do not have the words to express my gratitude to have the opportunity to share my life with you. Thank you for all the moments of adventures, parties, caring and sharing.

Finally, I save a very special thanks to my best friend Catarina. For all the motivation, support and above all for all the love and sharing. I am really lucky to live the best years of my life by your side. Thank you for this final push, all the path to this point was insane, but we made it, girl.

Last but not least, I would like to express my sincere thanks to my boyfriend Nelson, for giving me daily inspiration to follow my dreams, no matter what. Thank you for showing me a new and tasteful world.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

The fast growth of the technology market has triggered the highest levels of competitiveness and globalization in the current business environment. The present VUCA (Volatile, Uncertain, Complex and Ambiguous) world and the continuous pursuit for new products and services has increased customer's demands, requiring the companies to be more flexible than ever by delivering software with quality, in time and within budget while in search of excellence and success.

To reach this purpose, many software organizations have been adopting Agile Methodologies combined with Continuous Improvement projects, in order to achieve an improvement in customer satisfaction, providing products with higher quality, lower costs and in shorter time to the market.

The ENG-P department at Bosch Braga, had a track record of some practices of Scrum but it was recognized that most teams were inefficient in taking advantage of the improvement opportunities that arose during the Sprint Retrospective ceremonies and transform them into improvements for the team.

The objective of this dissertation project was to design a process for the inception of Continuous Improvement (CI) in one team of the ENG-P department, that could define the actions that needed to be taken after the Retrospective ceremony, in order to make the most out of the suggestions given by the team. By combining the Scrum cycle and the Lean philosophy, this CI process helped the team analyze problems or improvement opportunities and proceed to their prioritization, planning, implementation and evaluation, making the team more effective in problem-solving, more motivated and, consequently, more productive. The first step was to define which CI strategy would be implemented. Between analytical and benchmark approaches, the PDCA was the one selected to serve as foundation for the CI process due to its application to improve processes, products or services in any organization.

After the implementation and standardization of the CI process in the daily work of the team, the final results showed that this new method doubled the suggestions implementation rate when compared with the previous results obtained by only performing the Retrospective ceremony. In addition, the accumulation of small improvements throughout time were responsible for substantial changes in the team. The level of motivation, involvement, empowerment and teamwork increased, there was a better visual task and time management, the communication improved, and a mindset of continuous improvement was built within the team.

KEYWORDS: Continuous Improvement, PDCA, Scrum, Sprint Retrospective, Software Teams

RESUMO

O rápido crescimento do mercado tecnológico despoletou os mais altos níveis de competitividade e globalização no atual ambiente de negócios. O presente mundo VUCA (volátil, incerto, complexo e ambíguo) e a procura contínua por novos produtos e serviços levou a um aumento das exigências dos clientes, obrigando as empresas a tornarem-se mais flexíveis e a fornecer *software* com qualidade, atempadamente e dentro do orçamento, enquanto procuram alcançar a excelência e o sucesso próprio.

Com foco neste objetivo, muitas empresas de desenvolvimento de *software* têm recentemente adotado metodologias ágeis em simbiose com projetos de melhoria contínua, com a finalidade de obterem uma maior satisfação do cliente, fornecendo produtos com maior qualidade e custos mais baixos.

O departamento ENG-P da Bosch Braga já possuía no seu histórico algumas práticas de *Scrum*, mas reconheceu-se que a maioria das equipas eram ineficientes no aproveitamento das oportunidades de melhoria que surgiam durante a cerimónia Retrospetiva e transformá-las em melhorias para a equipa.

O objetivo deste projeto de dissertação foi projetar um processo para a implementação de uma estratégia de Melhoria Contínua numa equipa do departamento ENG-P. Este definiu as ações a serem tomadas após a Retrospetiva, de forma a aproveitar ao máximo as sugestões dadas pela equipa. A combinação entre o ciclo *Scrum* e a filosofia *Lean* auxiliou a equipa na análise de problemas ou oportunidades de melhoria, na sua priorização, planeamento, implementação e avaliação. Estas ações permitiram que a equipa se tornasse mais motivada e eficaz na resolução de problemas e, consequentemente, mais produtiva. A primeira etapa consistiu na definição da estratégia de melhoria contínua a ser implementada. Entre abordagens de *benchmark* e analíticas, o ciclo PDCA foi a técnica selecionada como base ao processo devido à sua aplicabilidade na melhoraria de processos, produtos ou serviços.

Após a implementação e padronização do processo no trabalho diário da equipa, os resultados demostraram que a sua adoção duplicou a taxa de implementação de sugestões da equipa, quando comparado a resultados obtidos apenas com realização da Retrospetiva, aumentando a capacidade de resolução de problemas da equipa. Além disso, pequenas melhorias alcançadas ao longo do tempo trouxeram grandes mudanças para a equipa. O nível de motivação, envolvimento e *teamwork* aumentou, existiu uma melhor gestão visual das tarefas e do tempo, a comunicação melhorou e foi construída uma mentalidade de melhoria contínua dentro da equipa.

PALAVRAS-CHAVE: Melhoria Contínua, PDCA, Scrum, Sprint Retrospective, Equipas de Software

INDEX

List of Figures

LIST OF TABLES

List of Equations

List of Abbreviations and Acronyms

Algo	Algorithm
A&V	Activities and Violence
CI	Continuous Improvement
DMAIC	Define-Measure-Analyze-Improve-Control
DevTeam	Development Team
DSDM	Dynamic Software Development
IoT	Internet of Things
OPL	Open Point List
PDCA	Plan-Do-Check-Act
PO	Product Owner
ROI	Return of Investment
SAFe	Scaled Agile Framework
SM	Scrum Master
US	User Story
USA	United States of America
WIP	Work in Progress
XP	Extreme Programming

1. INTRODUCTION

1.1 Context

The global competition in the current market, resulting from quick technological changing and a continuous pursuit for new services and products, has turned Project Management more complex over the last decades, since the business environment is always changing. Since 1970 to onwards, project management has been intrinsically connected with software development projects. Starting from traditional to agile approaches, these two fields have been evolving together through history (Seymour & Hussein, 2016).

Over the last 30 years, billions of dollars have been invested in software development projects that failed. The problem is that with the current fast changing market towards shorter lead times and volatility in requirements, what was planned in the beginning of the project may not reach costumer's expectations in the end (Morien, 2005; Bergmann & Karwowski, 2019).

This increasingly competitive business environment has forced organizations to adopt more flexible working methods, able to embrace change more easily. In the area of software development, they are known as agile methods, and have become more popular since 2001 with the creation of Agile Manifesto. Over the last decade, the percentage of companies taking up some agile methodology increased from 14% to 65% (Dyba & Dingsøyr, 2008; Tripp, Armstrong, & J., 2018). These methods follow the 4 core principles of Agile Manifesto: Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation and Responding to change over following a plan (Beck et al., 2001).

The ability to deliver quality software, in time and within budget, will determine the success of a company. This goal requires the organizations to seek excellence, pursuing an improvement in costumer's satisfaction, providing services and products with higher quality, lower costs and shorter time to market. All these concepts arise together in 1990, in an attempt to define Lean, the name given by Jim Womack and Dan Jones, to the Toyota Production System. In the last decade the idea of combining lean principles to agile software development has been raising interest among authors, who support the implementation of a Continuous Improvement culture within software companies, resulting significant benefits like reduced lead time, improved flow and improved the defects fix rate (Kišš & Rossi, 2018).

1.1.1 Bosch Global

Robert Bosch, founded in 1886 the "Workshop for Precision Mechanics and Electrical Engineering" in Stuttgart, constructing and installing all types of electrical equipment, giving his first steps in the creation of the Bosch Group (Robert Bosch GmbH, 2019b). In 1887, Bosch manufactured his first magneto ignition devices, achieving with this product his first economic success. Ten years later, Bosch was the first to adapt it to a vehicle engine, becoming a global player in the market of automotive industry (Bosch Group, 2009).

Since the very beginning, Bosch is synonymous of innovation, strength and social commitment. Nowadays is one of the world's leading international providers of technology and services, but all this was achieved through its corporate responsibility based in compliance, sustainability, diversity and climate action.

We are Bosch is its mission statement, projecting us to its top objective of guarantee the company's future, generating strong and meaningful development, providing innovative, useful and exciting products and solutions to enhance quality of life – technology that is "Invented for life" (Robert Bosch GmbH, 2015). "*We are Bosch explains what drives us, what we have in common, and what we stand for: We want to leave a lasting trance in the world – achieved by a unique outstanding team."* (Denner, 2015).

In 2018, Bosh increased its headcount to 410.000 associates worldwide, generated 78.5 billion euros and invested 7.3 billion euros in research and development projects. Its operations are divided into four business sectors: Mobility Solutions, Industrial Technology, Consumer Goods, and Energy and Building Technology. Currently the business strategy of Bosch is to become a leader in Internet of Things (IoT), and it already offers to its costumer solutions for smart homes, smart cities, connected mobility and connected manufacturing (Robert Bosch GmbH, 2015).

Bosch Car Multimedia (CM) is a division from Bosch, part of the Mobility Solutions business area that supplies connectivity (V2X), cluster/display, and Human-Machine Interface (HMI) solutions for passenger cars, trucks, two-wheelers, and off-highway vehicles. Its main focus is become the integration of entertainment functions inside the car, navigation and drive assistance more flexible and efficient, shaping the future of connected mobility and become the vehicles the third living space, beside the home and workplace (Robert Bosch GmbH, 2019a).

The unit in Braga, Bosch Car Multimedia Portugal, S.A., with almost 2800 employees is responsible for the production of components that are parts for navigation and info-entertainment systems; instrumentation systems development of displays and control systems like steering angle sensors, high-end car radios for the automotive, electronic controllers for heating equipment, and controllers for

household appliances as well as development of innovative solutions in the area of the Human Machine Interface for the driver's cockpit (Robert Bosch GmbH, 2019c).

As part of Car Multimedia division, ENG Development Department aims to be a Development Center of reference for Bosch, promoting Innovation and passion for Engineering by developing excellent products. This dissertation project was developed in this department, with teams dedicated to work in the business area of Innovation, ENG-P and CM/PJ-IVS4 (Figure 1), developing new products and services (Robert Bosch GmbH, 2019a).

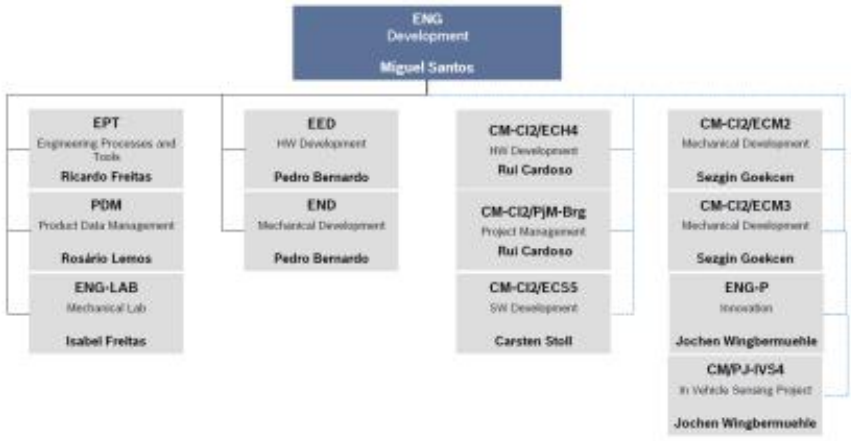

Figure 1 - Brg ENG Organization Chart

1.1.2 Challenge

This project focuses on the study and monitoring of two software development teams from the new product development department (Eng) of Bosch Car Multimedia Portugal, S.A. Since this automotive company is focused on technological innovation and aims to be an Internet of Thing (IoT) leader, besides being highly competitive, needs to be at the forefront of the market.

Scrum is currently one of the most used agile framework for software development teams around the world in order to manage and organize their work (Dyba˚ & Dingsøyr, 2008). Although the department already had some Scrum practices in its history, at the start of this project, these practices were nonexistent in the teams under study, so it turned out to be a priority to present this framework and introduce its practices into team's working method.

Authors such as Jeff Sutherland and Hugo Heitz, argue that virtually everything in Scrum can be explained by Lean concepts and that the implementation of this framework can be greatly improved by those who understand its philosophy. Lean is a way of delivering to the customer exactly what he wants, when he wants and where he wants, at the right time "just in time" (Sutherland & Heitz, 2011).

Lean is a way to reduce costs by solving problems, exposing practices that produce waste and empowering people by giving them the conditions to reach their full potential by solving increasingly more complex problems.

In Scrum cycle, this opportunity comes at the Retrospective ceremony. The Retrospective ceremony takes place at the end of each sprint, and as its name suggests, happens when the development team looks back over the last sprint and identifies opportunities for improvement. Each member of the development team recognizes what went well during the last sprint and what the team should keep, what didn't go so well and the team should improve, and then suggests creative ideas and solutions for the team to develop and apply.

In theory everything seems to work, but previous experience with this framework in the department has revealed that the retrospective ceremony has often become a "landfill of good ideas" (see Figure 2). The suggestions for improvement given by the team end up being stored in the record files waiting for someone to take the initiative to implement them, and most of the time that never happens, creating a pessimistic perception about this ceremony and the team starts to feel discouraged.

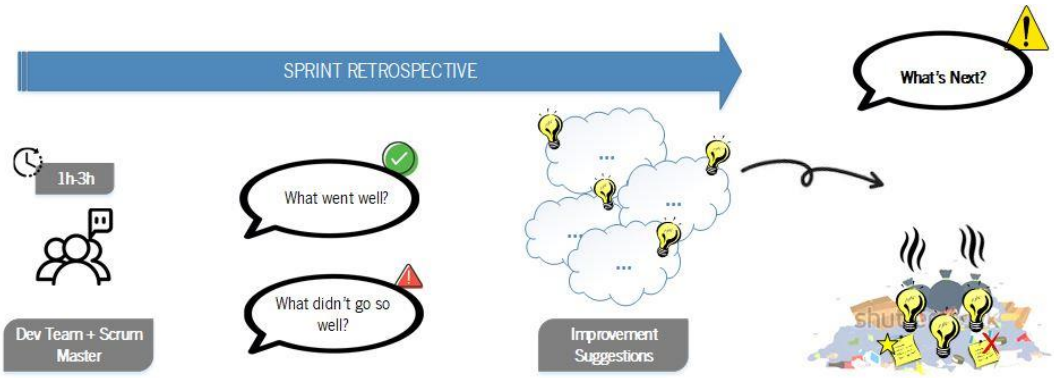

Figure 2 - Sprint Retrospective Problem Illustration

This shows a clear example that while Scrum can help the team improve their performance, it is not necessarily effective in solving all problems. If it were possible to design a process that would define the actions that the team should take after the Retrospective ceremony, and as Jeff Sutherland believes, combine this Scrum cycle with Lean philosophy, through a Continuous Improvement system capable of guiding/assisting the team in analyzing simple and complex problems, proceed to their prioritization, planning and implementation, the team would become more effective in problem solving, more mature and motivated and therefore more productive.

1.1.3 Goals

The main goal of this dissertation is to design a process for the inception of Continuous Improvement in ENG-P Department, that can be executed by software development teams that are simultaneously applying Scrum as their working method.

Therefore, the aim is to create the opportunity to obtain a continuous improved backlog, through a simple model that provides the right tools for addressing the need of continuous improvement. The CI process should be carried out together with the Scrum cycle, and facilitate the resolution of the obstacles identified at Retrospective Ceremony that are reducing team's velocity. To execute this CI process is essential for the team to engage the work and keep them motivated, once they are the responsible to bring creativity to the project, promote innovation and achieve improvement. It is also important to define how the team will contribute to this CI process, how it will be performed together with Scrum and how the team will benefit from it.

The work accomplished during this dissertation project intends to develop a CI behavior within the teams and encourage them to contribute to overall department success, by getting better teams at each sprint, through small improvements. In conclusion, creating innovation routines, reduce processes' waste and impediments, enhance team creativity and problem-solving potential are also targets to achieve during this project.

1.2 Research Methodology

This dissertation project began with a detailed literature review regarding the topics of Agile Methodologies, more specifically Scrum, Lean, Continuous Improvement and Software Process Improvement. During this phase, was conducted a research and analysis of scientific articles, books and dissertations related to these themes.

The action research methodology was the one chosen to adopt during this dissertation project. The concept "Action Research" was introduced by Kurt Lewin in 1946, describing a pioneer approach regarding social research, and as its name suggests, presupposes research work, combined with action by the researcher on the system under study, or in other words, acting (changing) and researching (understanding) at the same time (Susman & Evered, 1978). This methodology assumes that the researcher changes the system under study through actions taken over the time and then understand the results of those actions and produce knowledge. This approach implies a relationship of cooperation and collaboration between the researcher and the company where this project takes place.

Authors like Baskerville and Wood-Harper (1996) or Susman and Evered (1978), defend that the ideal example of action research must be described as a stages spiral, where each stage includes a cycle of five phases: Diagnosing, Action Planning, Taking Action, Evaluating and Specifying Learnings, promoting a continuous refinement of the process (See Figure 3).

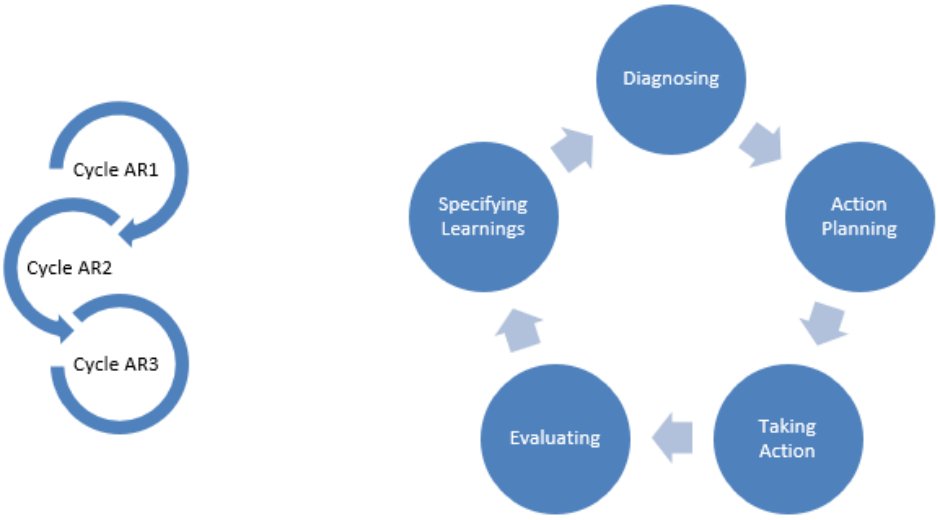

Figure 3 - Phases of Action Research Model (Adapted from Susman (1983))

However, projects that are developed according to the action research methodology may differ in the number of phases that will be addressed during the project, depending on the context and objectives of the project (Susman & Evered, 1978).

This dissertation will address only one complete cycle of Action Research with these five phases:

1. Diagnosing – Identifying or defining a problem. Initially, considering the project overview, was necessary to define the "state of agility" running in the company, realize what agile practices are currently being adopted by development teams in order to determine the goals of the effort. It was also important to describe the impact of the problems identified to the project and to the teams. In this phase the data was gathered through direct observation and consulting the existent documentation.

2. Action planning – Considering alternative courses of action for solving a problem. During this stage, a brainstorming was performed to find possible actions to address the problems identified in the previous phase. The definition of the events that should be implemented and when, template and guidelines creation, definition of new tools and processes to address scrum and continuous improvement needs were also done in this phase, with the purpose to attenuate the resistance to change.

3. Taking Action – Selecting a course of action. Before the next steps were performed, in case of more than one action was proposed a prioritization of the proposals was done, depending on the impact of the

proposed actions implementation. After the actions were prioritized and approved, they were executed and monitored in a defined time-frame.

4. Evaluating – Studying the consequences of an action. In this step an analysis of the achieved improvements was made, comparing the current state with the initial state. After this comparison was defined if each problem was actually solved or on the hand if it still could be improved.

5. Specifying Learning – Identifying general findings. In the final level of the action research cycle, the general findings regarding the main problem identified and its improvements are presented in section 4. Also were suggested some possible actions and research topics for future work in section 5.

This project counted with the participation of the researcher very close to the company, performing as Scrum Master for the teams in study. Although the researcher has accompanied four teams, only the work developed with two of them will be covered in this dissertation. This decision relies on the fact that both teams are working on innovation projects, are more open to receive change and adopt new practices, have the indicated size to work with Scrum framework and due to them project stability.

"How it is possible to technology companies, keep competitive and follow the market pace?", "How do we build a continuous improvement mindset in our organization?" and "Which challenges did the teams face during this CI implementation?" are the research questions that motivated this project and that will be answered during this dissertation project.

1.3 Adapted Action Research

As a matter of fact, the development of the continuous improvement process required a lot of effort once its implementation was incremental and gradual. The introduction of new practices in the routine of the teams was done in a smooth way, in order to sustain the system, reduce resistance to change and convert this association of small changes along time into the spirit "is the way we do things around here".

For that reason, the continuous improvement system gone through changes and was developed in a few iterations. When following the Action Research cycle, at the Evaluation phase was made an analysis of the consequences of the actions implemented, the benefits and problems were identified. In this specific case, did not seem appropriate go to the next stage of Specifying Learning or conduct a new Diagnosis once a new iteration would be executed right after. Then this interpretation done in the Evaluation phase produced the inputs necessary to brainstorm a new plan and follow up the cycle.

Hence, the development of the Continuous Improvement System ended up following one complete cycle of Action Research, but the iteration process only passed for 3 phases of the cycle: Action Planning, Taking Action and Evaluating (See Figure 4).

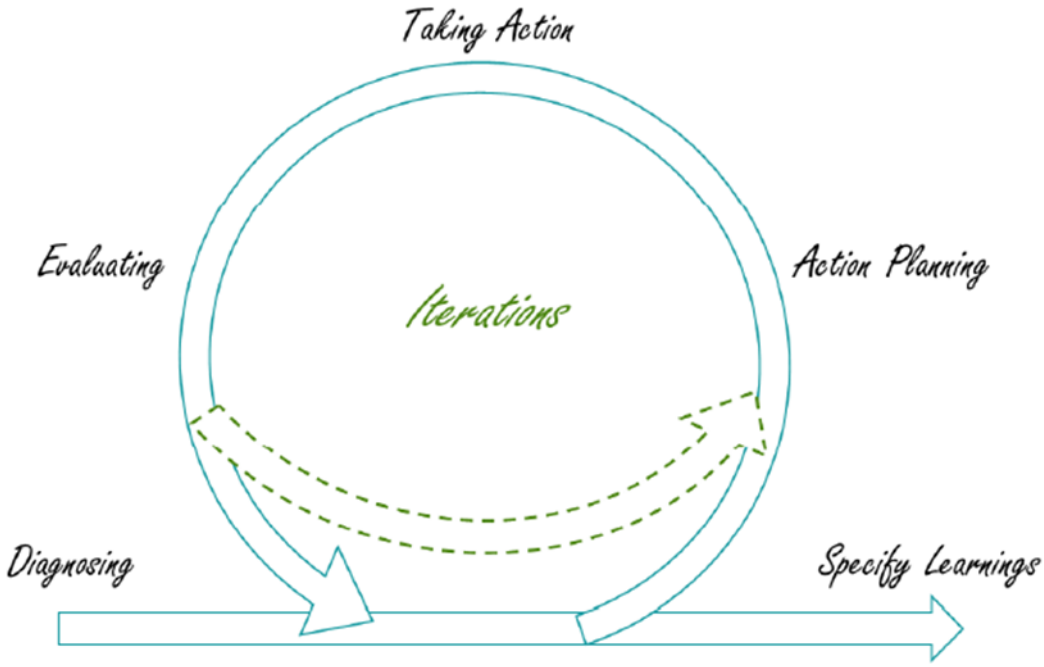

Figure 4 - Adapted Action Research Cycle from (Ribeiro, Tereso, Deborah, & Andrade, 2017)

1.4 Structure of Dissertation

This dissertation is framed in six chapters. Chapter 1 provides a background of the theme, defines the main challenge and the goals of this project and the research methodology used throughout this dissertation is presented. The second Chapter explores the scientific foundation of this work, presenting a literature review for the main knowledge areas involved, starting with the evolution of Project Management and its application in software industry. From Traditional to Agile approaches, Scrum is the one analyzed in a meticulous way once it is the framework used during this project. This chapter also covers research that addresses Continuous Improvement topic, benchmark and principle-based models like DMAIC and PDCA, the tools used and some success stories of its application in companies. Moreover, it is presented the last works done in software industry regarding Continuous Improvement implementation in software development procedures, like Software Process Improvement, CMMI and Lean Software Development, it is also mentioned some success cases of its implementation and its advantages and difficulties. The designing of the Continuous Improvement Process and its implementation its exhaustive described during the chapter 3. In this chapter is done an analysis of the current situation, and the problems existing in the department are defined. The iterations realized during

this project and the final propose can also be consulted over this chapter. Following, the results of the project are discussed in section 4. Here are shown survey's results and its interpretation as well as the opinion of team members regarding Scrum and Continuous Improvement Process implementation. Lastly, in section 5 are presented the conclusions of this dissertation project, and is suggested future work to keep improving teams' performance.

2. Literature Review

2.1 Project Management

In the last decades, Project Management have become increasingly more complex over the years. The fast growth of the technology market has triggered the highest levels of competitiveness and globalization. The present VUCA (Volatile, Uncertain, Complex and Ambiguous) world and the continuous pursuit for new products and services had increased costumer's demands, requiring the companies to be more flexible than ever to respond to these constant swift changes (Popova, Kryvoruchko, Shynkarenko, & Zéman, 2018).

Project Management is practiced as long as humans exist and has been applied in many different contexts over the years. According to Project Management Institute (PMI), "Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements." (Project Management Institute, 2017). There are many examples in history of great projects that were successfully completed, the Great Wall of China, the Pyramid of Giza or the Coliseum are good examples of projects that were developed many years ago, but their success has prevailed over time and is still visible today. The development of these projects required a big planning, a great management of the workforce involved, and a good management of the necessary resources, in order to accomplish the commander expectations. Although, the documentation existing regarding this historical projects is rare, and some authors defend that historical projects are very distant from the modern concepts of project management because the workforce was mainly from slavery, the budgets were unlimited and without an economic return and the projects were built over many years or even decades (Kozak-Holland, 2011).

It was only in the 1950s and 60s, that organizations started to apply systematic tools and techniques to complex projects. The U.S. Navy and the Manhattan project had contributed to the creation and documentation of principles of modern project management methodologies and techniques. Landing the man in the moon contributed to enlarge the knowledge regarding the management of projects with bigger scope and in the 1970s, with the technological explosion, emerged the field of software project management, with companies such as Oracle staying in the spotlight (Seymour & Hussein, 2016). In the last decades, project management has been extended for academic degrees and project management tools and techniques, from traditional to agile are spread around the world.

2.1.1 Traditional Methodologies

Traditional Methodologies has been used for several decades, and is characterized by a top-down model, where all decisions are established at the executive management level, flowing from the top to the bottom of organization's hierarchy. This approach is rather directed to the plan itself, once the whole plan is defined in the beginning of the project and the team should stick with it. Traditional project management also assumes that occurrences are predictable, the requirements are well defined in the initial phase of the project and that all tools and techniques are well understood. This approach is based on command, control, and hierarchy. It ignores uncertainty, defends an exhaustive planning and is resistant to change (Bergmann & Karwowski, 2019). From Code-fix, Waterfall, Rapid Application Development (RAD) to Spiral Development, many are the software development approaches that follow this path.

In today's market, where a highly demanding, dynamic, complex and technology-oriented environment is explored, this approach quickly meets its limitations. The lack of flexibility inherent in traditional project management is a disadvantage, customers often have difficulty setting all the requirements at the beginning of the project, it requires approval of heavy documentation and generates a lot of rework.

Along with, over the last 20 years, billions of dollars have been invested in software development projects that failed (Morien, 2005). Because of its limitations, the projects based on this approach cannot keep up with the competitive pace of the market, costumer's needs are not satisfied and at the end, the project fails.

a) Waterfall model

The Waterfall Model is a very popular traditional software development approach. This linear process is composed of five distinct sequential development phases, i.e., requirements analysis and definition, system and software design, implementation and unit testing, integration and system testing and operation and maintenance. The pure Waterfall Model defends that the project should only proceed to the next phase after a careful consideration of the output from the previous phase, however among the time, some stages started to be overlapped (Ming Huo, Verner, Liming Zhu, & Babar, 2004) (See Figure 5). Although, there is an agreement regarding the problems and limitations associated to it. This model does not deal well with change, once the requirements of the project should be well defined in the initial phase and the project should follow an inflexible plan, the hardware is usually selected in the early stages and the fast technology evolving can represent a threat, making the hardware obsolete, it requires approval of heavy documentation, generates a lot of rework and additionally the integration and testing the overall system is only performed at the end of the project. Because of its limitations, the projects

based on this approach cannot keep up with the competitive pace of the market, costumer's needs are not satisfied, and in the end, the project fails (Khalifa & Verner, 2000; Petersen, Wohlin, & Baca, 2007;Ming Huo et al., 2004).

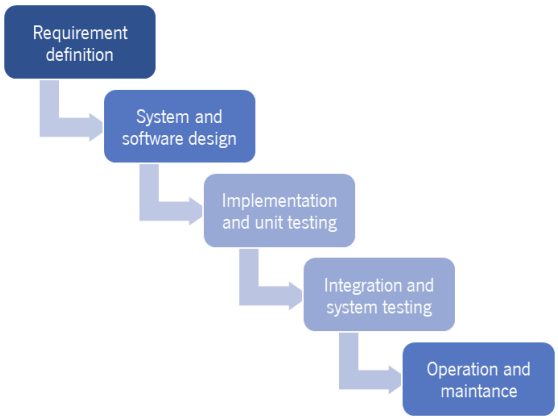

Figure 5 - Waterfall Model (Adapted from ((Ming Huo et al., 2004))

2.1.2 Agile Methodologies

Then a series of new methods known as agile development methodologies emerged, and claim to overcome the limitations of traditional projects. These approaches have on their core a culture of embracing change and empowering people. The planning is made incrementally and continuous throughout the project, the acceptance of new requirements during the project is easier, the evaluation of the return of investment is made in planning phases and assume that the project is a learning process. They also defend an active customers participation during the development of the project, rapid feedback to all the stakeholders and focuses on deliver working features as soon as possible. (Nerur, Mahapatra, & Mangalaraj, 2005).

Agile methodologies have become widely popular since the publication of the Agile Manifesto in 2001, with over 65% of companies taking up some agile methodology (Tripp et al., 2018). Software development methods such as Dynamic Systems Development Method (DSDM), eXtreme Programming (XP), Lean Software Development, Feature-Driven Development or Scrum fall into this category, and while they all claim to be called agile, they advocate a significantly different set of agility practices (See Table 1). However, all of these methodologies respect the fundamental principles of the Agile Manifesto (Dyba˚ & Dingsøyr, 2008):

- Individuals and Interactions Over Processes and Tools;
- Working Software Over Comprehensive Documentation;
- Customer Collaboration Over Contract Negotiation;

- Responding to Change Over Following a Plan.

Throughout the time is seen a growing tendency for dominance of a lean mindset, to minimize all unnecessary work and documentation, clients or stakeholders are no longer simply on the edge of the project, but actively monitor product, service or software development and there is an acceptance of the uncertainty associated with software development (Dingsøyr, Nerur, Balijepally, & Moe, 2012).

Table 1 - Description of the Main Agile Development Methods (Adapted by (Dyba˚ & Dingsøyr, 2008)*)*

Agile Method	Description
Dynamic software development (DSDM)	Divides projects in three phases: pre-project, project life-cycle, and post project. Nine principles underlie DSDM: user involvement, empowering the project team, frequent delivery, addressing current business needs, iterative and incremental development, allow for reversing changes, high-level scope being fixed before project starts, testing throughout the lifecycle, and efficient and effective communication.
Feature-driven development	Combines model-driven and agile development with emphasis on initial object model, division of work in features, and iterative design for each feature. Claims to be suitable for the development of critical systems. An iteration of a feature consists of two phases: design and development.
Lean software development	An adaptation of principles from lean production and, in particular, the Toyota production system to software development. Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.
Scrum	Focuses on project management in situations where it is difficult to plan ahead, where feedback loops constitute the core element. Software is developed by a self-organizing team in increments, called ‘‘sprints”, starting with planning and ending with a review. Features to be implemented in the system are registered in a backlog. Then, the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. One team member, the scrum master, is responsible for solving problems that stop the team from working effectively.
Extreme programming (XP; XP2)	Focuses on best practice for development. Consists of twelve practices: the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers, and coding standards. The revised ‘‘XP2” consists of the following ‘‘primary practices”: sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming, and incremental design. There are also 11 ‘‘corollary practices”.

However, according to Kalenda (2018), agile methodologies have been developed to be applied in the context of a single team. When dealing with larger projects, it is often necessary for multiple teams to collaborate effectively towards the same goal, generating the need to pass these agile practices to another dimension. Achieving and maintaining agility in large-scale contexts is a complex task with major challenges (Kalenda et al., 2018). According to Paasivaara and Lassenius (2014), the major challenges in projects with multiple interdependent teams include achieving coordination between multiple agile teams, addressing the lack of architectural planning and requirements analysis, as well as addressing adaptation issues with the organization. Despite these challenges, many companies have already chosen to adopt agile methods. Currently frameworks such as Scrum of Scrums (SoS), Scaled Agile Framework

(SAFe), Large-Scale Scrum (LeSS) or Lean Scalable Agility for Engineering (LeanSAFe) can be used to develop the scaling process in organizations, reaching a higher level of management (Kalenda et al., 2018).

2.1.3 Scrum

Scrum is the most popular agile methods. It is flexible, adaptative and iterative where the development team works as a unit to achieve a common goal. The term Scrum emerged in the 1980s when Takeuchi and Nonaka observed that successful teams at Honda exhibited similar behaviors to a rugby team. In rugby, this term is mentioned referring to the move that involves players packing closely together to gain the possession of the ball and move forward in the field (Sutherland & Heitz, 2011).

This framework defends a continuous planning, is based in self-organizing and empowered teams who divide their work into short work cycles called Sprints. Its incremental development through Sprints certifies that stakeholders give a rapid feedback regarding the developing product and that the team delivers the maximum business value as soon as possible. Scrum values ensure transparency in communication, commitment, focus courage and respect for others. This framework is based on a cycle of inspect and adapt, where team members periodically stop doing what they are doing, review what they have already done, and consider improving their work to become continuously more effective. To create these control points and manage the development team's tasks, this framework advocates the practice of Scrum activities such as Daily Stand-up Meetings, Sprint Planning, Sprint Review, Sprint Retrospective and Grooming Session (SCRUMstudy, 2017). Scrum principles can be applied to any type of project and must be adopted in order to guarantee a successful Scrum implementation (Figure 6).

Figure 6 - Scrum Principles adapted from ((SCRUMstudy, 2017)

The Scrum cycle is triggered by a meeting where the Stakeholders present to the Product Owner (PO) the requirements of the project and the Project Vision is settled. Then the PO starts writing the story telling of the project by translating these requirements into prioritized Epics (big User Stories), Features (functionalities) and User Stories (US) to be developed by the Development Team (DevTeam), creating the Product Backlog (PB). The PO must ensure that all the PB is clear and all the team understand it (Schwaber & Sutherland, 2017)(See Figure 7).

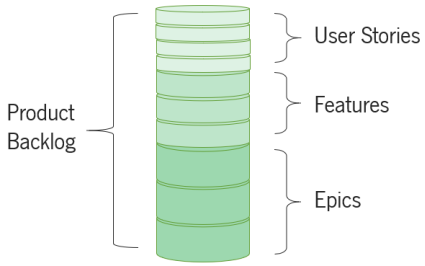

Figure 7 - Product Backlog Illustration

Then the first Scrum event, Sprint Planning occurs in the Scrum cycle. During this meeting all the Scrum team should be present, PO, Scrum Master (SM) and DevTeam. During the first part of this ceremony the PO shares with the team the initial Sprint Goal. The Sprint Goal is an objective to be achieved during the execution of the Sprint, by the implementation of a group of User Stories from the PB. The Sprint is a time-box of one month or less, during which a potentially releasable product Increment is developed. After the presentation of the Sprint Goal, the team negotiates with the PO and makes an agreement for the realistic Sprint Goal, with the entire team being committed to achieve that goal in the end of Sprint. Next the team estimates the complexity of each US and punctuates them. This punctuation usually is done using the Fibonacci sequence and the PO should not influence it. The punctuation is only assigned to the US when all the team members agree with it. If there is a disagreement, means that there is no a good understanding regarding the scope of the US, and the team members who gave the highest and the lowest punctuation should explain their point of view, and after that, the team makes a new voting. This process is repeated until the User Story's punctuation be consensual (See Figure 8)(SCRUMstudy, 2017).

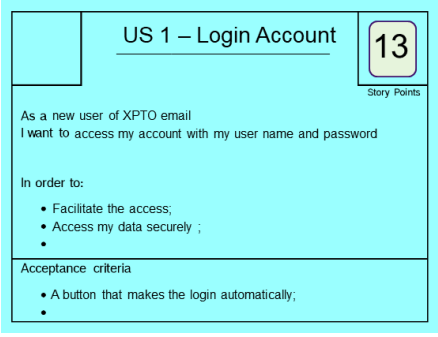

Figure 8 - User Story Example

The first part of the Sprint Planning ends when all the User Stories belonging to the realistic Sprint Goal are estimated. Following, the second part of this activity counts only with the presence of the DevTeam and the Scrum Master. The Scrum Master is a facilitator who ensures that the agile events takes place, the time-box and rules are respected and that everyone understands the meetings purpose and maintain the team's focus. During the second part, the DevTeam decides how they want to achieve each US and breaks them down into a set of tasks, creating the Sprint Backlog. Before the Sprint execution, the US and tasks committed are added to the Kanban board (see Figure 9).

Figure 9 - Scrum Cycle

The Kanban board is a Scrum artifact, used by the team during the daily meetings to track the progress of the User Stories and respective tasks throughout the Sprint execution (see Figure 10). During this meeting, the SM ask each developer "What have you done?", "What are you doing?" and "Is there anything blocking you?". These three questions synchronizes the team, allowing each developer to have the perception of what the other developers are doing, and if emerged some problem that is blocking the progress of a US, the SM is alerted and can work in the problem resolution (Sutherland, 2014b).

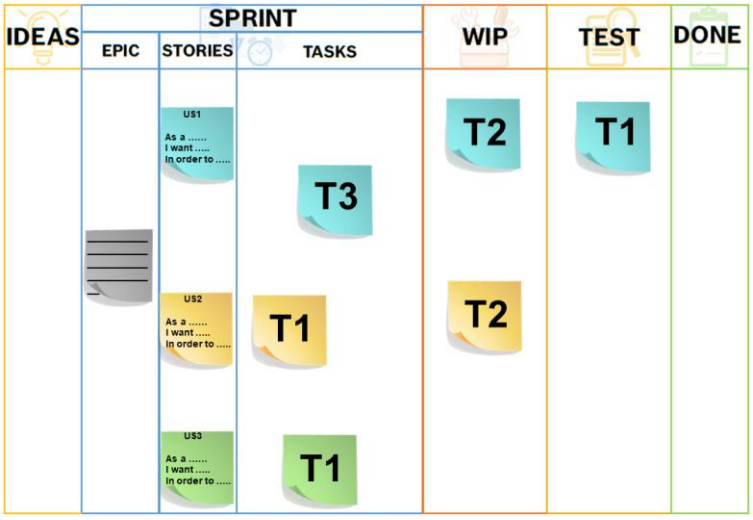

Figure 10 - Kanban Board

After the Sprint execution, occurs the Sprint Review. This meeting is Time-box to four hours for a one-month Sprint. During the Sprint Review the DevTeam presents to the PO the deliverables of the Sprint and he accepts or rejects the completed User Stories according to the compliance of the acceptance criteria. Besides that, the team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved. Then is considered what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning.

Finally, the Sprint Retrospective is the last ceremony of the Scrum cycle. This meeting is Time-box to four hours for a one-month Sprint. During this ceremony, the Scrum Team gets together and reflects about the last sprint. At the Sprint Retrospective meeting, each element should identify what went well during the Sprint and the team should keep, what did not go so well, and the team should stop doing, and finally search for opportunities for improvement regarding team's processes, tools, communication and other relevant topics to the project. Right after the conclusion of this Sprint, a new Sprint starts (Schwaber & Sutherland, 2017).

2.1.4 Sprint Retrospective

The Sprint Retrospective ceremony, as its name implies, aims to lead the Development Team through a reflection on what was done during the last sprint, what problems arose, how they were overcome, what went wrong, what went well and why, and what can be done differently next time to improve the process, and avoid blockages that might delay the team's work. The purpose of this meeting is to identify and prioritize improvement opportunities so that can be taken corrective actions, in order to achieve those improvements as quickly as possible by incorporating them into the product backlog at the next sprint (Sutherland, 2014a).

According to Vasanthapriyan (2018), this meeting will lead to continuous improvement of the entire team, by improving the productivity of every team member. Moreover, the retrospective ceremony is a key practice in continuous improvement, as it is a process of reflection on what has been done, and seeks to find ways to generate a better product for customers by delivering more value as soon as possible (Denning, 2010). The benefits derived from Sprint Retrospective are cumulative in nature. Each ceremony results in improvement actions that will be the basis for further improvement in the future. If a project fails, the Sprint Retrospectives will allow team members to learn from their mistakes and thereby overcome them. Its structure encourages discussion among team members about how they can achieve improvements, avoiding accusations or feelings of guilt among them. However, the retrospective ceremony not only addresses the improvement of the software development process, but also creates

opportunities for members of the development team to celebrate its success and recognize its merit (Kerth, 2001).

2.2 Lean Production

Lean is the name given to the Toyota Production System by Jim Womack and Dan Jones in their book "The Machine That Changed the World". Lean is as a way of "doing more with less", and can be seen as a methodology, a set of practices or tools or a business strategy, that aims to improve products' quality, reducing cost by identifying practices that create waste and deliver to the customer what he wats, when he wants just in time (Womack & Jones, 1996).

The Toyota Production System (TPS) emerged in Japan after the World War II, when the conventional ideas for the industrial development of the country were no longer viable. The lack of capital from customers and the lack of raw materials has led to a decrease in mass production efficiency. Therefore, Eiji Toyoda along with Taichii Ohno felt the need to improve their manufacturing process. Contrary to Ford's mass production system, that used wasteful batch production methods creating huge banks of work-in-process inventory throughout the value chain, Toyota did not have the luxury of creating waste, the factory space was reduced, it lacked warehouse and money. Moreover, they needed to produce small quantities of several models with high quality, low cost, short lead times and flexibility. Although, Ford's original idea of continuous flow was adopted, in order to create a system of one-piece flow that was flexible and adjustable to customer demand and efficient at the same time (Womack, Jones, & Roos, 1990).

Focused on the total elimination of waste the two core concepts of TPS are Just-in-Time (JIT) and jidoka. *Jidoka* relates to an "intelligent autonomation" or "autonomation with human touch", when an abnormal situation arises the machine is intelligent to stop and the worker will stop the production line, to prevent the defect from advancing in production. On the other hand, the concept JIT relates to a set of principles, techniques and tools that allows the company to produce and deliver small quantities of products, with shorter lead time, delivering to the client what he wants, at the right time in the right amounts (Jeffrey Liker, 2004). The concept JIT comes along with a pull system, where the preceding process mus always do what the subsequent process says, otherwise JIT won't work.

The TPS house, created by Fujio Cho, the president of Toyota, is a symbol modern production and considered a structural system consisting of principles and objectives. However, when Fujio Cho published The Toyota Way 2001 he changed the two pillars of lean to "Respect for people" and "Continuous Improvement" (See Figure 11).

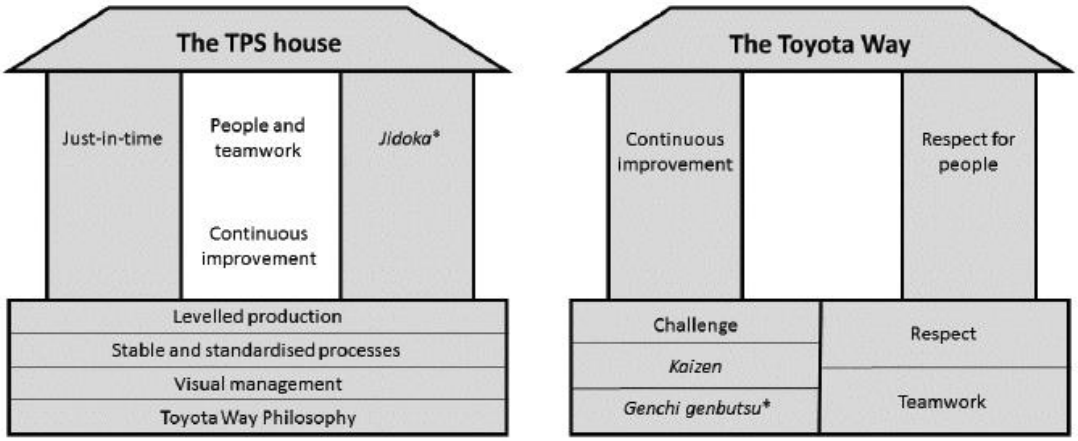

Figure 11 - Differences Between TPS House and Toyota Way adapted from (Jeffrey Liker, 2004)

In the Toyota Way, everything is about people: working, communicating, resolving issues, and growing together. It supports, encourages and promote employee involvement. They depend upon the workers to reduce inventory, identify hidden problems, and fix them and the workers have a sense of urgency, purpose, and teamwork (Liker, 2005).

2.2.1 Lean Wastes

Waste reduction is one of the premises of the TPS house. Waste is any activity that does not add value to the final customer and that he is not willing to pay (Ohno, 1998). Although, between wastes it is important to distinguish those which are "necessary", even though they don't bring value to the customers, they are essential to the productions process, and those which are "unnecessary", i.e., waste that adds nothing to the process and should be eliminated (Hines & Taylor, 2000).

Therefore, there are three types of activities, activities that create value, activities that do not create value but are necessary and activities that create no value and are unnecessary. According to Hines and Taylor (2000), usually only 5% of the activities in the companies are activities that add value. Within the activities that do not add value, 35% although do not adding value are necessary and 60% do not add any type of value. Ohno (1998), recognizes the following seven wastes:

- **Overproduction:** Waste from making more product than customers demand;
- **Waiting:** Waste from time spent waiting for the next process step to occur;
- **Transportation:** Wasted time, resources and costs when unnecessary moving products and materials;
- **Motion:** Wasted time and effort related to unnecessary movements by people;
- **Extra-Processing:** Wasted related to more work or higher quality than required;

- **Defects:** Waste from a product or service failure to meet customer expectations;
- **Inventory:** Wastes resulting from excess products and materials that are not processed.

Several authors, including Liker (2004), argue for the existence of an eighth waste, the unused human talent. The underutilization of the skills, talent and creativity of the different employees of the company, leads to missed opportunities for improvement and growth.

"Shortening lead time by eliminating waste in each step of a process leads to best quality and lowest cost, while improving safety and morale." (Womack, 1990)

2.2.2 Lean Thinking Principles

The success of Lean Production caught the attention of western companies, which began to adopt inherent practices of this model which have as foundation the continuous improvement of the organization through the continuous elimination of waste. This adaptation in other industrials sectors, not just by copying the techniques applied in Toyota is known as Lean Thinking and is based on the following five principles (See Figure 12) (Womack & Jones, 1996):

1. Specify Value: Define value from the customers perspective and express value in terms of a specific product;
2. Map the Value Stream: Map all the steps - value added, non-value added – that bring a product or service to the customer;
3. Establish flow: The continuous movement of products, services and information from end to end and through the process;
4. Implement Pull Based Production: Nothing is done by the upstream process until the downstream customer signals the need;
5. Work to Perfection: The complete elimination of waste so all the activities bring value to the customer, by continuous improvement.

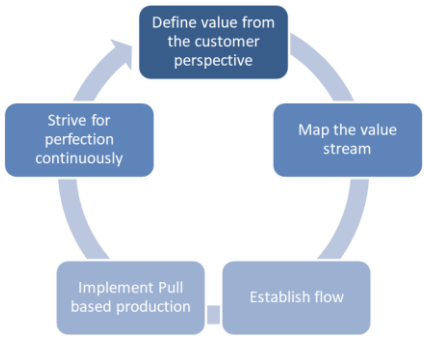

Figure 12 - Lean Thinking Principles (adapted from (Alves & Leão, 2015)

2.3 Continuous Improvement

Continuous Improvement (CI) or pursuing of excellence is one of the five principles of the lean thinking and is essential for companies that want to ensure their competitiveness in the market (Womack & Jones, 1996). Deming describe it as "improvement initiatives that increase successes and reduce failures" (W. E. Deming, 1993), while authors like Bessant, defines CI as " a company wide process of focused and continuous incremental innovation" (Bessant & Francis, 1999).

Currently, continuous improvement has become a "buzzword" in most organizations, but it represents different things to each sector of the companies. For the investors is to improve financial results, to the managers is to improve business results and production processes and to the operator is to improve the activities which they are involved (Bititci & Nudurupati, 2002). Although there is not an agreement between authors regarding a standard definition, all of them converge into a philosophy in an organization that intends to eliminate the waste in all processes through sustained improvement (Liker, 2004).

To be effective, CI requires the involvement of all employees at many levels. They are the people that best know the difficulties and the problems that arise in daily work, and their involvement and teamwork is essential so they can contribute effectively to the overall company's improvement (Kumar & Harms, 2004). Besides that, the company's management have the responsibility to encourage and support all the changes and improvement initiatives (Moica, Harea, & Marian, 2018). According to Dinis-Carvalho et al (2016), continuous improvement is a duty to all employees, and although there is no consensus on the best practices for getting employee engagement and involvement, the creation of a continuous improvement system with visual management, teams and daily meetings with a standardized schedule are some of the methods considered effective for achieving greater employee involvement.

The CIRCA (Continuous Improvement Research for Competitive Advantage) project based at the University of Brighton has developed a behavioral model describing the evolution of CI capability. This model provides a specification for the behaviors which need to be taken by an organization in order to enable CI capability. The gradual integration of key behaviors over time, will lead to an evolution of the performance improvements across the organization. Table 2 illustrates the key features of this model (Bessant & Francis, 1999).

Table 2 - Evolution of CI performance and practice from (Bessant & Francis, 1999)

Level	Performance	Practice
0= No CI activity	No impact from CI.	Problem-solving random; No formal efforts or structure; Occasional bursts punctuated by inactivity and non-participation; Dominant mode of problem-solving is by specialists Short-term benefits; No strategic impact.
1= Trying out the ideas	Minimal and local effects only; Some improvements in morale and motivation.	CI happens as a result of learning curve effects associated with a particular new product or process - and then fades out again. Or it results from a short-term input - a training intervention that leads to a small impact around those immediately concerned with it. These effects are often short-lived and very localised.
2= Structured and systematic CI	Local level effects; Measurable CI activity - e.g. number of participants, ideas produced, etc; Measurable performance effects confined to projects: Little or no 'bottom line' impact.	Formal attempts to create and sustain CI; Use of a formal problem-solving process; Use of participation; Training in basic CI tools; Structured idea management system; Recognition system; Often parallel system to operations; Can extend to cross-functional work but on an ad hoc basis.
3= Strategic CI	Policy deployment links local and project level activity to broader strategic goals; Monitoring and measurement drives improvement on these issues which can be measured in terms of impact on 'bottom line' - cost reductions, quality improvements, time savings, etc.	All of the above, plus formal deployment of strategic goals; Monitoring and measurement of CI against these goals; In -line system.
4= Autonomous innovation	Strategic benefits, including those from discontinuous, major innovations as well as incremental problem-solving.	All of the above, plus responsibility for echanisms, timing, etc., devolved to problem-solving unit; High levels of experimentation.
5= The learning organisation	Strategic innovation; Ability to deploy competence base to competitive advantage.	CI as the dominant way of life; Automatic capture and sharing of learning; Everyone actively involved in innovation process; Incremental and radical innovation.

2.3.1 Continuous Improvement Teams

Authors like Haug & Enz (2011), defend that the best way to directly involve the employees in the improvement process of the company is through the creation of work teams, by sharing common goals and performance targets. However, the performance success of the teams is influenced by two key critical factors: the autonomy and the decision-making opportunity (Gallie, Zhou, Felstead, & Geen, 2012). Self-organized teams can set their own goals, take individual initiatives at work, and they show being more

committed to the organization and more involved in the continuous improvement process, while non-self-organized teams, express a reduced level of satisfaction and the opportunity for workers to control their own work is reduced (Gallie et al., 2012).

2.3.2 Suggestion systems

Suggestion systems is an employee-oriented CI strategy and performance method, which leads the company to continuous improvements, and highlights the benefits of increasing employees' morale and involvement (Bengtsson, 2018). According to Marx (1995), the potential of the employees' ideas should not be neglected and he even argues that "idea is the most tremendous human force in the world". However, the support and investment from the top management is indispensable to guarantee the suggestion system success.

The world's biggest example of suggestion system, is certainly Toyota's suggestion system, which through fourty years logged more than twenty million ideas, and that it is not a coincidence that the company is the world's leading manufacturing firm (Moica et al., 2018).

2.3.3 Kaizen

The concept of continuous improvement comes from the Japanese word "kaizen" that means change for the better, and is the process to achieve cumulative improvements, no matter how small, in any context of life, whether personal or professional. In the industrial area, kaizen has become a world-wide continuous improvement philosophy, first adopted by the Japanese company Toyota (Imai, 1986). Because of Japan's success, the Kaizen philosophy has been adopted by organizations around the world as a way to improve performance while also improving employee safety and morale (Khan, 2011).

According to Imai (1986), this philosophy requires the commitment and involvement of the entire organization, starting with top management. Since Kaizen implies a change in organizational culture and a systematic search for both process and product improvements, all employees must be involved. Kaizen should be part of daily work as any other operational task, it should empower employees and encourage them to continuously question and improve the way they do their jobs, but top management must ensure that this is the direction the organization wants to take in order to Kaizen be successful implemented and effective in the long term (Imai, 1986). According to Al-baik & Miller (2016), Kaizen teaches individuals to challenge their problem-solving capability, to collect and analyze data and pushes the decision making to the workers.

Moreover, Imai (1986) highlights the difference between the improvements achieved through Kaizen and the innovative improvements, i.e., the small improvements achieved through incremental changes, resulted from little improvements in the daily-work basis at low costs and the innovative improvement that are associated to major changes, usually related to big investments in new equipment and technology. With Kaizen, all the small and incremental improvements accumulated throughout the time turn into major improvements, and will contribute to the improvement of the overall organization at the long term, as they are essentially based on eliminating waste. On the other hand, radical improvement is based on the purchase of technology that requires big investments and the company that invest on it, will only be at the forefront of its competitors while they do not have access to that technology, once they have it, it will lose the competitive advantage. Through kaizen, a company can improve its productivity and achieve an increasing performance without making huge investments, since it seeks improvements with the existing equipment and human resources (Liker, 2004).

2.3.4 PDCA

Deming also encouraged the Japanese to adopt a systematic approach to problem solving, firstly developed by Shewhart in 1939, which later in the 80s became popularized and known as Deming Cycle or Plan-Do-Check-Act (PDCA) Cycle, a cornerstone of continuous improvement (Rother, 2010)
The Plan-Do-Check-Act is a continuous improvement strategy originated in manufacturing area, with the purpose to improve production lines or products through the successive execution of PDCA cycles, obtaining a contribution of small improvements every day, that would become into a major improvement at long term (Singh & Singh, 2015).

This approach is composed for four steps, starting with Plan, where the current state is analyzed and a problem or opportunity for improvement is identified, consequently a plan to implement a corrective action or an improvement is created. In this step are also defined the targets wanted to achieve.

Next, in the Do phase the plan designed in the previous step is tested, ideally in a small scale with a controlled environment to validate the improvement.

In the Check step is done an observation of the improvements in order to proceed to their evaluation, and compare them with the predictions done in the Plan phase. If the plan did not result as expected, should be performed an analysis to understand the root-causes of the problem. Otherwise if the results were positive, the last phase Act should be carried on. In this step is made the decision to implement or not in large scale the improvements tested previously. Then the cycle goes on and is repeated consecutively (Schmidt, Elezi, & Lindemann, 2014).

In 1993, Deming adapted the PDCA Shewhart cycle to the PDSA cycle and called it *Shewhart cycle for learning and improvement*, as a "flow diagram for learning and improvement f a product or process" (See Figure 13) (Deming, 1993).

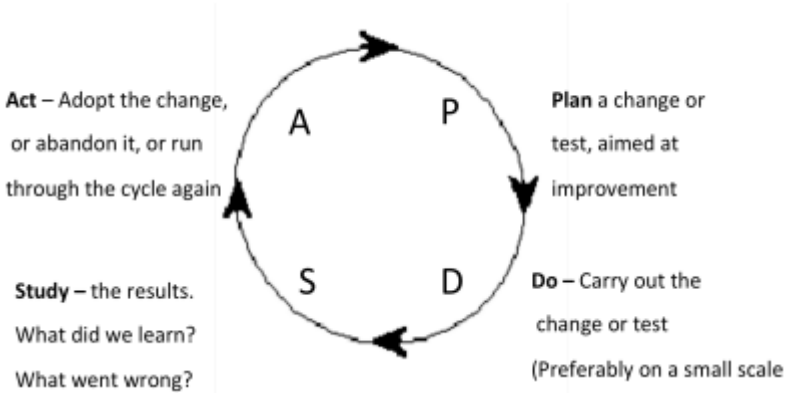

Figure 13 - PDSA Cycle from (Deming, 1993)

2.3.5 Six Sigma

Six Sigma became popular in the US by the introduction of this CI technique by the company Motorola inc., in 1986, when this organization was challenged to improve considerably the quality of its products. This technique was created with the purpose to measure the process quality using a statistical method control (Haddas et al., 2017).

According to Linderman (2003), Six Sigma is "an organized and systematic method for strategic process improvement and new product and service development that relies on statistical methods and the scientific method to make dramatic reductions in the customer defined defect rates".

DMAIC (Define-Measure-Analyze-Improve-Control) has been widely used for the implementation of Six Sigma projects, first in manufacturing companies, as it follows the principles of the PDCA cycle, but it also revealed to be effective in areas like finance, management and software development (Tonini & Spinola, 2006). This method is composed by five stages that should follow the next sequence:

- Definition: identification of problems and situations to improve, remarking what is critical to quality;
- Measurement: data gathering regarding the process, input variables and output defects, selection of critical characteristics to quality through necessary measures in the process;
- Analysis: identification an understanding of the root-causes that are responsible for the identified problems;

- Improvement: specification of processes characteristics to be improved;
- Control: procedures and rules to maintain the obtained improvements, documentation and follow-up of the new process conditions.

After the impressive improvements achieved by Motorola with the application of Six Sigma, this method was adopted by companies like Ford, Honda Sony, and GE for achieving excellence (Haddas et al., 2017).

2.4 Continuous Improvement in Software Development Context

The life as we know it today, has dramatically improved by the use of software and it is everywhere, whether on supporting the most simple services as electronical shopping to the software responsible for flying airplanes or performing remote medical surgery (Ferreira, 2016). Due to its vital role on today's society, software development companies have been pressed to constant look for ways to improve and stay in the forefront of their competitors (Tonini & Spinola, 2006). If a company wants to be successful, must learn how to monitor and evaluate its work in its dynamic environment. That is why over the last years, there is an increasing adoption of continuous improvement initiatives to the process of software development, known as Software Process Improvement. Software process improvement was inspired in the principles of the quality gurus like Deming, Juran and others with the attempt of "the pursuit of quality", believing that improving the management of the software development processes the software reliability and quality, costumer's satisfaction, business profitability and market share will also improve (Ferreira, 2016). Below are presented some software process improvement models (Halloran, 1999):

- The Capability Maturity Model, developed by the Software Engineering Institute (SEI) in theUSA;
- TRILLIUM, developed by Bell Canada in Canada;
- The Software Technology Diagnostic, developed by the Scottish Development Agency and enhanced by Compita in the UK;
- The Bootstrap method developed by the European ESPRIT project.

In fact, there are two approaches to software process improvements, the benchmark and the analytical approaches. The benchmark based approaches define the requirements or establish a set of practices adopted by the high performing organizations that intend to improve their software process, collect evidences of those practices and set the current state of the art in software engineering. The Capability

Maturity Model Integration for Development (CMMI-Dev) and ISO 12207 are two of the best practices of software process improvement (Chrissis, Konrad, & Shrum, 2011).

On the other hand, the analytical approaches mention that before initiating any improvement process, a quantitative evidence must be used to determine where improvement are needed, and evaluate if the improvement application met the expected results. The Shewhart PDCA cycle and the Quality Improvement Paradigm are examples of analytical software improvement approaches (Ferreira, 2016).

Moreover, in the last years, emerged a recent trend in software process improvement initiatives, of adopting multiple improvement models originating multi-model environments. The higher pressures and the competitiveness existing in the current market are the main reasons for organizations to adopt multiple improvement models, expecting that the combination of them will result in cumulative added value of each model into a single environment. According to Ferreira (2016), the combination of the CMMI-Dev with ISO9001 standard is the most common combination of benchmark software improvement models.

2.4.1 Benchmark Approaches

a) ISO 12207

The standard ISSO/IEC 12207 / IEE 12207 provides a common framework for software life cycle process, and definition of the base terminology to be used in the software industry. This standars aims to define a set of practices that facilitates communication with contractores, suppliers and other stakeholders during the software development of a product (International Standard, 2017).

b) ISO 9001

ISO 9001 support the adoption of a process to design and implement a quality management system in an organization, defining the requirements to be achieved by the overall systems driven to manage quality. The purpose of this standard is to provide the ability to continuously afford products that meet customer's needs and regulatory requirements to improve customer's satisfaction. ISO 9001 is applicable at any type or size organization since its focus is continuously improve processes (International Standard, 2017;Apcer, 2015).

c) CMMI for Development

CMMI model, introduced an approach that integrates several disciplines that characterize the business process, composed for the best practices for development and maintenance activities applied to product and services. CMMI foundation is that "quality of a system or product is highly influenced by the quality

of the process used to develop and maintain it", and it focus on improving the organization's process (Chrissis et al., 2011).

This model uses levels to define the evolution that the organizations can achieve in their processes of development and maintenance. Through these representations, a company can be flexible plan the evolution from ad-hoc and immature to disciplined and mature processes.

CMMI is organized by Process Areas (PA) namely: Process Management, Project Management, Engineering and Support. Each area describes goals and practices to be achieved and implemented, and Process Areas are arranged in a staged representation in levels from one to five. To achieve the level certification a predefined set of process areas must be implemented (Ferreira, 2016).

2.4.2 Analytical Approaches

a) PDCA

The Plan-Do-Check-Act cycle is an improvement strategy and its detailed description can be consulted in section 2.3.4

b) DMAIC-Six Sigma

Six sigma is a continuous process improvement approach, originated in Motorola but in the last years has been applied to software engineering, monitoring and controlling with the purpose of reducing process variation (Ferreira, 2016). A more detailed explanation of this model can be seen in section 2.3.5.

c) Total Quality Management

Total Quality Management aggregates all the analytical approaches mentioned above, in a system perspective on quality management. TQM provides a philosophical view along with a set of principles for organizations to achieve products with quality. The concept Total relates to the fact that everyone within the organization must be aware and work in the sense to produce improvements. The concept Quality is linked to customer's satisfaction and is achieved by meeting the customer's requirements at the lowest possible cost, with no loss or waste in resources (Crosby, 1979). Below are presented the TQM principles:

- A focus on preventions of problems rather than an acceptance to cope with then in a fire-fighting manner.
- Quality improvement must involve management.
- Everybody must be involved, from all levels across all functions.
- Every job must add value.

- There must be an emphasis on measurement to help to assess and to meet requirements and objectives.
- A culture of continuous improvement must be established (steady improvements and/or dramatic leaps forward).
- An emphasis in promoting creativity.

3. Designing the Continuous Improvement Process

This chapter describes and analyzes the initial conditions of the teams and their working methods: what practices were being held, what framework and tools were supporting them and how they organized, planned and managed their work. For two months the researcher observed the interaction inside and between teams, in order to gather information to understand the state of agility of each one of them.

Despite the researcher followed all the four teams in the innovation department, Algo Team, Activities & Violence (A&V) Team, Cloud Team and MobiBUS, the scope of this project is limited to two development teams, Algo and A&V Team, due to the team size and the nature of their projects.

During the second week of observation, the researcher was introduced to the teams as their future Scrum Master, in order to "break some ice" and they could interact to each other as soon as possible. During this observation period, the main problems were identified and described in the Diagnosis phase (3.1). Next, in section 3.2 the steps needed to achieve the continuous improvement process that can be executed together with the Scrum cycle are detailed as iterations, including the following phases: Action Planning, Taking Action and Evaluating (See Figure 14). Then, the Standardized Process of Continuous Improvement can be consulted in section 3.3, as well as the guides to apply the Tools and Artifacts to support its utilization. Lastly, the resolution of communication problems emerged in the interaction between the Algo team and the other collaborative teams are discussed together with the development of Sprint Retrospective techniques, in the section 3.4.

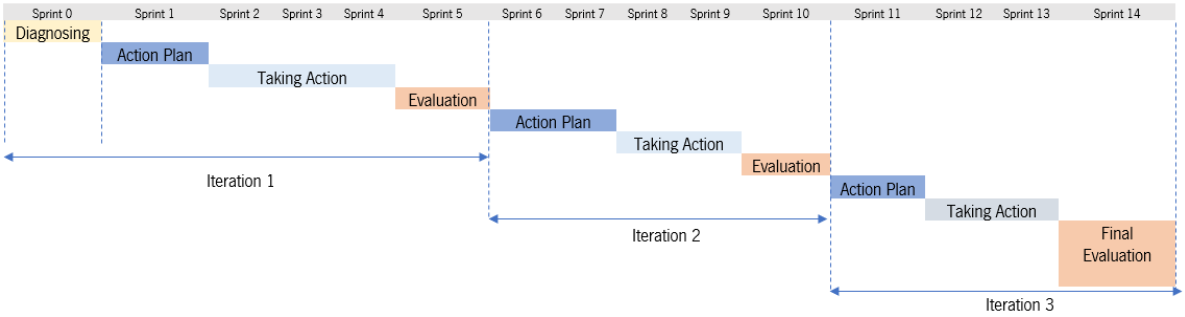

Figure 14 - Timeline of the CI Process Development

3.1 Diagnosing

For two months the investigator reviewed Bosch documentation to gain a better understanding of the company's culture, employees' actions and practices, environment and the context of the projects under study. To perform the Diagnosis phase, the researcher applied data collection methods such as direct observation or consultation of pre-existing documentation.

When this dissertation project started, the Algo Team had their project in a pre-development stage. For one month, the team received new team members and done research work for being prepared to kick off the new project. The team was completed with the entrance of the 8th developer to integrate the Development Team. Algo team will contribute with innovation algorithms for a bigger project that aims to be managed from Plymouth, constituted for software, algorithm, mechanical, electrical and system and test teams localized in Plymouth (USA), Braga (Portugal) and in Hildesheim (Germany) (See Figure 15).

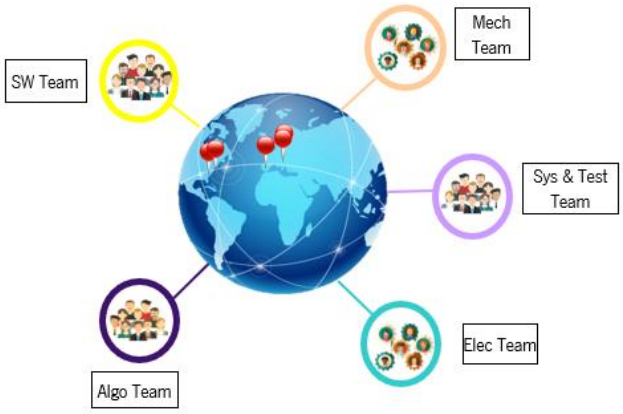

Figure 15 - Illustration of the Geographic Distribution of the Interdependent Teams

As mentioned above, the Algo Team had all the team members localized in Braga (Portugal), but they were part of a bigger project, and must collaborate with four more teams, localized across USA and Germany. All these teams, although being geographically separated, they are interdependent, and even being separated they had to define a working method that fits in the global project, so they can align and coordinate the deliverables between them. This method should also be capable to attenuate the problems that usually distributed teams face, like lack of trust and transparency, communication issues, leading to misunderstanding of the requirements, difficulty in transferring knowledge, time zone mismatch and lack of in-person interaction. Imagine a day when the Algo team leaves an unfinished User Story that would be indispensable for the team in Plymouth to continue their work. This would result in a block for the Plymouth team, and in the worst-case scenario, it could result in the sprint's failure. Even if there are strong reasons to believe that it was an inadvertent slip from the Algo team, this could cause resentment in the Plymouth team, resulting in the increasing lack of trust between teams. When teams are geographically distributed, is very common for each team to do their part, deliver the work for the other team and then point fingers if something goes wrong.

With Bosch being such an important company in the technology market, that wants to be in the forefront of the IoT world, did not seem reasonable not to have any agile method to support the team in organizing

and managing their work as well as manage the interaction between teams. So, the management of the project decided that each team should use Scrum as their agile framework, with the purpose that it can be scaled to its higher level as SAFe (Scaled Agile Framework) and promote the right interaction between teams.

At this point the Algo team was constituted for one Product Owner and one Development Team with eight elements but was not applying any agile methodology or framework like Scrum. The team did not have a Scrum Master or a structured process to prioritize the backlog items or to distribute tasks. Therefore, it was necessary to implement the Scrum framework in their working method.

On the other hand, A&V Team was already a mature team in the area of Machine Learning when this project started. The team was composed by ten elements that already know each other very well. They had no Project Manager neither a Product Owner, the team was responsible for itself, and decided what to do, when to do it, who will do it and how it will be done. Even though they are an independent team, the team members were geographically distributed across Portugal (Braga), USA (Plymouth) and Germany (Hildesheim) (See Figure 16).

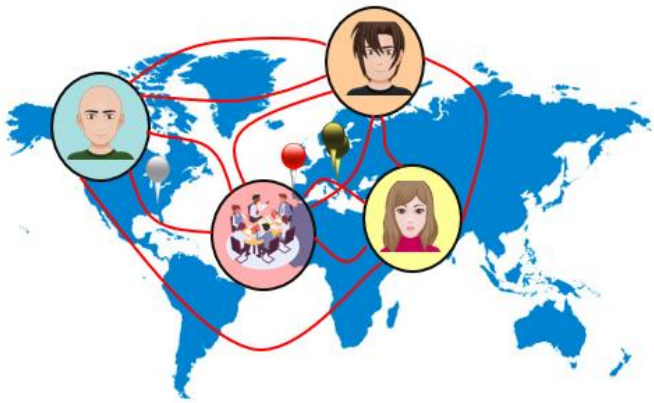

Figure 16 - Illustration of the Geographic Distribution of the Elements of A&V Team

When observing their working methods, some issues related to their organization were noticed: The team only had one excel with features to develop and due dates to delivery their products, already outdated; They were not using any framework to help them in planning, distributing tasks or solving problems; They had no Product Owner, Scrum Master, or Project Manager (See Table 3); The team was self-organized but that was causing overhead to some team members, that had to manage the relationship with the stakeholders, define what features should be developed next by the team, distribute the work for the rest of the elements and still do their development work, ending up having a conflict of responsibilities.

Table 3 - Scrum Roles, Activities and Artifacts existing in the Teams

Roles/Activities/Artifacts	Algo Team	A&V Team
Product Owner	✓	✗
Scrum Master	✗	✗
Development Team	✓	✓
Product Backlog	✓	✓
Sprint	✗	✗
Sprint Backlog	✗	✗
Kanban Board	✗	✗
Sprint Planning	✗	✗
Daily Meetings	✗	✗
Sprint Review	✗	✗
Sprint Retrospective	✗	✗

To have a clearer idea how the A&V team felt about themselves and their work was performed a Diagnosis Meeting, where each team member had to identified positive and negative aspects that have occurred in the team over the last three months. The result can be seen in Figure 17.

Figure 17 - Diagnosis Meeting's Results

Being part of an innovation project, developed in collaboration with other Bosch centers required a lot from the team, many issues regarding communication, planning and problem-solving were mentioned in

this meeting, so it was identified the necessity to find a way to mitigate some of these problems, with the implementation of the agile framework Scrum in A&V Team.

Nevertheless, the previous experience with Scrum in the department has revealed that with time, the teams started to feel demotivated. Although Scrum could help the teams improve their performance, it was not necessarily effective in solving all problems. The team was involved in high levels of problem-solving regarding systematically improve their understanding and application of innovation practices.

The revision of the department's documentation evidenced a negative feeling existed within the teams regarding the Retrospective Ceremony. This ceremony is the ideal moment for the team to discuss the lessons learned throughout the Sprint, reflect about what needs improvement, and become better at each sprint. This is the right event to trigger a continuous improvement culture within the team, identify the wastes of the processes so they can be eliminated, and make the team continuously better. But the teams in the department that already had utilized Scrum were not efficient in applying the suggestions given by the team members during the retrospective. This ceremony instead of creating a good opportunity to improve, had more become like a "landfill of good ideas", waiting for someone to do something about it. This defensive behavior was creating a big impediment to the construction of a continuous improvement culture in the teams' core (See Figure 18).

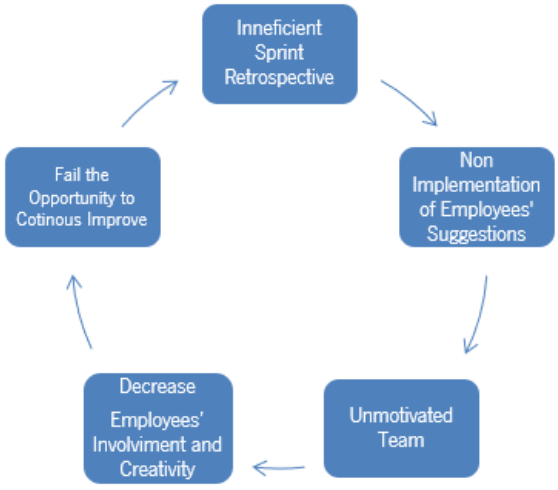

Figure 18 - Retrospective Problem's Cycle

Therefore, it was identified the main need to create a simple process, that could define the next steps to take, after the retrospective ceremony to solve real-life problems. This process should help the team in bring to light deeper problems, proceed to their prioritization, planning and implementation, creating and sustaining a continuous improvement mindset and behavior, exploring the implementation of the creative ideas given by the employees during the retrospective ceremony and keep the employees motivated and involved in their work.

3.2 Continuous Improvement Process Development

The Continuous Improvement Process was designed, implemented and optimized for four months in A&V Team. It was defined that before proceeding with the development of the Continuous Improvement Process, first would be necessary to implement and stabilize the Scrum process within the team. Once the CI process would have as input the results of the Retrospective ceremony, this activity captured a special attention in this study. Although, the team where the CI process was implemented was a distributed team, so the Sprint Retrospective had to occur online, and it was not possible to perform the activities planned by the scrum master in order to make that ceremony more dynamic and productive. Thus, in section 3.4 are presented different Retrospective activities applied in the Algo team in order to create the right environment to obtain good results from the ceremony throughout the time.

3.2.1 Iteration 1

Action Planning

Firstly, was necessary to build the right conditions to design and implement the continuous improvement process. If the CI Process would be carried out together with the Scrum cycle, was necessary to define how Scrum would be implemented in the first place. For that reason, the Scrum framework was planned to be introduced to the A&V team, with templates and guidelines developed by the Scrum Master to support this presentation.

Next, was opened a discussion with the team regarding the Scrum activities and artifacts that they consider beneficial to adopt. The team decided to translate de Stakeholders requirements into Epics, Features and User Stories to be added to the Product Backlog. Then adopt the Sprint Planning and create the Sprint Backlog, execute a two-weeks Sprint, and in the end of each Sprint, perform the Sprint Review and Sprint Retrospective, as can be seen in Figure 19. It was also agreed that the Daily Meeting will be realized only two times per week, at Tuesdays and Thursdays with a duration of thirty minutes each, due to schedule incompatibility with the team members that were in Germany.

Once the team was geographically separated, it would affect the way that every Scrum procedure will need to occur. To mitigate this problem all the Scrum activities would have to be realized or supported by digital platforms like Skype and IT project management tools like Azure DevOps.

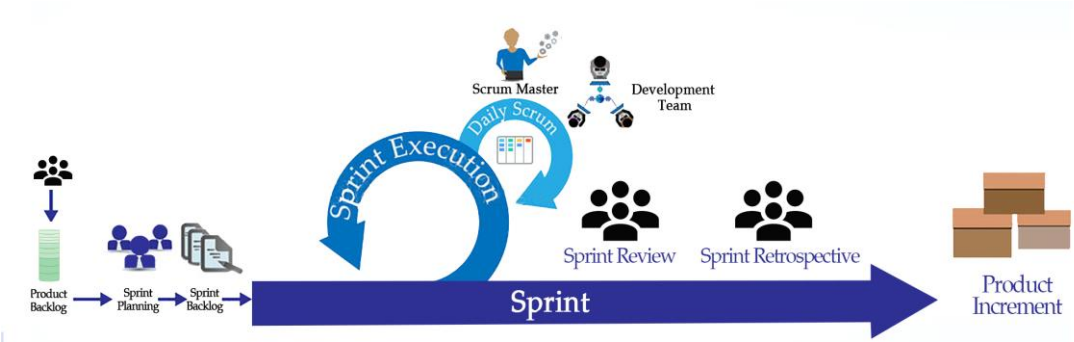

Figure 19 - Scrum Roles, Activities and Artifacts Adopted by A&V Team

Taking Action

The implementation of Scrum framework started with the introduction of this framework to the team through a few presentations, where the Scrum Master exposed the Scrum principles and values. During this introduction period were also presented the Scrum ceremonies: Sprint Planning, Sprint Review and Sprint Retrospective, by answering the questions: what, when, why, who and how these ceremonies should be performed, so every element of the team could be clarified about what was supposed to do in each one of them, and the process could flow smoothly (see Figure 20).

Figure 20 - Examples of Templates and Guidelines Used During Scrum Introduction

Since the team still had no PO, the first step taken in the direction of Scrum implementation was build the story telling of the product by themselves in Azure DevOps tool (See Figure 21), i.e., create Epics, and break them down into Features and User Stories to accomplish the requirements imposed by the stakeholders. From that moment, the team could start to plan their work and refine the Backlog throughout the time.

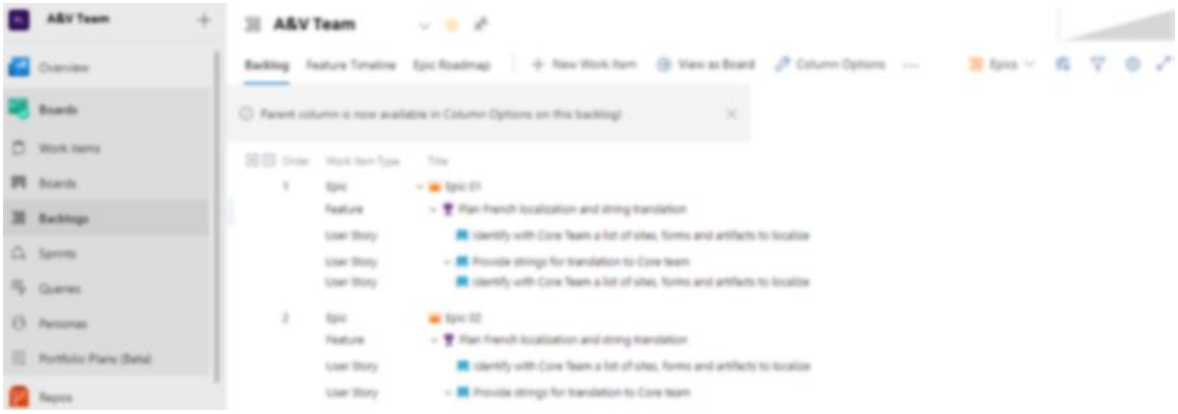

Figure 21 - A&V Product Backlog

a) Sprint Planning

Next, the team joined in a room to perform their first Sprint Planning. Once not every team member was in Braga, this meeting, from that moment, would always happen online, through a Skype conference call, so everyone could be part of it. During the Sprint Planning, the team shared the Sprint Goal and decided which User Stories would be committed to the next Sprint and to whom they will be assigned, creating the Sprint Backlog. It was team's decision not to give punctuation to the User Stories. Then a Kanban board was automatically created with the User Stories to execute during that Sprint. The responsible for each User Story was under obligation to define what tasks were necessary to execute to consider that User Story as "Done" and must add the tasks in the Kanban Board too, in order to improve visibility and transparency.

b) Weekly Meeting

During the Sprint, the team met two times a week to monitor the progress of the User Stories. Each team member answered the questions: "What have you done?", "What are you doing?" and "Is there anything blocking you?", and updated the status of each task for: "To Do", "In Progress", "Blocked", "Review" or "Done", depending on their development stage (See Figure 22). These weekly meetings had the duration of thirty minutes each, and were replacing the more usual Daily meeting in the Scrum cycle. This moment was ideal to keep everyone in the team informed about what the other team members were doing, preventing two people from doing the same for lack of information. After the execution of some Sprints, the team decided that would not be allowed to a team member to work at the same time in two different User Stories, limiting the work in progress, so everyone can focus on finishing one User Story at a time, and only after that, if there is still time, develop another one. This measure was implemented after the team realize that were closing Sprints with too many "Undone" User Stories by lack of focus.

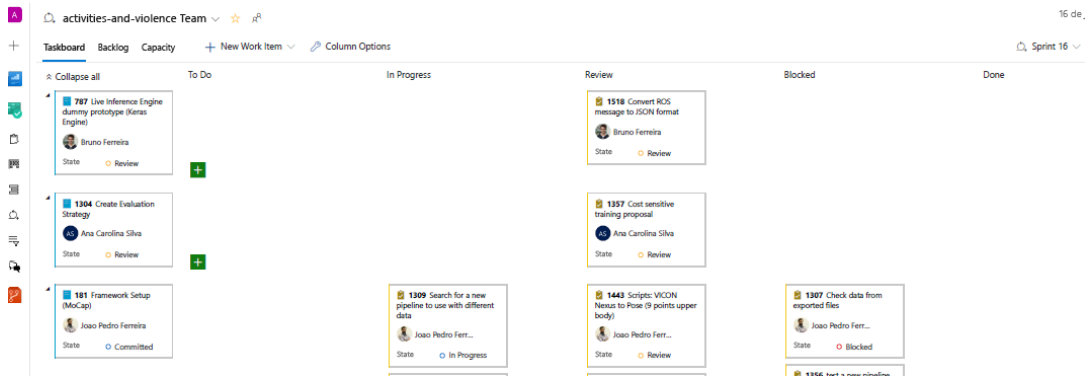

Figure 22 - A&V Kanban Board

c) Sprint Review

The Sprint Review ceremony occurred in the end of Sprint with all the team members present and if possible, some Stakeholder too, and had an average duration of one hour.

During the Sprint Review, every team member exposes the work that was done during the last sprint to complete the assigned User Stories. The team discuss what went well during the Sprint, what problems it ran into, and how those problems were solved. They also consider what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;

Occasionally, the team prepared demos and invited their stakeholders to this meeting, so they could give their feedback regarding the progress of the project.

To facilitate the execution of this ceremony, each team member was responsible to document on the wiki page of the Azure DevOps platform the progress and results of their work to the assigned User Story.

The page dedicated to each User Story provides the following information:

- Responsible;
- Description;
- Acceptance Criteria;
- Results;
- Additional Comments;
- Future Work;
- Link to the User Story in Backlog.

This page serves as support for that presentation, once the team can see in a very clear way if the acceptance criteria were achieved and evaluate if the US is considered “Done” or not.

Besides facilitating Sprint Review, these pages create an historic with the steps taken, results and explanation of the decisions made regarding each User Story throughout the project, facilitating the integration of possible new members in a specific topic already developed by the team, or just simplify the consultation of the documentation for all the team members about any User Story.

Since the team has no PO, the team members responsible to perform that role are the ones who accept or not the User Stories of the increment of the product. If the US was consider Done, i.e. all the acceptance criteria were met, its status would be updated to "Done" and it leaves the Kanban board, otherwise the team must reflect and understand if the US is still valuable, if it changed its scope but keeps on the backlog or if it can just be deleted because it no longer brings value to the Backlog (see Figure 23).

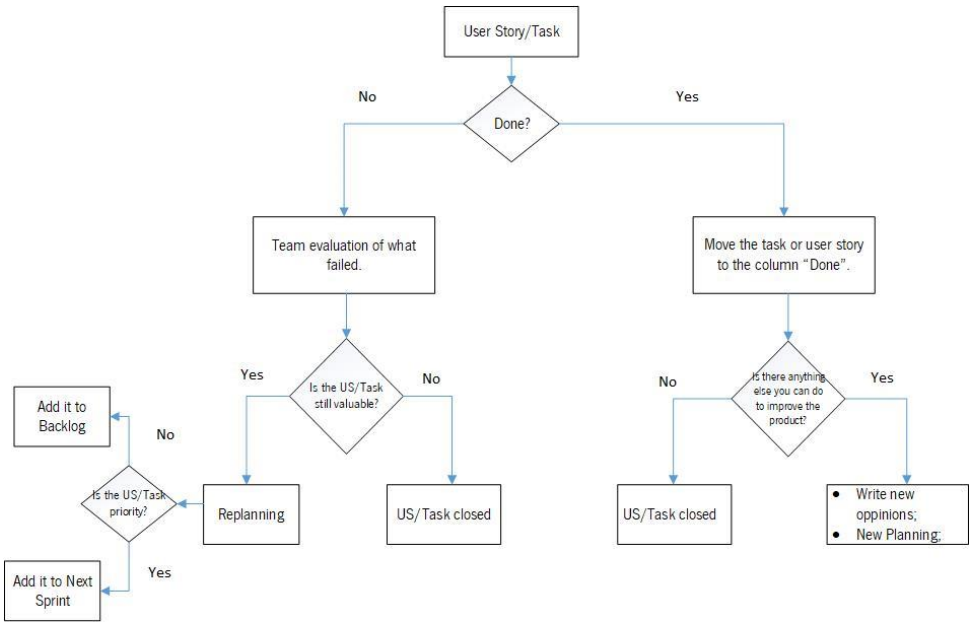

Figure 23 - Sprint Review Decision Making

d) Sprint Retrospective

The Sprint Retrospective ceremony is the last activity of the Scrum cycle, it took place after the performance of the Sprint Review ceremony and had an average duration of one hour. The retrospective ceremony, as its name implies, aims to lead the development team through a reflection on what was done during the last sprint, what problems arose, how they were overcome, what went well, what didn't go so well and why, and finally what can be done differently next time to improve the process, promoting the cycle of inspect and adapt (See Figure 24).

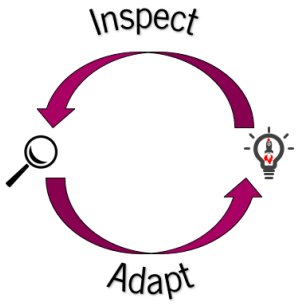

Figure 24 – Inspect and Adapt Cycle

Anything that affects how the team creates the product should be discussed, including processes, practices, communication, environment, artifacts and tools.

Since the team was geographically separated, the Scrum Master couldn't perform the in-person activities idealized for this ceremony, and it was also necessary to realize this event online, using the Retrospectives extension of Azure DevOps, thus every team member could participate and contribute with his opinion (See Figure 25).

This ceremony followed the net steps:

1. Collect feedback about what went well and what didn´t go so well during the sprint;
2. Cluster all the information;
3. Each element votes in the cluster that influenced the sprint the most;
4. Select things that the team should keep doing and stop doing;
5. Suggest actionable improvement solutions.

By the end, they should have identified improvements to implement in the next Sprints.

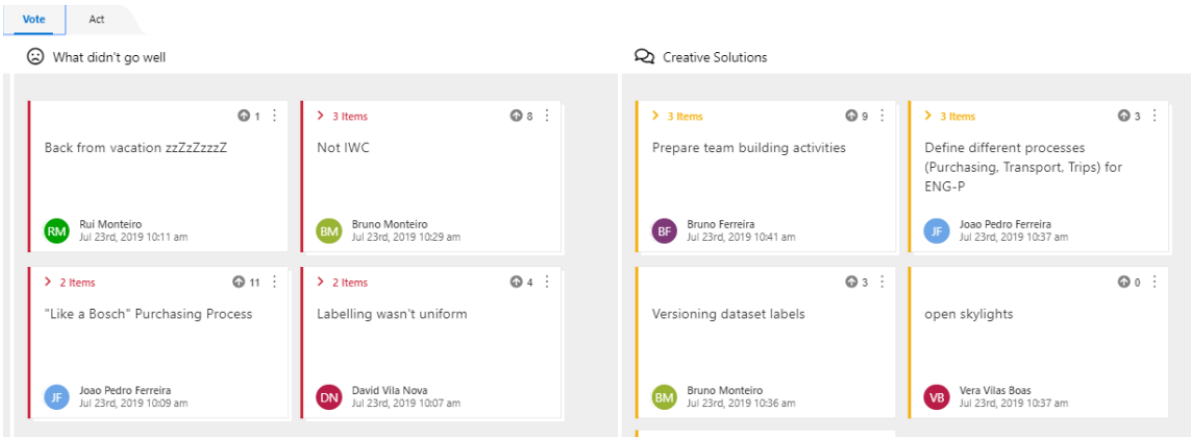

Figure 25 - Example of Sprint Retrospective's Output from A&V

Through this cycle of inspect and adapt, The Sprint Retrospective ceremony gave the team the opportunity to customize Scrum to its own circumstances, it also allowed it to take advantage of perceptions and

information before they get lost and provided the chance to apply early and incremental learning throughout the time.

Evaluating

The Scrum cycle was implemented and after two months the process was stable, and the team was able to perform it very naturally. Table 4 depicts the exiting Scrum roles, activities and artifacts, in the A&V team at the beginning of this project and the end of this implementation process. As can be seen in the table below, currently the Product Owner is the only Scrum role that is not represented by one person in the team, by team decision the responsibilities associated to this role were distributed for three team members of the DevTeam.

Table 4 - Initial and Current Scrum Roles/Activities/Artifacts existing in the A&V Team

Roles/Activities/Artifacts	Initial State	Current State
Product Owner 	✓	✗
Scrum Master 	✗	✓
Development Team 	✓	✓
Product Backlog 	✓	✓
Sprint 	✗	✓
Sprint Backlog 	✗	✓
Kanban Board 	✗	✓
Sprint Planning 	✗	✓
Daily Meetings 	✗	✓
Sprint Review 	✗	✓
Sprint Retrospective 	✗	✓

In order to understand the opinion and acceptance of the team members regarding the Scrum activities and artifacts adopted during this project, a questionnaire developed with the following questions:

Q1 – I think that team meetings are important.

Q2 – Team Meetings promote communication and problem solving.

Q3 – I think is important to participate in Daily/Weekly Syncs.

Q4 – I think is fundamental to update the Sprint/Kanban board.

Q5 – The use of Scrum enhances my daily work.

Q6 – I think is important to define and share the sprint goal with the team.

For each question were available five possible answers following a Likert scale: strongly disagree; disagree; neutral; agree; strongly agree. The questionnaire was delivered to the nine team members that constituted the team in that time by e-mail obtained a valid reply from all of them.

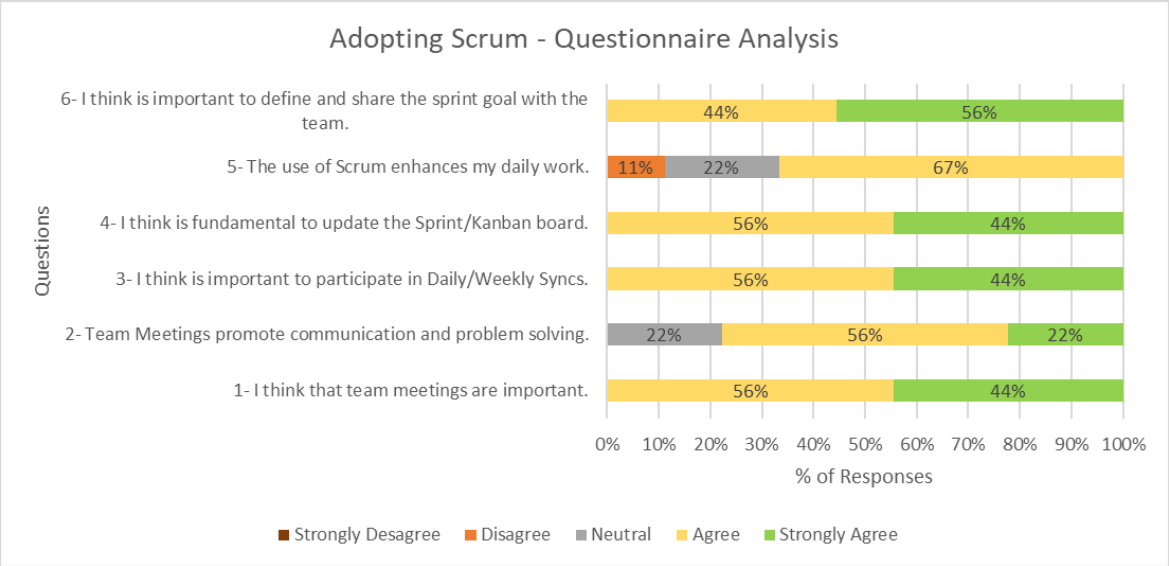

Figure 26 – Adopting Scrum - Questionnaire Results

The results of the questionnaire can be seen in Figure 26, that depicts the general good feedback given by the team members regarding the scrum activities and artifacts adopted during this project, with a total of 91% respondents considering "agree" or "strongly agree" with them. The questions Q1 - "I think team meetings are important.", Q3 – "I think is important to participate in Daily/Weekly Syncs.", Q4 – "I think is fundamental to update the Sprint/Kanban board" and Q6 – "I think is important to define and share the sprint goal with the team.", were the questions that obtained a percentage of 100% of positive feedback, clearly showing the importance that the team members give to their participation in team meetings and involvement in the process improvement. This is strongly associated to the main Lean idea of respect for humanity as one of the basic concepts of TPS besides the use of employee creativity. The results of the questionnaire also evidence the perception of the employees regarding the benefits of visual management acquired with the use of Kanban boards in the daily routine and Weekly Meeting. Written comments were also provided by some members in order to clarify aspects that were not covered in the questionnaire or to provide specific feedback. Some examples are:

Team Member 1 – "*From a global viewpoint, I think the scrum meeting are quite important in order to describe the team work being developed in the last few weeks and to understand, from a global viewpoint, the overall status of the project.*"

Team Member 2 – "*The ceremonies and interaction between team members created a great environment to work in. In the future retrospectives we should focus more (steer the team) to the problems we can actually solve and spend less time with the ones we can't do much about (ex. resources).*"

Team Member 3 – "*Good work*."

Team Member 5 – "*Pretty good work, with potential for improvements*."

Team Member 6 – "*During Retrospective Ceremony every member should talk briefly about positives/Negatives. Writing is not enough! Focus more on what the team can actually solve.*"

3.2.2 Iteration 2

Action Planning

Once the Scrum cycle was successfully implemented and after two months was already part of the team's routine, it was considered that the time to start developing the Continuous Improvement Process as come. During this iteration it will be described the inception of the CI process into the team's development cycle, that should be performed besides the Scrum ceremonies and activities. Considering that the objective of this project is to improve in every process every day, this CI process should be capable of identify problems or opportunities for improvement, ideate solutions, proceed to their development and/or implementation and finally evaluate the effects of the solutions implemented. This process should also be embedded in team's daily work until it become in that mentality "It is the way we do things around here!".

In the first place was necessary to define which CI strategy would be implemented in the team. This decision had to consider the fact that this process would be implemented in software development teams and the activity of software development is not like a batch or mass production designed to produce standardized products like cars or packaged food. In the area of software development, every product is different and is highly dependent on human creativity and intellect, therefore it is intrinsically linked to a great source of variability. As result of this variability it is hard to have a consistent data's gathering and analysis, thus during this project it was not possible to obtain a sample of data with reliable measures, like mean or standard deviation to proceed to its statistical analysis. Considering this limitation, technics like DMAIC-Six Sigma were not an option to adopt.

Between benchmark and analytical approaches like ISO 9001, ISO 15504, ISO 12207, BOOTSTRAP, CMMI and PDCA, Balanced Scorecard, Quality Improvement Paradigm, IDEAL, DMAIC-Six Sigma or Total

Quality Management, respectively, the PDCA was the one selected to serve as basis of the CI process due to its application to improve processes, products or services in any organization, also its iterative strategy, that through the repetition of the PDCA cycle allows the team to get closer to their goals. In addition, the researcher's frequent contact with this approach was one of the reasons that led to its selection.

PDCA

As mentioned in section 2.3.4, the Plan-Do-Check-Act is a continuous improvement strategy originated in manufacturing area, with the purpose to improve production lines or products through the successive execution of PDCA cycles, obtaining a contribution of small improvements every day, that would become into a major improvement at long term.

This approach is composed for four steps, starting with Plan, where the current state is analyzed and a problem or opportunity for improvement is identified, consequently a plan to implement a corrective action or an improvement is created. In this step are also defined the targets wanted to achieve. Next, in the Do phase the plan designed in the previous step is tested, ideally in a small scale with a controlled environment to validate the improvement. In the Check step is done an observation of the improvements in order to proceed to their evaluation, and compare them with the predictions done in the Plan phase. If the plan did not result as expected, should be performed an analysis to understand the root-causes of the problem. Otherwise if the results were positive, the last phase Act should be carried on. In this step is made the decision to implement or not in large scale the improvements tested previously. Then the cycle goes on and is repeated consecutively.

Taking Action

The continuous improvement process implemented in the daily work of the A&V Team was designed so that in cooperation with some Scrum cycle activities, all the PDCA phases could be fulfilled so the team could improve their problem-solving capability.

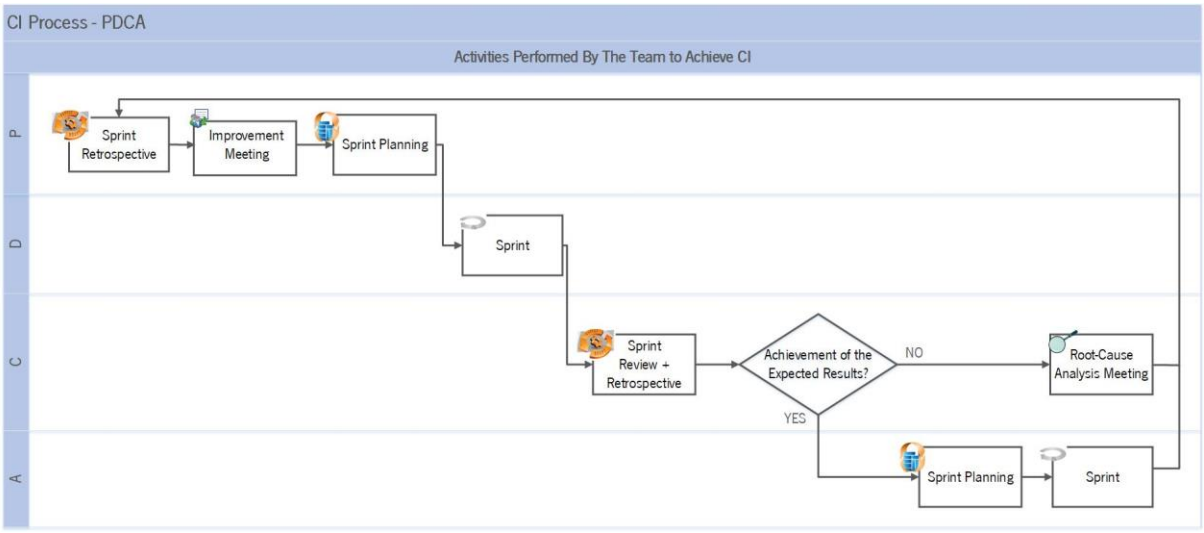

Figure 27 – First Proposal of CI Process

In Figure 27 besides the Scrum ceremonies and activities, can be seen the introduction of the Improvement Meeting, a point of decision and the Root-Cause Analysis Meeting in order to guarantee the right inputs and outputs for the execution of the continuous improvement process.

To accomplish the Plan phase of the PDCA cycle, the team had to perform a Sprint Retrospective ceremony, an Improvement Meeting and a Sprint Planning. During the Sprint Retrospective the problems or opportunities for improvement were identified and then were suggested some improvement or corrective actions. Once the team were very inefficient in gathering these suggestions and implement the improvement actions, an Improvement Meeting was introduced. This meeting occurred after the Sprint Retrospective ceremony, by a Skype conference call that could count with the participation of the Scrum Master and the team members responsible to manage and refine the backlog. During the Improvement Meeting, the outputs of the Sprint Retrospective ceremony were analyzed in detail, the improvement suggestion given by the team member were evaluated, and highlighted which ones would bring more benefits to the team without requiring too much effort. Then those suggestions were translated into User Stories – Improvement User Stories - and added to the top of the Product Backlog. This process is represented below through an ETVX diagram (See Figure 28).

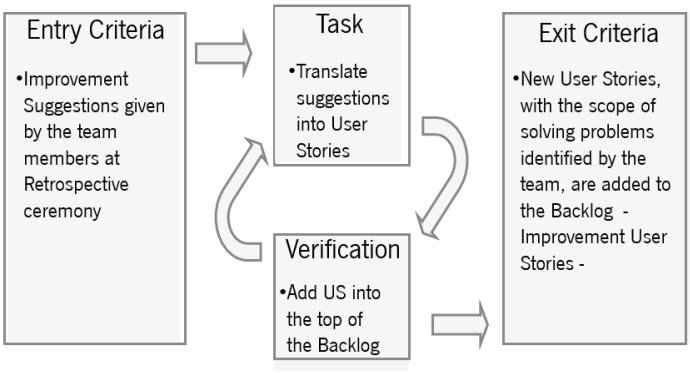

Figure 28 - Improvement Meeting ETVX Diagram

Next, when the team performed the Sprint Planning, the User Stories with the scope to implement an improvement or corrective action, were already on the Product Backlog, together with the top priority items. Then the team could select one or more Improvement User Stories and add them to the Sprint Backlog and assign them to team members. This way, the team would be committed to make some effort during the next sprint, in order to improve or solve some problem.

Following the PDCA cycle, the next phase to achieve would be the Do step, where the plans defined previously are implemented or tested. This step is fulfilled with the performance of the Scrum activity - Sprint. During the Sprint, the team work to accomplish all the User Stories planned in Sprint Planning, inclusively the Improvement User Stories. So, through the execution of Sprint, the plans designed in the Plan phase were implemented or tested. This implementation can be divided into two moments, first if necessary, the team had to develop the solution, it can be a tool for example, and only after that implement it.

Subsequently, the cycle follows for the Check step. To fulfill this phase, the team performs the Sprint Review and Sprint Retrospective. During the Sprint Review, the DevTeam presents the Improvement User Stories that were done and that were not done during the Sprint and discussed the emerging problems. Following for the Sprint Retrospective, what went well and what did not go so well were identified and the team reflected if the improvement suggestions were implemented and if they observed or not an improvement, creating the first checkpoint of the process. In this point, the team evaluate if the implementation of what was planned brought the expected results for the team. If the goals projected in the Plan phase were not achieved, during the next sprint the team had to perform a Root-Cause Analysis Meeting in order to understand the root-cause of why the improvement did not worked out. This process is represented below through an ETVX diagram (See Figure 29).

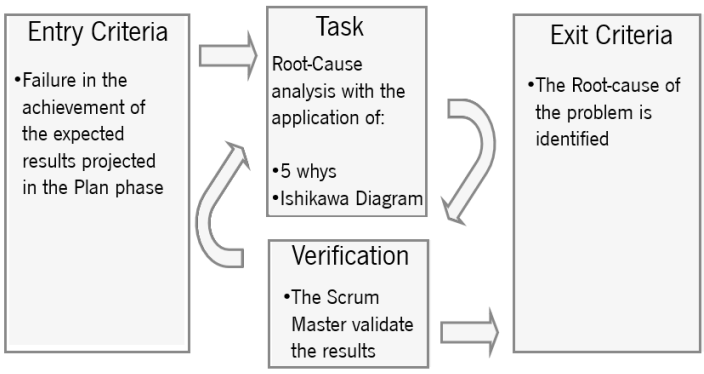

Figure 29 - Root-Cause Analysis Meeting ETVX Diagram

The output of a Root-cause Analysis Meeting performed by the A&V Team can be seen in Figure 30. In this case the team analyzed the problem: Lack of human resources.

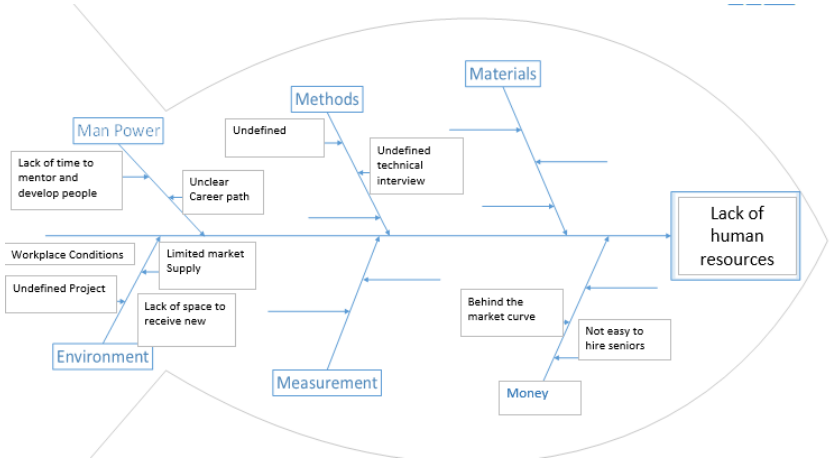

Figure 30 – Output of Root-cause Analysis

Once the Root-cause of the problem was identified, the PDCA cycle jumped the Act phase and followed for the Plan step, where a new plan was defined, taking into account the new considerations emerged from the Root-cause Analysis Meeting and the cycle was repeated.

On the other hand, if in the checkpoint the improvement implementation had a positive evaluation, the PDCA cycle goes on, following for the Act phase, where the improvement actions tested in the sprint are planned during the Sprint Planning to be implemented in large scale during the next sprint. Then the cycle is repeated.

Evaluating

As can be seen, the implementation of the new activities: Improvement Meeting, checkpoint and Root-cause Analysis Meeting, brought some structure to the CI Process. In the first instance, forced the team to make some effort to create the Improvement User Stories so they could be implemented during the

sprints and then evaluate the effects of their implementation. If the results were negative, the CI Process would direct the team into searching for the reasons of the failure and guide them to design a new plan.

However, during the execution of the Improvement Meeting was noticed some entropy, this meeting exceeded too many times the time-box planed, because the three team members participating in this meeting had some difficulties in decide which improvement suggestions gave by the team during the Retrospective should be added to the Product Backlog in the form of User Story. Although the team had voted previously in the suggestions, these three elements had the responsibility to define which of these suggestions could bring more value to the team, always considering the effort required to implement them. This decision usually should be PO's responsibility, but once this role was divided into these three members, this choice could be influenced by their own motivations as individual developers, neglecting the team's vision.

A review of the overall process evidenced the lack of monitorization of the problem resolution status, also was identified a lack of documentation regarding the actions, tools and processes implemented. In order to solve these gaps, was essential to plan and perform a new iteration.

3.2.3 Iteration 3

Action Planning

With a view to address the needs mentioned in the Evaluating phase of the previous iteration, some changes were introduced in the CI Process. First, was decided to implement a new activity in the process, in order to facilitate the decision making regarding the selection of the suggestions given by the team during the Retrospective ceremony, that would be translated into Improvement User Stories. This activity ideally should count with the participation of all team members, so everyone could contribute with their opinion in the identification, selection and prioritization of suggestions. Finally, the lack of documentation existing during the CI process would be addressed with the adoption of an Open Point List (OPL).

Taking Action

The Figure 31 depicts the introduction of the new activity - Selection of the Priority Suggestions - in the CI Process, as well as the adoption of an OPL.

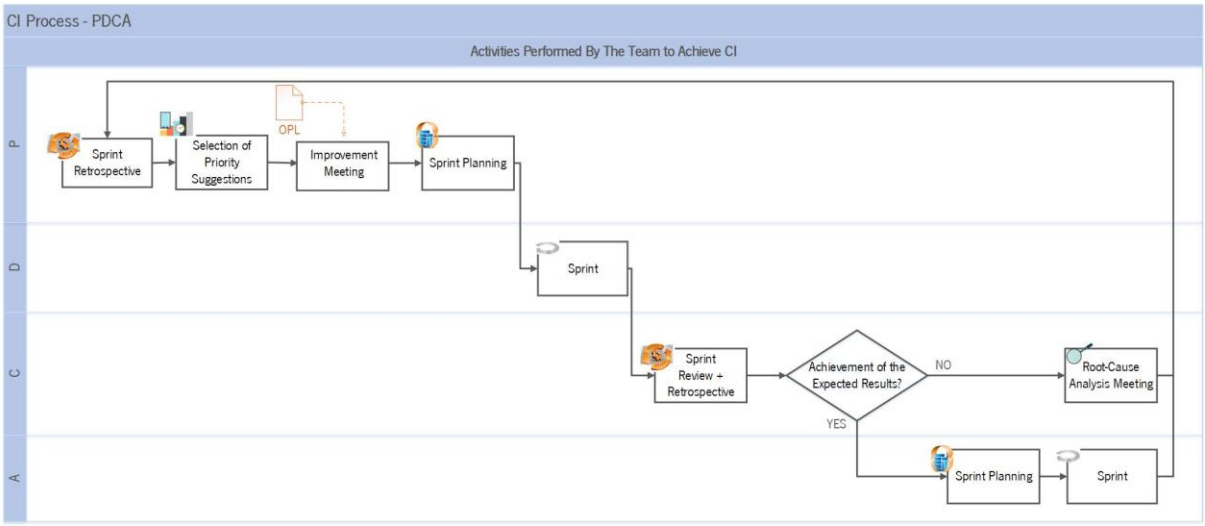

Figure 31 – Second Proposal of the CI Process

The activity Selection of the Priority Suggestions was introduced right after the Retrospective ceremony, with every retrospective's participant in the room, the Scrum Master conducted the team into a moment of discussion and brainstorming, to evaluate the advantages, costs and effort required to implement each suggestion given by the team.

To support this meeting, a Value x Complexity Matrix tool was used, so the team could classify each suggestion given by its complexity and value and uncover the suggestions that promised to deliver the most value for the least effort. The Complexity parameter, represented in the YY axis, had into consideration the implementation time, the dependency of people out of the team and the monetary cost. On the other hand, on the XX axis, was considered the value that the implantation of that suggestion could bring to the improvement of the team. The suggestions were then plotted on a quadrant and prioritized accordingly (See Figure 32).

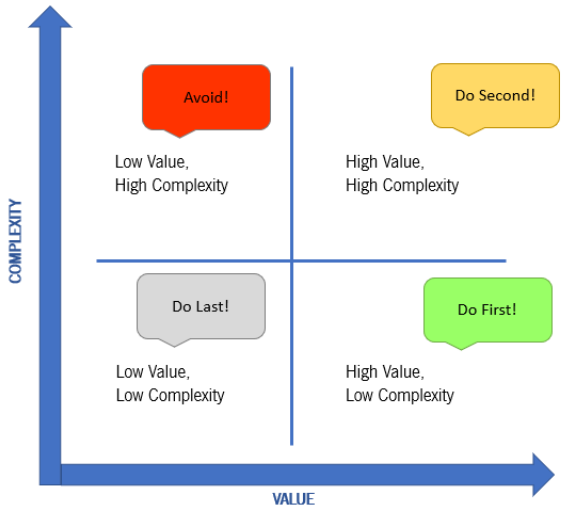

Figure 32 - Illustration of a Value x Complexity Matrix

The suggestions that were placed in quadrant with high value and low complexity, are the top priority items and should be prioritized above the others, once the team predicted that they could bring high value with low effort. However, might exist a few or any item in this quadrant, because due to its easy win is most likely that the team has already implemented it.

The high value, high complexity suggestions are worth prioritizing too, second after the high value, low complexity, but due its high effort required to implement them, these initiatives might have to be shelved. Next, the suggestions placed in the category low value, low complexity might have some interest in the team. They could represent an opportunity for small wins, without expending a great effort or resources, and the benefits would be cumulative throughout time. Lastly, the suggestions plotted in the quadrant relative to low value, high effort should be avoided. In fact, this framework was very useful to identify in a great visual way these initiatives, so the team make sure that will not waste their development time working on that type of suggestions.

The new activity Selection of the Priority Suggestions introduced in the CI is represented below, through an ETVX diagram (See Figure 33).

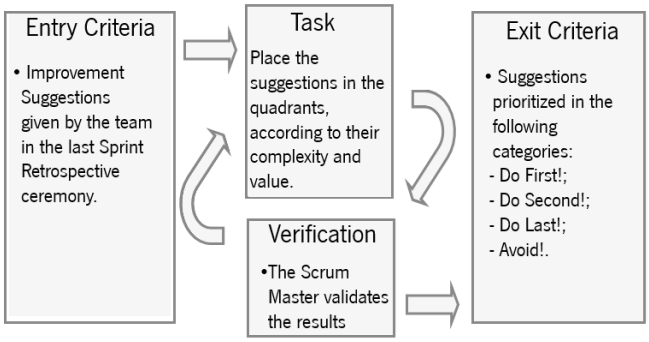

Figure 33 - Selection of the Priority Suggestions Activity ETVX Diagram

On Figure 34, can be seen a real example of an application of the value x complexity matrix by the A&V team during the Selection of the Priority Suggestions activity. The team evaluated the seven suggestions given in the last Sprint Retrospective ceremony and plotted them in the quadrants according their value and complexity. After some discussion the team decided to prioritize the suggestion represented with the number four: acquire the material needed that was in the Do first quadrant, and the suggestion number three: hire more people, plotted in the Do second quadrant. Although there was one more suggestion in the Do first quadrant, the team considered more important, in this case, to make more effort to work in hiring more people.

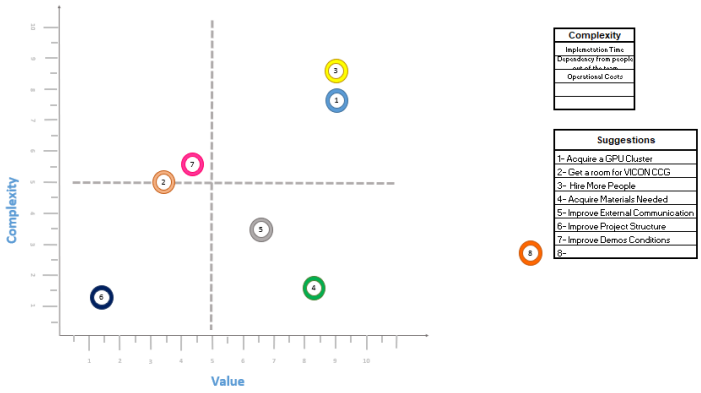

Figure 34 - Result of a Selection of the Priority Suggestions Activity

Moreover, was created an OPL in order to record the emerging problems and monitor their progress in a very visual way. This OPL provided information regarding the definition of the problem, the progress of its solution, its current status and some details of how the problem was resolved, so every time that a similar problem occurs, the team could consult this OPL and follow the instructions available.

This OPL was introduced for the first time during the Improvement Meeting, and after the Improvement User Stories to implement were written, the Scrum Master presented and explained how was supposed the team fill the OPL with the information regarding the description of the problem, when it was identified and its priority. Then the team members together with the scrum master defined due dates for the problem to reach each phase of the PDCA cycle. During the CI Process the team should maintain the status of the problem updated and record the procedures taken to solve the problem.

In Figure 35 can be seen the OPL created by the Scrum Master, and filled by the A&V team.

Creation Date	Problem/Opportunity	Date	Tracking								Status	Comment/ Process Description
			Plan		Do		Check		Act			
4-Jun-19	Difficulty in contacting the team members that were in another location	Real			5-Jun-19		18-Jun-19		18-Jun-19		Resolved	Add the link to the Skype conference call to meeting invitation.
		Planned			5-Jun-19		18-Jun-19		18-Jun-19			
2-Jul-19	Lack of people for audio-labelling	Real	2-Jul-19		4-Jul-19						In Progress	Interview and hire people for make a behavior and audio analysis of the recordings.
		Planned	16-Jul-19		17-Jul-19							
2-Jul-19	Bad communication between teams	Real	2-Jul-19		5-Jul-19		16-Jul-19		16-Jul-19		In Progress	Schedule regular sync meetings with external teams.
		Planned	2-Jul-19		16-Jul-19		30-Jul-19		30-Jul-19			

Figure 35 - OPL from A&V Team

Evaluating

The changes implemented during this iteration increased the involvement of the all team in the decision-making process, promoting the empowerment of the team members and provided a simple way to monitor the progress of the problem resolution.

The CI process was then reviewed step by step in order to find possible improvements in its structure, activities, tools involved and their sequence. In general, the process seemed complete and its goal achieved although there is always something that can be improved.

Firstly, was noticed that the suggestions that depended from people out of team expired frequently the planned dates, demotivating the team. In order to mitigate this problem is suggested the involvement of the top management in this process. Scheduling a meeting once per month with the project leader might help the team in speed up the problem-solving capacity, moreover if the project leader follows the team, and is able to experience and understand their problems, he might increase his effort to overcome some barriers that are not within the team's reach.

During this process, the Scrum Master also noticed that throughout the time, when performing the Sprint Retrospective ceremony, the team members sometimes had their answers and suggestions memorized, indicating that the data could be influenced or that the team member was not fully focused on the meeting, at that time. This might happen because the Sprint Retrospective followed the same steps every sprint and once the meeting had to be executed online, the Scrum Master was not able to try new and different Retrospectives approaches with the A&V team. Although, the development of the right environment and new activities were introduced and performed with the Algo team, in order to get at every sprint fresh data, and can be consulted in the section 3.4 – Complementary Work.

3.3 The Continuous Improvement Process

3.3.1 Standardized Process

The CI Process, developed and implemented during this dissertation project, was finalized in the third iteration. Once this project had a temporal limitation of nine months, although the Scrum Master had identified some improvement opportunities, the lack of time to implement new changes led to the standardization of the process. Figure 36 depicts the Standardized Continuous Improvement Process with all the activities, tools and artifacts applied during its execution.

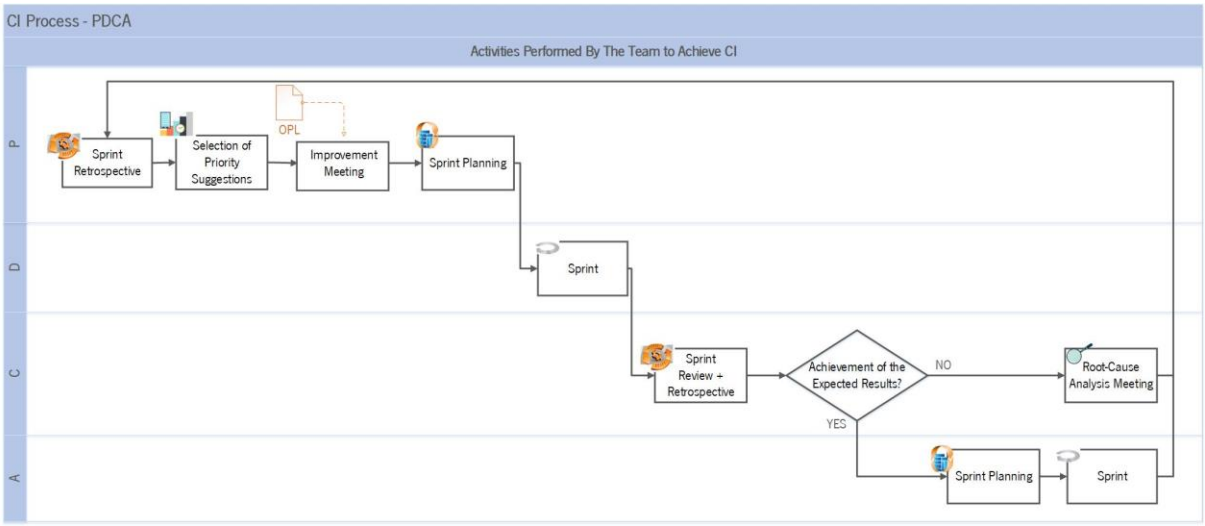

Figure 36 - Standardized Continuous Improvement Process

This CI process provides the necessary conditions for the teams to be able to identify the emerging problems or opportunities for improvement and give suggestions to address the problems or to improve some process during the Sprint Retrospective ceremony. Then the prioritization of the suggestions given is made at the Selection of the Priority Suggestions, to help the team in the definition of which suggestions should be transformed into Improvement User Stories at the Improvement Meeting, to later implement that Improvement User Stories during the Sprint. Next, the presentation of the results obtained from the implementation of the suggestions and the evaluation of its impact are made during the Sprint Review and Sprint Retrospective ceremonies. If the implementation of the Improvement User Stories failed or did not reached the expected results, a Root-cause Analysis Meeting is performed in order to identify and understand the contributing factors or causes of a problem that was inhibiting the implementation or getting good results, and a new plan is drafted. Otherwise, if the implementation of the Improvement User Stories met the expectations, the improvement actions tested during the sprint are planned at the Sprint Planning ceremony to be implemented in large scale during the next Sprint. Then the cycle is repeated.
A detailed explanation of each activity performed during the CI process can be consulted in section 3.2. The tools and artifacts applied during the process are described in the following section 3.3.2.

3.3.2 Tools and Artifacts

Sprint Retrospective Board

The Sprint Retrospective Board was used at every Sprint Retrospective ceremony. This artifact provided an easy way for the team to identify and cluster the team members opinion, regarding what went well and what did not went so well during the Sprint (See Figure 37) and also to register the improvement suggestions given by the team member (See Figure 38).

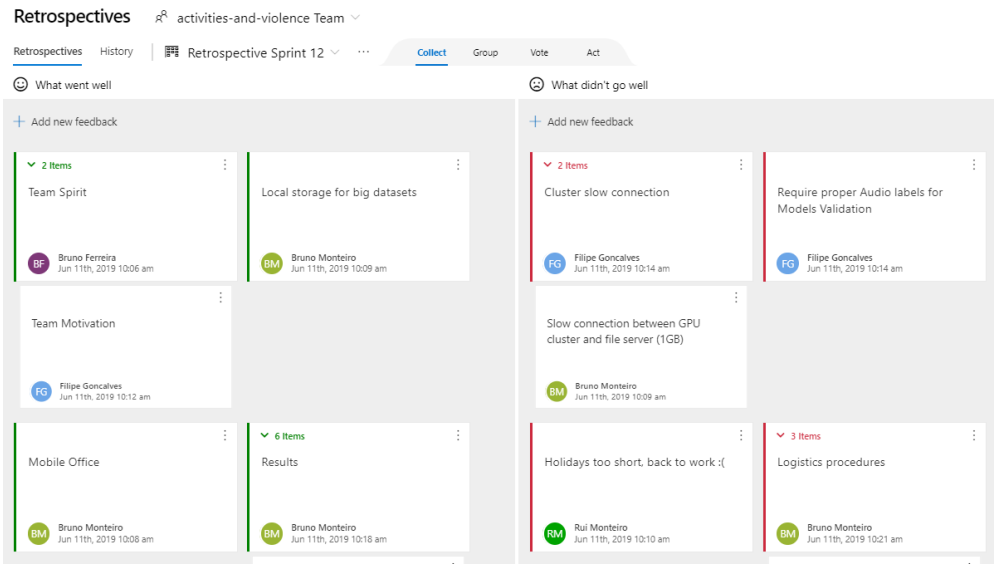

Figure 37 - Sprint Retrospective Board from A&V Team

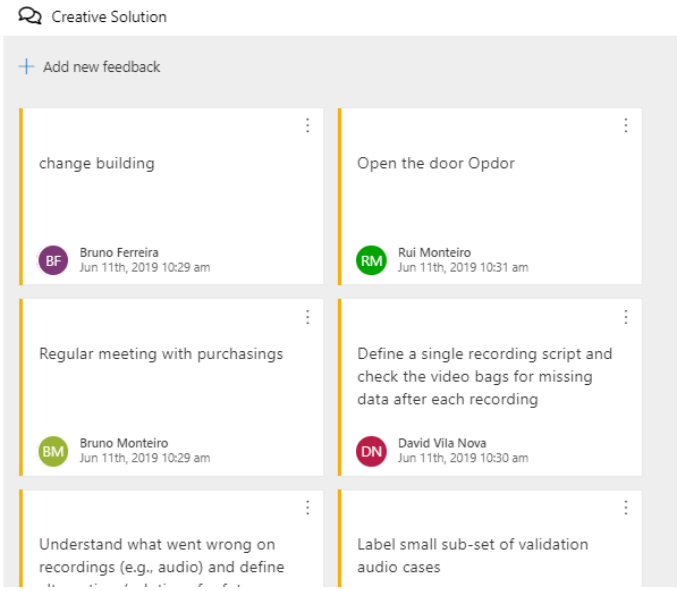

Figure 38 - Suggestions Board from A&V Team

Value x Complexity Matrix

This tool was used during the activity Selection of the Priority Suggestion, so the team could classify each suggestion given during the Retrospective ceremony, by its complexity and value, and uncover the suggestions more interesting to implement (See Figure 39). A detailed explanation of this tool can be consulted in section 3.2.3.

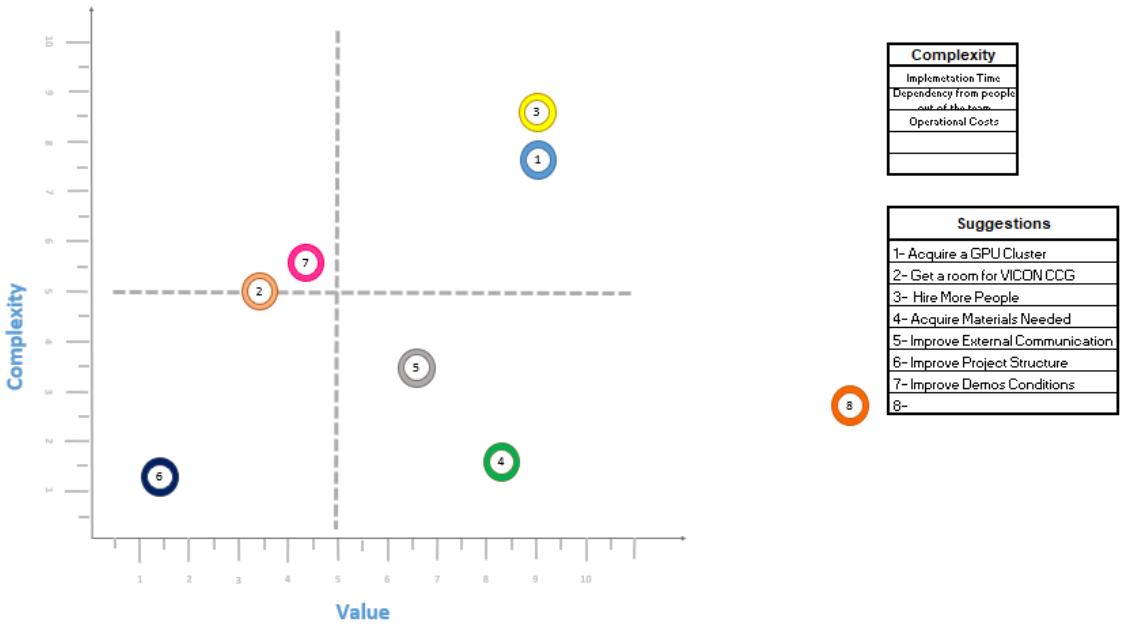

Figure 39 - Value x Complexity Matrix from A&V Team

Open Point List

As mentioned before, didn't exist any follow up of the problems identified at the Retrospective Ceremony, so most of them ended up of being forgotten.

During this project, was created an OPL, in order to record the emerging problems and monitor their progress in a very visual way. This OPL provide information regarding the definition of the problem, the progress of its solution, its current status and some details of how the problem was resolved (See Figure 40).

Creation Date	Problem/Opportunity	Date	Tracking				Status	Comment/ Process Description
			Plan	Do	Check	Act		
4-Jun-19	Difficulty in contacting the team members that were in another location	Real		5-Jun-19	18-Jun-19	18-Jun-19	Resolved	Add the link to the Skype conference call to meeting invitation.
		Planned		5-Jun-19	18-Jun-19	18-Jun-19		
2-Jul-19	Lack of people for audio-labelling	Real	2-Jul-19	4-Jul-19			In Progress	Interview and hire people for make a behavior and audio analysis of the recordings.
		Planned	16-Jul-19	17-Jul-19				
2-Jul-19	Bad communication between teams	Real	2-Jul-19	5-Jul-19	16-Jul-19	16-Jul-19	In Progress	Schedule regular sync meetings with external teams.
		Planned	2-Jul-19	16-Jul-19	30-Jul-19	30-Jul-19		

Figure 40 - Open Point List from A&V Team

Every time an Improvement User Story was written, the problem associated to it should be entered in the OPL, then the team should record the entry date and describe the problem clearly. Then they should plan a due date to each phase (Plan, Do, Check, Act), and record the date when the problem finished that phase – Real Date. In case the real date exceeds the planned date, the user would be warned by a red symbol. On the other hand, the user receives a yellow warning if there are only seven days or less, until the planned date. Lastly, if the problem is resolved before seven days until the deadline, the user receives

a green warning meaning that the problem had timely resolution in that phase. This allow the team to have a visual monitoring of the problem, and be aware of the delays that may cause the failure of what was planned, contrarily it can also show the team that they resolved the problem within time. Beside this, this OPL have a column designated as Status that provides a very fast perception of the current state of the problem resolution progress. Finally, it also have a Comment/Process Description column, so that the team can record a brief explanation of what was done to resolve the problem, creating a history of resolution processes that can be consulted every time that a similar problem emerged.

This tool provides a visual management by recording the progress of prioritized problems identified at Retrospective Ceremony, banishing their oblivion.

Fishbone/Ishikawa Diagram

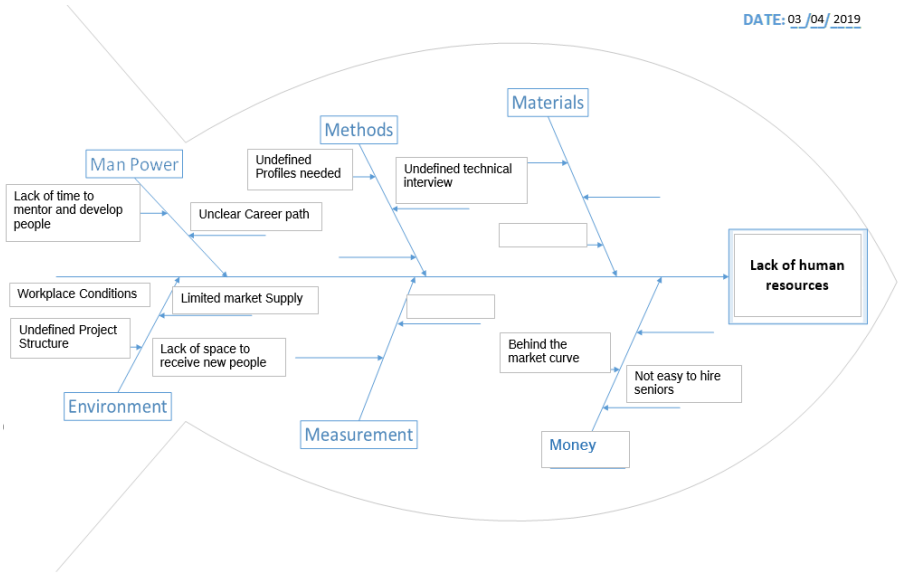

Figure 41 - Fishbone Diagram from A&V Team

Fishbone diagram was one of the tools used for the teams in the Root-cause analysis Meeting. This tool assisted the teams in identifying and understanding the contributing factors or causes of a problem that was inhibiting the implementation of the Improvement User Stories, and therefore help them in developing a better solution.

Through a brainstorming, the team started to identify the possible causes of the problem, regarding each one of the standard categories: Materials, Methods, Man Power, Environment, Measure and Money (See Figure 41). After an overall analysis of the causes identified in each category, the team drafted a new possible plan to address this causes, using a 5H2H framework.

5 Why's

The 5 Why's was another tool used during the Root-cause Analysis Meeting, that provided a fact based and structured approach to identify the causal link between the problems and their root causes. Like its name suggests, the team question five times consecutively why the problem was occurring, until they reach its root cause. An application of 5 why's analysis during a Root-cause Analysis Meeting of the A&V team can be seen in the Figure 42.

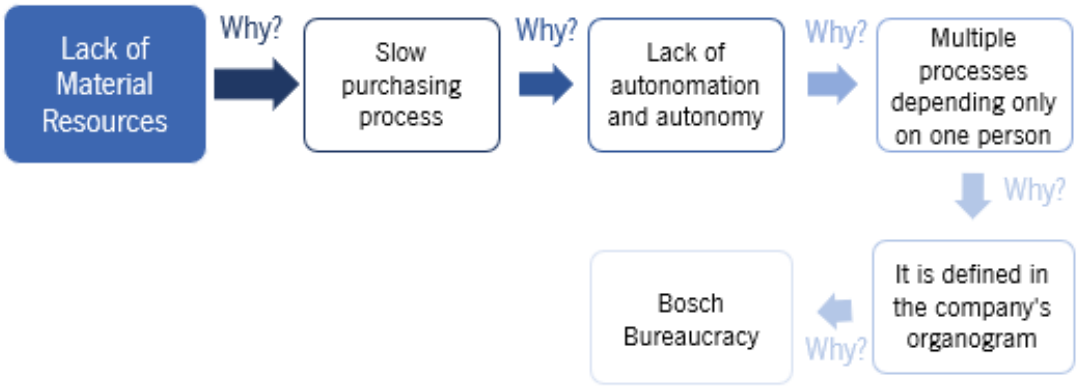

Figure 42 - 5why's Analysis from A&V Team

After the identification of the root cause analysis the team should follow with a corrective action, in order to prevent problems from occurring again.

5W2H

Undefined Profiles Needed

What?	Why?	Where?	When?	Who?	How?	How much?/ How many?
-Define long term goals and necessary competencies - Identify profiles - Identify profiles priority	- Is necessary to reinforce the team in key areas - More topics than developers	- Gen 2 A&V	ASAP (Before the end of April, profiles must be defined and approved)	To be defined	• Prepare Competencies Matrix • Write technical profiles • Align with ENG-P HC • Work with HR	Improve performance of the team

Figure 43 - 5W2H Framework from A&V Team

The 5W2H tool was used by the team to draft a new plan after discovering the root-cause of a problem that led to the failure of the implementation of an Improvement User Story. This tool promoted a brainstorming within the team to define what will be done, why and where it will be done, who will be the responsible, how the problem will be solved, and how much the team will benefit from it. An application of this tool by the A&V team can be seen in Figure 43.

CI Board

The CI Board is an artifact created by the Scrum Master, as a replication of the OPL. This board was placed in the team's open space, to improve the visual management. With this artifact, every team

member could easily perceive the current state of a problem and how many problems had already been solved. The CI board was also a good way to represent the achievements of the team regarding the work done with the continuous improvement scope (See Figure 44).

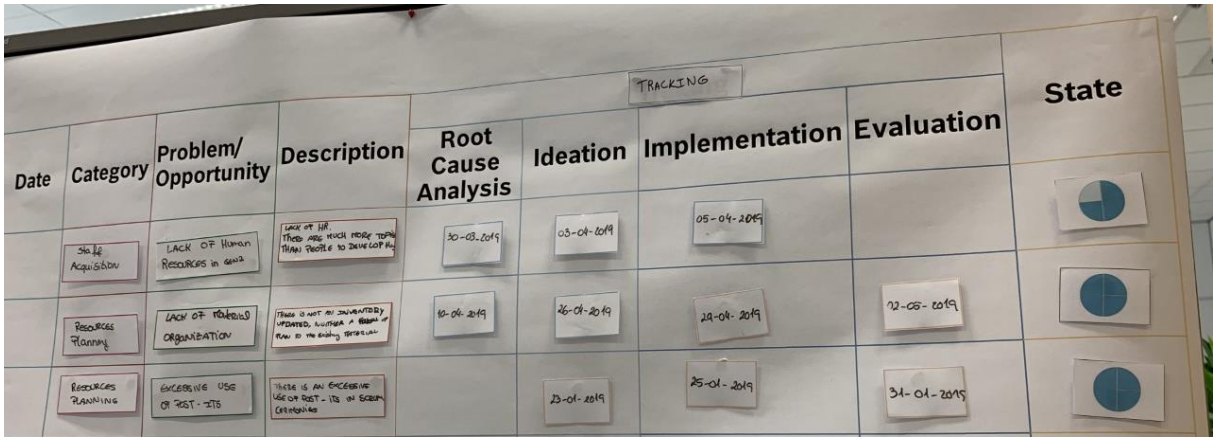

Figure 44 - CI Board from A&V Team

3.4 Complementary Work

As mentioned in the section 3.1, communication problems represented a big challenge within distributed teams, leading to lack of trust and visibility. All the interaction between the Algo team and the other teams in USA and Germany was extremely complicated. During this project, Braga Agile team composed for two Scrum Masters and one communication facilitator, applied some techniques during the global and local meetings, and worked together with other agile coaches from the other locations in order to improve communication, identify dependencies, clarify some Scrum's concepts and procedures and discuss the big picture, in order to mitigate some of the problems identified previously.

3.4.1 Video Conference

The first step to promote a healthy trust between the virtual team members, to work collaboratively was the introduction of video conference, during the global meetings, where the alignment between teams was made. The introduction of the video technique allowed the capture of visual aspect of communication, like body language, increasing the connectivity during the dialogues.

3.4.2 Team Building Activities

Subsequently, the Agile team organized and monitored some team building activities for the Algo team, to improve the relationship between the local team members, and one global team building session, to include all the teams in the project.

At the local level, one of the team building activity was called “Guess the artist” and consisted in the following:

1. Before the meeting, it was asked to each member of the team to create an image or find a picture that they could use to tell a unique story or fact about their lives.
2. Everyone putted their image on a white envelope.
3. In the opening of the meeting, the agile team spread the images on the board, and everyone voted, predicting whom each picture belongs to.
4. Afterwards, when the guesses were made, the person shared their story with the group.

The result of the Guess the Artist activity performed by the Algo team can be seen in Figure 45.

Figure 45 - Team Building Activity from Algo Team

Moreover, other team building activities were also made, like board games sessions e.g., Party&Co or the Collaborative Face Drawing activity.

Then on the international level, a team building session was organized to bring the distributed teams together and find points that could bring them together despite the different cultures. Once it was the first time that the teams were having a team building activity, and having in consideration the number of total participants (twenty-seven elements), the activity was divided into two steps. Firstly, with the support of video conference, each element seated in front of the camera and quickly made a presentation of himself. Then it was proposed to each participant to do a small power point presentation about their background, role on the team, hobbies, what was the best time to contact them during the work and by witch platform they preferred to be contacted. They should also add a photo of themselves. Further, the participants were challenged to answer the question: “If your life was a movie, what would the title be?”. Due to the

high number of participants these presentations were shared in an online platform, where everyone had access to it, and could easily check who was who.

This activity only counted with the participation of the teams in USA and the Algo team, since the teams in Germany did not want to participate.

3.4.3 Ice Break Activities

As mentioned in section 3.2, the Sprint Retrospective ceremony deserved a special attention during this project. Once it was the moment where the team members could identify the problems and suggest creative ideas to improvement, the team must be in an environment where they feel comfortable in sharing their true opinion, and that promote team creativity.

In this sense, before the Sprint Retrospective ceremonies, the Agile team prepared some ice break activities to break down some barriers, and the team members could start to evaluate the last Sprint making an analogy to some object. This technique improved the criticism because it was easier for them to talk about the things that went wrong without having the fear of being rude.

LEGO Activity

During LEGO ice break activity, Lego parts were placed above a table, and all the team members should pick the necessary pieces to create some Lego construction that represented the worries and challenges faced during the last sprint and the ones that are still yet to come.

Figure 46 – Lego Activity Performed by the Algo Team

Figure 46 depicts a lego's construction made by two elements from the Algo Team during this ice break activity. In the central picture, the team member created a big dinosaur with small hands, an archery and fire. The big dinosaur represents the big project that the team was working on and the small hand the hardness that was to reach the project goals. The archery ready to shoot to destroy the dinosaur, represents a project's developing feature that eventually would start causing problems and could jeopardize the project. Lastly, the fire near the dinosaur is an analogy to the mess existing in the project.

On the other hand, the team member responsible for the picture on the right, created a flight of stair with a close door in the end, and a man with a gift in his hands, walking in the stair's direction. The stairs represented the improvement on the algorithm's accuracy. The man with one leg bigger than the other represented the team with difficulties to achieve the expected results, and the gift on his hands was an analogy to the algorithm that the team was developing. Lastly, the door represents the happiness of the client, and it is always closed because on that team member's opinion the client is never satisfied.

Ice Cream Analogy

During this activity, the team was asked to answer the question: "If the Sprint was an ice cream how would it be?". Next, each team member should draw an ice cream that represented the last Sprint, and then give a brief presentation of the drawing to the rest of the team and place the draw in a happiness scale, accordingly how the team member felt about the overall Sprint. The results of this activity can be seen in Figure 47.

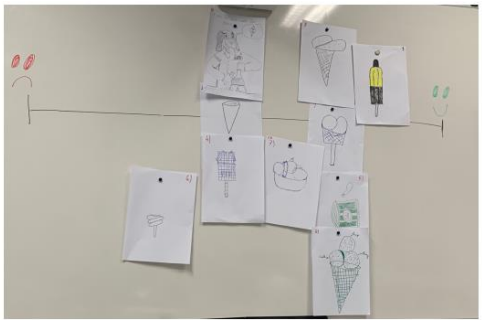

Figure 47 - Ice Cream Analogy Activity Performed by Algo Team

How Do You Feel?

"How Do You Feel" activity, was the most performed ice break activity during this project. In the opening of the Sprint Retrospective ceremony, the Scrum Master distributed for each element eight emojis that expressed the following feelings: Happy, Excited, Confused, Scared, Sad, Frustrated, Exhausted and Sick. Then the Scrum Master asked to the team: "How do you feel about the last Sprint?" and each element chose the emoji that best portrayed his feeling and share his reaction to the rest of the team (See Figure 48).

Figure 48 - Performance of the Activity "How Do You Feel?"

Happiness Radar

This activity was a useful way to open the Retrospective ceremony, narrowing down its context. The team first heard about people's feelings for each one of the topics discussed, and only after that the team followed for the data gathering during the Retrospective. Happiness Radar consists as following:

1. Decide and draw the target areas to collect feedback on happiness, as table column titles. "*For the given context, we would like to know your feelings for each of these areas".*
2. Draw the happy, ok and sad faces on the board, as table rows titles.
3. Ask the team to place their marks on the board. "*For each of the areas, let us know how you felt in average during the last Sprint*. For example*, if you were always happy regarding people, please make a one mark on the people-happy cell*" (Caetano, Caroli, & Ramos, 2016).

In this case the areas evaluated were: People, Technology, Intern Process and Scrum, but they can be customized accordingly the team's needs (See Figure 49).

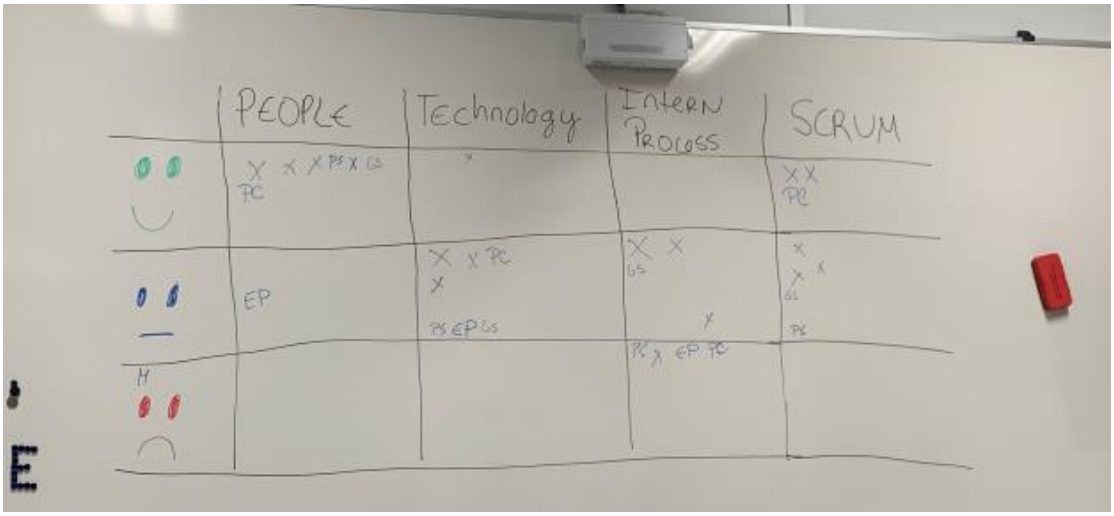

Figure 49 - Happiness Radar Performed with Algo Team

3.4.4 Retrospective Activities

In this section are presented different Retrospective techniques applied in the Algo team in order to create the right environment, increase the team's engagement and obtain better results from the ceremony throughout the time.

Starfish

Starfish is a data gathering activity to stimulate a brainstorming regarding the team's practices and their effects on the team. It allows team member to share their opinion on such practices, and give some improvement suggestions. To run the activity the following steps were done:

1. Split the board in five areas:

   Keep Doing – something that the team is doing well and should me sustained due to its beneficial effect;

   Less Of – something that already being done but is preferable to cut a little bit;

   More Of – something that already being done, and it will be better if done even more;

   Stop Doing – something that the team is doing but is not bringing value to the team;

   Start Doing – a new idea to implement, or to improve something.

2. Ask the team members to individually write notes on post-its for each of the areas.
3. Each team member stands up, placed their notes in the board accordingly and share with the team his opinion regarding each area.

The output of this activity performed by the Algo team can be seen in Figure 50.

Figure 50 - Starfish Retrospective Performed by Algo Team

Keep Doing, Stop Doing, Creative Solutions

This Retrospective activity has de target to identify the positive and negative aspects occurred during the Sprint, and then define things for the team to keep doing, stop doing, and finally give improvement suggestions (See Figure 51). The running of this activity follows the next steps:

1. Divide the board into two sides, one for the positive aspects and another for the negative ones;
2. Ask each element to identify and write down on post-its at least two positive and negative aspects occurred during the Sprint;
3. Each element places the post-its on the board, accordingly;
4. The Scrum Master creates clusters within the positive and negative aspects, with the help of the team.
5. Each element gives two votes in the cluster that he thinks that had influenced the Sprint the most, and one vote for the second cluster more influencing, for the positive and negative clusters;
6. Ask each element to identify and write down on post-its things that the team is doing well and that should keep doing;
7. Ask each element to identify and write down on post-its things that the team is doing but it will be better to stop doing;
8. Ask each element to identify and write down on post-its, improvement actions, to address the needs of the most voted clusters;
9. Each element places the post-its on the board and shares his opinion with the team.

Figure 51 - Keep Doing, Stop Doing, Creative Solutions Activity performed by Algo Team

Hot-air Balloon – Bad Weather

This activity is described by Paulo Caroli and Tainã Caetano in their book *Fun Retrospectives* as "forward-thinking exercise, with an eye on the past. It is a mix of retrospective and future perspective, which uncovers risks." (Caetano et al., 2016).

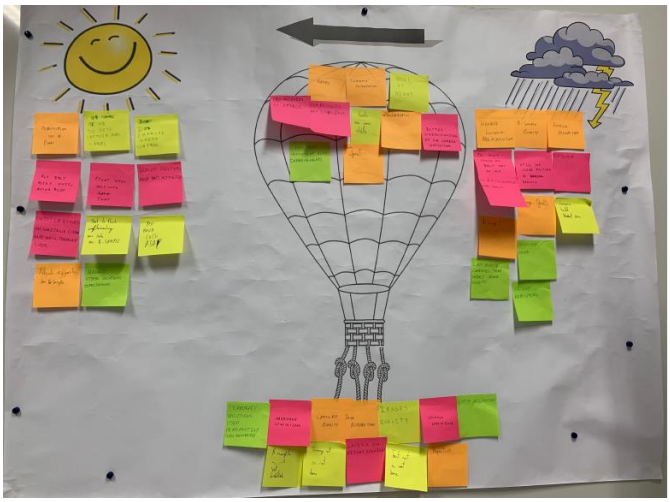

Figure 52 – Hot-air Balloon - Bad Weather Activity Performed by Algo Team

To perform this activity, the Scrum Master elaborated a big drawing with a Hot-air Balloon picture with forces pulling it down, a storm ahead the balloon and a sun behind it.

The balloon represents the past, the last Sprint. The fire and the hot-air are an analogy for the things that occurred during the Sprint that helped the team in going higher and pushed them forward. The forces pulling down the balloon represents the things that were blocking or slowing down the team.

Next, looking for future are pictured a storm ahead the balloon, representing the risks and challenges that the team will face and in the opposite side is drawn a sun, symbolizing the things that should be done to overcome the challenges ahead, avoid the storm and turn toward sunny days.

Each participant was asked to share their notes with the team regarding each of the retrospective areas (See Figure 52).

Three Little Pigs

Another activity performed during this project was the Three Little Pigs retrospective. This one was especially very fun because it used the Three Little Pigs story to cultivate a conversation about improvement for getting the structure of the team more solid.

This activity consisted as following:

1. Present and explain to the team the 3 columns:

- House of straws: what do the team does that just about hangs together, but could topple over at any minute? (e.g. "*our deployment script is very manual, and promote to error*");
- House of sticks: what do the team does that is pretty solid, but could be improved? (e.g. "*our automated tests are pretty good, but sometimes fail for no reason, and we have to run them again*");
- House of bricks: what do the team does that is rock solid? (e.g. "our automated deployment has never failed. It rocks").

2. Ask at each team member to share their notes on post-its and place them on the board accordingly.
3. Discuss and cluster the notes.
4. Each team member votes in the notes that most influence the team structure.
5. Improvement suggestions are given to the three most voted topics, and the plans for improvement are defined.

The result of the performance of this activity during a Sprint Retrospective ceremony of the Algo team can be seen below in Figure 53.

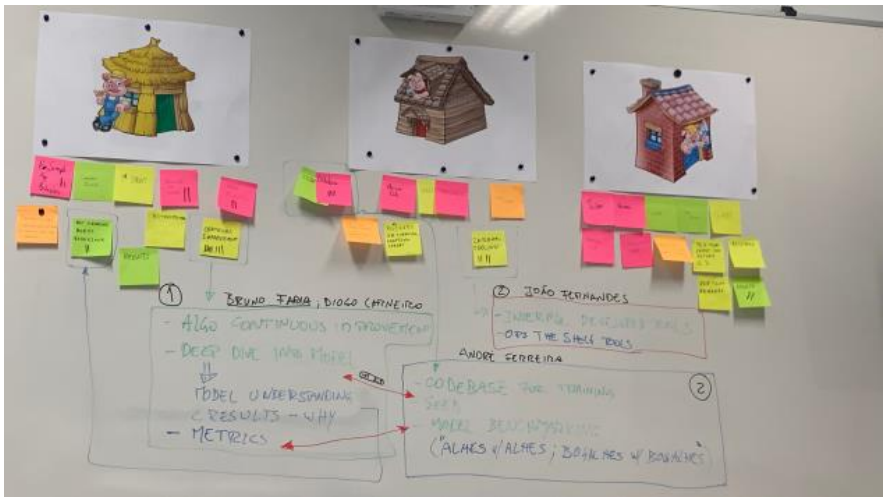

Figure 53 - Three Little Pigs Activity Performed by Algo Team

3.4.5 Check-out Activities

The check-out activities were performed right after the ending of the data gathering activity at Sprint Retrospective ceremony. During these activities, the Scrum Master gathered information about the meeting itself: how valuable for the team it was, what did they learned, how worthy was and got any specific feedback.

Return of Investment (ROI)

This activity was performed for closing a Sprint Retrospective ceremony of the Algo team. After the Scrum Master has drawn a scale for the ROI radar, from very little ☹ - to very high ROI 😊, the team members were asked to write down on a post-it their feedback about the Retrospective ceremony they had just held. "*For this last hour that you spent in this meeting, how do you measure the return of investment?* This was really worth my time: goes on top; the opposite goes at the bottom". Then each member should stand up and place his post-it on the ROI radar. The result of an ROI activity can be seen below in Figure 54.

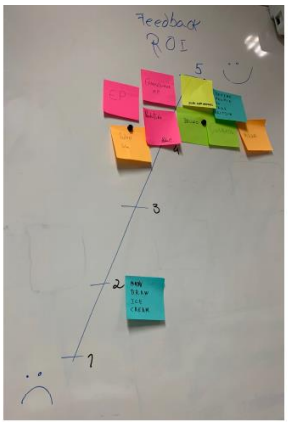

Figure 54 - ROI Activity Performed by Algo Team

Feedback

The feedback activity was also a very simple activity performed for closing the Sprint Retrospective meeting, that consisted in asking at each team member: "*How do you feel after this meeting*?". Then every participant should write down on a post-it an adjective that reflected their feeling on that moment and share their notes with the rest of team. In the image below are pictured the results of two feedback activities, and answers like: "*Satisfied, Hopeful, Worried, Puzzled, Confident and Excited*" can be found in Figure 55.

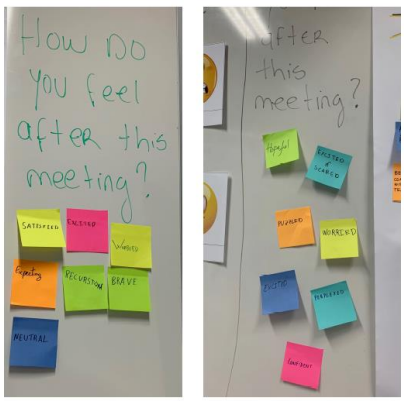

Figure 55 - Feedback Activity Performed by Algo Team

4. RESULTS

This chapter exposes the results obtained during this project, from the collaborative relationship between the Continuous Improvement Process and the Scrum framework implemented, on a software development team, as well as the main benefits of this implementation. Then a synthesis of the results achieved throughout this dissertation project is presented. In order to understand the opinion of the employees, the results of the surveys done by the A&V team, regarding the adoption of Scrum and CI process activities and artifacts are analyzed in detail and compared with the results of the Algo team – another software development team where the CI process was not implemented. Lastly, there's a short discussion of the overall results.

4.1 Problem-solving

One of the goals of this dissertation project was to improve the problem-solving capability of the team. In this section are presented the problems solved in each iteration of the development of the CI process.

All the Improvement activities presented in this section were carried out by the team itself, in some cases with the help of the Scrum Master. During the sprint, the Improvement User Stories were developed alongside the daily work of the team, creating the opportunity for the organization to benefit directly from their employee's ideas and involvement. Through the implementation of small improvements sustained throughout time, it was possible to achieve major benefits without making a big investment.

4.1.1 Iteration 1

In this section are exposed the problems that were solved by the team during the first iteration of the development of the CI process, when introducing Scrum (See Table 5). These problems were solved just by applying the Retrospective ceremony.

Table 5 - Problems Resolved During the First Iteration by A&V Team

Problem	Corrective Action Implemented
Excessive use of post-its	Performance of online Retrospective with no use of post–its
Lack of results' documentation	Creation of Sprint Review wiki page, with a template to document the results of every US.
Forgetting to document results in advance before meetings	Add the responsibility to Scrum Master to send an e-mail in the day before the meeting, to every team member, remembering them to document the results.
Weekly Meetings too long	Time-box meetings introduced
Lack of standard procedures during the meetings	Elaboration of guidelines for each meeting

4.1.2 Iteration 2

In this section are exposed the problems that were solved by the team during the second iteration of the development of the CI process, when introducing the first proposal of the CI process (See Table 6).

Table 6 - Problems Resolved During the Second Iteration by A&V Team

Problem	Corrective Action Implemented
Team members' delays on meetings	The Scrum Master was responsible to add an automatic warning to the meetings invitations, that would notify the team 10 min before the upcoming meeting.
User Stories badly defined	for every US write the respective Acceptance Criteria's and Definition of Done.
Delay the beginning of meetings, waiting for people that ended up not being present	Was stipulated a rule within the team that when a member knew in advance that he wouldn't be present at a meeting, he should notify the entire team in time, via e-mail.
Messy cupboard with shareable materials	Implementation of 5S in the office's cupboard and the inventory was updated.
Too many unfinished User Stories	WIP User Stories were limited. Every team member could only work at one US at a time.
Lack of feedback from the project leader	The Project leader started to be invited to Sprint Reviews and Sprint Retrospectives.
Lack of resources to record scenes for VICON system	Team went to Germany to do the recordings with the right equipment.
Lack of room to record scenes for VICON system	The team contacted some entities, and got a room in CCG – University of Minho
Lack of people to participate in VICON's recordings	Scrum Master was able to find volunteers to participate in the recordings.
Lack of materials to execute VICON's recordings	Scrum Masters elaborated a list of the necessary missing items necessary and bought them.
Lack of human resources	It was elaborated a list of the profiles needed and the project leader hired people to integrate the DevTeam.
Team members' overload	A Project Manager was hired to help in planning of the project
Lack of demos during Sprint Review	The team was encouraged whenever possible, to make the sprint's results more visual and to present them in video/graph format during Sprit Review.
Lack of an integration process for new team members	The onboarding process was defined and were created welcome activities to perform with the team and the new members.

4.1.3 Iteration 3

During the third iteration, it was presented the second proposal of the CI process, with the introduction of the Selection on Priority Suggestions Meeting and the OPL in the Improvement Meeting. The problems solved or in process of resolution during this iteration can be consulted in the OPL represented below (See Figure 56).

Creation Date	Problem/Opportunity	Date	Tracking								Status	Comment/ Process Description
			Plan		Do		Check		Act			
4-Jun-19	Difficulty in contacting the team members that were in another location	Real			5-Jun-19		18-Jun-19		18-Jun-19		Resolved	Add the link to the Skype conference call to meeting invitation.
		Planned			5-Jun-19		18-Jun-19		18-Jun-19			
18-Jun-19	Lack of cummuication devices	Real			19-Jun-19		2-Jul-19				Resolved	Scrum master got new devices after talking with IT partner.
		Planned			21-Jun-19		2-Jul-19					
18-Jun-19	Lack of people for labelling	Real	18-Jun-19		24-Jun-19		2-Jul-19		2-Jul-19		Resolved	All the labeling tasks were distributed for all the team members, including Scrum Masters.
		Planned	18-Jun-19		3-Jul-19		9-Jul-19		9-Jul-19			
2-Jul-19	Lack of people for audio-labelling	Real	2-Jul-19		4-Jul-19						In Progress	Interview and hire people for make a behavior and audio analysis of the recordings.
		Planned	16-Jul-19		17-Jul-19							
2-Jul-19	Bad communication between teams	Real	2-Jul-19		5-Jul-19		16-Jul-19		16-Jul-19		In Progress	Schedule regular sync meetings with external teams.
		Planned	2-Jul-19		16-Jul-19		30-Jul-19		30-Jul-19			
2-Jul-19	Bad quality of open space's air	Real	2-Jul-19		10-Jul-19		16-Jul-19		16-Jul-19			Schedule meeting to define problems, list the problems and the register the days where the team felt the air more heavy and sent request to the department leader to do an air analysis.
		Planned	16-Jul-19		22-Jul-19		30-Jul-19		30-Jul-19			
16-Jul-19	Ineficient labeling tools	Real	16-Jul-19								In Progress	Search for new and beter labeling tools, and acquired them.
		Planned	16-Jul-19		30-Jul-19		27-Aug-19		27-Aug-19			
9-Apr-19	Lack of an appropriate local to work with cars	Real	23-Apr-19		17-May-19						In Progress	A garage was acquired but it still under construction.
		Planned	7-May-19		30-May-19							
16-Jul-19	Lack of team building activities	Real	16-Jul-19		19-Jul-19						Resolved	A paintball activity was planned and scheduled and realized.
		Planned	16-Jul-19		26-Jul-19							

Figure 56 - Problems Resolved During the Third Iteration by A&V Team

4.2 Results Synthesis

In order to assess the improvements achieved with the implementation of the CI process, the researcher counted the improvement suggestions given and implemented by the team in each iteration. Considering Equation 1, it is possible to conclude that with the application of the first proposal of the CI process, the suggestions implementation rate increased by more than double, achieving thirty-six per cent, when compared with the results obtained in the first iteration, only applying the Sprint Retrospective. These results contributed for the increasing of the problem-solving capability of the team. When comparing the results of the second with the third iteration, a slight decrease of five percentage points is noted, but this result may have been influenced by the short duration of the third iteration, which was forced to end by the end of the researcher's internship in the company (See Table 7).

Equation 1 – Suggestions Implementation Rate

$$Suggestions\ Implementation\ Rate(\%) = \frac{Nº\ of\ Improvement\ Suggestions\ Implemented}{Nº\ of\ Improvement\ Suggestions\ Given} x100$$

Table 7 – Suggestions Implementation Rate

Iteration	N° of Improvement Suggestions Given	N° of Improvement Suggestions Implemented	Suggestions Implementation Rate (%)
Iteration 1	32	5	≈ 16%
Iteration 2	39	14	≈ 36%
Iteration 3	29	9	≈ 31%

As can be seen in Table 8, the number of hours necessary to perform every Scrum activity during one Sprint is five hours and thirty minutes, while the performance of the CI activities only requires the investment of one hour and thirty minutes in total.

Table 8 - Scrum and CI Activities and Their Respective Duration

	Activities	Duration	Participants		
			PO*	SM	DT
Scrum	Sprint Planning	1h	x	x	x
	Bi-Weekly Sync	30min*3 =1h30m	x	x	x
	Sprint Review	1h	x	x	x
	Sprint Retrospective	1h		x	x
	Sprint Backlog update	30 min	x	x	
	Total	5h30min	4h	5h30min	4h30min
CI Process	Selection of Priority suggestions	30 min		x	x
	Improvement Meeting	30 min	x	x	
	Root-Cause Analysis Meeting	30 min		x	x
	Total	1h30min	30 min	1h30min	1h

If this analysis is done from the developer's point of view, considering that in average, the developer works six hours per day and one Sprint has the duration of ten days, he only spends seven point five per cent of his working time per Sprint in Scrum activities, and about one point seven per cent in the CI activities (See Table 9). The percentage of hours required for a developer to perform the extra CI activities is very low, however it is enough to be translated into huge gains in the improvement of team's performance.

Table 9 - % of Hours Dedicated by Each Developer During One Sprint for Scrum and CI Activities

Working Hours per Sprint	Hours dedicated to Scrum activities	Hours dedicated to CI activities	% of hours dedicated to Scrum Activities	% of hours dedicated to CI Activities
6*10= 60 hours	4h30min	1h	7,5%	≈1,7%

The CI process brought multiple positive effects. The level of motivation, involvement, empowerment and teamwork within the team increased. Now, every time a team member is out of the office the team provides back up, usually another team member gives the updates regarding the work done by the absent member. There is a better and faster integration of new team members in the team. The welcome and ice breaker activities accelerate this integration process, and the documentation existing in the wiki regarding the topics developed by the team very much helps a new member, or the stakeholders, to see the bigger picture of the project. The monitoring of problem-solving and the work itself had increased, now with the utilization of the OPL every team member can have immediate perception of the current status of the problem, what problems have already been solved and which were the steps taken to solve them. The application of the Kanban board and the performance of the Bi-weekly Sync Meetings allowed every team member to know what the other elements are doing. The team developed better time management strategies, the time-box meetings and the limitation of the WIP User Stories per developer helped a lot. The communication within the team and between teams improved, the meetings were standardized, and the visual management increased a lot, with the use of Scrum and CI artifacts. In the current CI mindset, it is very easy for anyone to raise their preoccupations, give their true opinion and pursuit improvement. In the end, all these little achievements, when accumulated over time, create a drastic change, and resulted in successful demos, with the team receiving very good feedback from the clients (See Table 10).

Table 10 - Goals Achieved During the Project

Accomplished Goal	Due to...
More motivated Team	• Realization of Team building activities; • Successful Demos; • Sense of integration;
Increasing of Team member's involvement and empowerment	• Participation in the improvement process of the team; • Team responsible to prioritize improvement actions;
Better management and distribution of the work	• Documentation on wiki; • Kanban board application; • Backlog creation; • Sprint Planning adoption;
Promoted team member's creativity	• Performance of different approaches of Sprint Retrospective ceremonies;
Increasing problem-solving capability	• Performance of CI process;
Better time management	• Time-box meeting's implementation; • Limited WIP per developer;
More Teamwork	• Scrum implementation; • Shared sprint goal;
Mindset of Continuous Improvement	• Scrum cycle; • CI process;
Increasing Visual Management	• Scrum cycle; • CI artifacts;
More Successful Demos	• Demos tested during Sprint Review; • Achievement and Goals conquered during the sprints documented;
Better and more communication within the team and between teams	• Definition and implementation of welcome activities, for new members; • Team building activities; • Video-conference calls; • Link to Skype calls inside the meeting invitation;
Standardized processes	• Guidelines for each meeting; • Templates for meetings;
Increasing work monotoring	• Kanban board; • Bi-Weekly sync; • Wiki Results documentation;
Increasing problem-solving monotoring	• CI process (OPL, CI Board);

The improvements exposed in this section should be carefully interpreted as they may not result only of the implementation of the CI process. During the development of this dissertation project, there were some uncontrolled variables that could've also influenced the results, like the constantly changing of team members, company's bureaucratic processes and the budget released for the team. Also, the short time of the internship was not enough to test and evaluate the long-term results of the CI process implementation.

4.3 Team's Member Feedback

4.3.1 Scrum - Sprint Retrospective

As it was mentioned before, the Sprint Retrospective ceremony deserved special attention during this dissertation project. Besides this activity being included in the Scrum and in the CI cycle, it was the moment where the teams had the time and space to identify gaps and bad behaviours that should be stopped and on the other hand, spot good practices and routines that were beneficial to the team. During these meetings, the team's creativity was put to test, and they were challenged at every Sprint to find opportunities for improvement and come up with creative and new ideas to put into practice.

The outcome of the Sprint Retrospective ceremony was the input for the CI process, in such manner that the quality of the data gathered during this meeting would directly influence the performance of the CI process.

In order to understand the opinion of the teams regarding this meeting, a questionnaire was made and sent to every team member of both A&V and Algo teams. The questionnaire was split into two sections: the first one, questioned the meeting and the Scrum Master performance in general and the second one consisted of an open response space regarding the implemented improvement. For each question of the first part, there were five possible answers available, following a Likert scale: Very Satisfied, Satisfied, Neutral, Dissatisfied, Very Dissatisfied. The first part consisted of the following questions:

1- How do you feel about attending the Retrospectives?

2- Did the scrum master explain to you the purpose of this meeting and present you the rules, steps and the objectives?

3- Are the issues discussed being relevant and meaningful to the team?

4- Are retrospectives helpful to you?

5- How effective was the Scrum Master in stimulating the team participation during this project?

6- Are the meetings generating prioritized and actionable items for improvement?

All the nine elements from A&V and the seven from Algo team answered the questionnaire, and the analysis of its results evidenced a very positive perception regarding the adoption of Sprint Retrospective and the content discussed during that ceremony. As can be seen in Figure 57, thirty-two per cent of the inquiries from A&V assumes being very satisfied with this meeting, besides the fifty-seven per cent considering being satisfied, making a total of acceptance of eighty-nine per cent, opposed to a total of five

per cent that consider being dissatisfied of very dissatisfied. When looking to Algo team, the results are very similar, getting a total of eighty per cent of satisfaction regarding the adoption of the meeting.

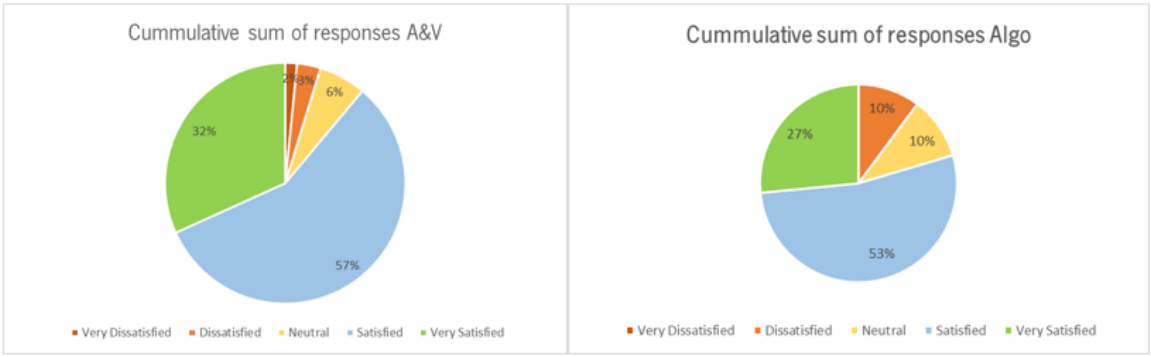

Figure 57 - Cumulative Sum of Responses from A&V and Algo Team Questionnaire's Results

When comparing the results of the question: "Are retrospectives helpful to you?" between the two teams, a big difference can be noticed. Even though the Sprint Retrospective ceremony is much simpler in A&V team, it obtained a total of one hundred per cent positive feedback, considering that this ceremony is helpful for them, with eleven per cent assuming being very satisfied and eighty-nine per cent considering being satisfied. However, the results of the surveys done to Algo team show that although the Scrum team had invested a lot in improving this ceremony, the opinions are not consensual (See Figure 58). This could take us to the challenge of this dissertation project, where it was exposed the problem of the inefficiency in transform the retrospective's ceremony output into real improvements.

Figure 58 - Comparation between A&V and Algo Team Questionnaire Results – Question 4

Figure 59 shows that besides the overall satisfaction concerning the relevance and meaning of the topics discussed, the A&V team is also convinced that the Sprint Retrospective is generating prioritized and actionable improvements to put into practice, that was one of the main goals pursued with the implementation of the CI process.

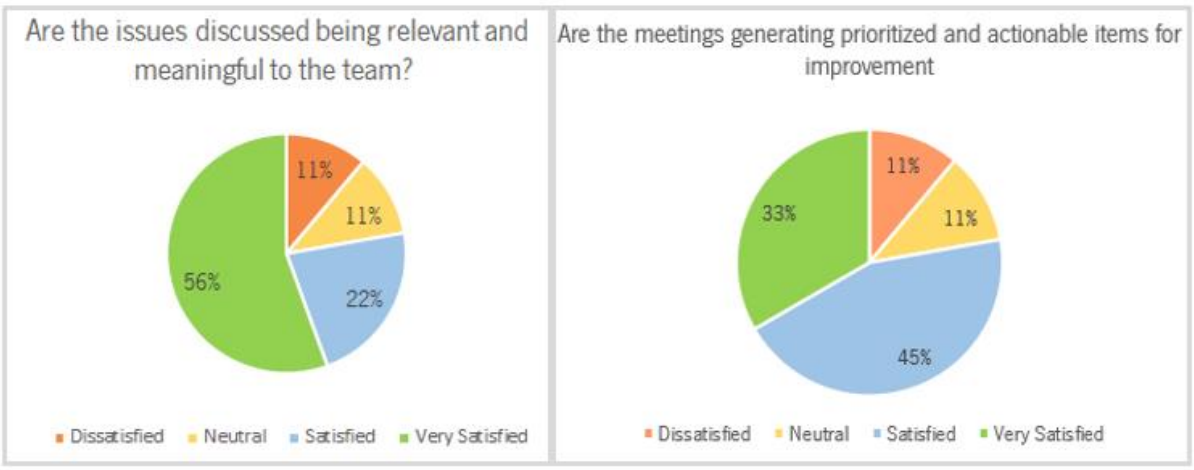

Figure 59 - Results of Question 3 and Question 6 from A&V Team

In fact, comparing these results with the outcome obtained with the analysis of the same questionnaire made to the Algo team, there is a considerable difference. While in the A&V team the percentage of team members who were very satisfied with the outcome of the Sprint Retrospective is thirty-three per cent, in the Algo team there is not even one element very satisfied, and the percentage of dissatisfied elements is more than double than in the A&V team (See Figure 60).

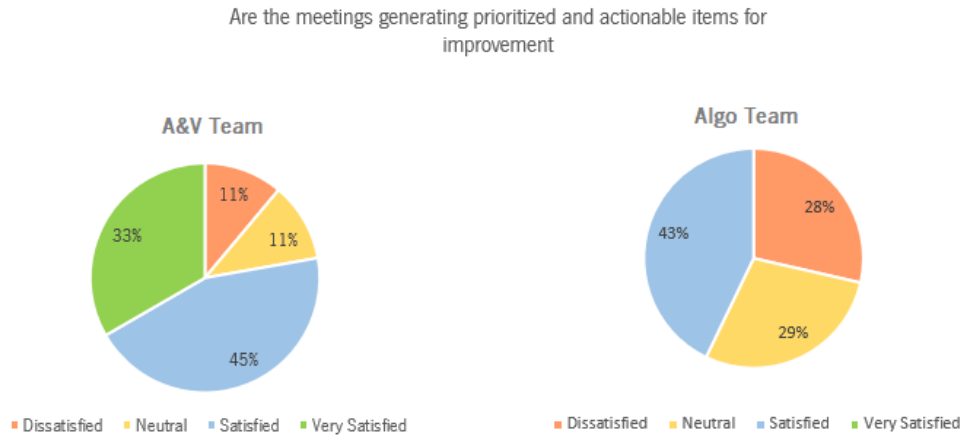

Figure 60 - Comparation between A&V and Algo Team Questionnaire Results – Question 6

Although when asked: "How do you feel about attending retrospectives?", seventy-eight percent of the A&V answered being very satisfied or satisfied, and eighty-six per cent of Algo's team members assumed being very satisfied or satisfied about attending the Sprint Retrospective ceremony, leading us to believe that the Algo team enjoys more their time during the Retrospective ceremony than the A&V team (See Figure 61).

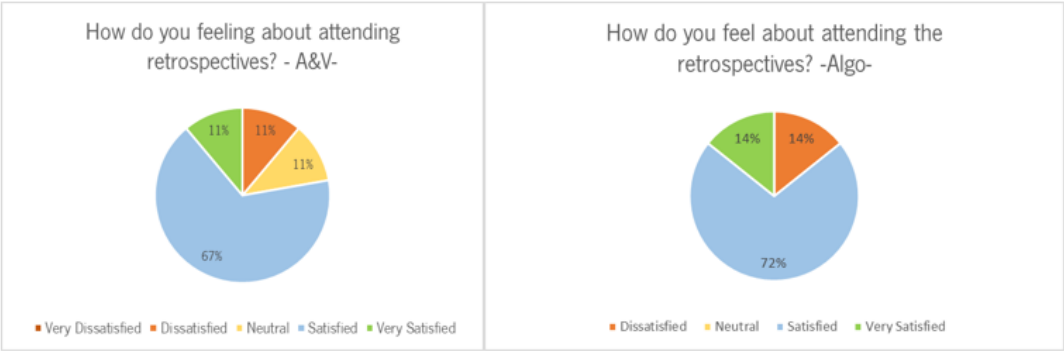

Figure 61 - Comparation between A&V and Algo Team Questionnaire Results – Question 1

Lastly, in the second section the teams were asked to mention:

1. What actionable improvement solutions have been implemented?;
2. What actionable improvement solutions have not been implemented?;
3. Do you have any suggestion for improve this meeting?

In Table 11 it's possible to see a big difference between the answered gave by two elements, one from A&V team and another to Algo team. The element of the Algo team, where the CI process was not implemented had a very negative feedback when compared to the feedback given by the A&V team member.

Table 11 – Feedback From A&V Team Member and Algo Team Member.

Respondent	Q1 - What actionable improvement solutions have been implemented?	Q2 - What actionable improvement solutions have not been implemented?	Q3 - Do you have any suggestion for improve this meeting?
A&V Member	• "*Organizations of the closets to improve the time management when looking for material;* • *Flying tasks have been reduced. In the beginning, these tasks disturbed a little the development of the main user stories."*	• *"The project leader still not being present on the retrospective meeting, he should be one per month at least."*	• *Sync meetings should be in a bigger room. The small rooms where we usually do the syncs are too small and get too hot, especially during the summer.* • *Every member should talk briefly about positives/Negatives. Writing is not enough! Focus more on what the team can actually solve.*"
Algo Member	•"*I can´t think of one. Maybe none."*	• "*Although everyone has spoken of "actions to improve", most of them have not been put to practice."*	• "*The outcome of this meeting should improve next sprint. So far, the outcome are problems in the sprint and possible problems in the next sprint. Good but not enough.*"

4.3.2 CI Process

In order to understand the opinion of the teams regarding the practices, values and applicability of the CI process, the researcher prepare a presentation to expose and explain the CI process to the Algo team and developed and also developed and sent a questionnaire to every team member of A&V and Algo teams. For each question were available five possible answers following a Likert scale: Strongly Agree, Agree, Neutral, Disagree, Strongly Disagree. The questionnaire consisted of the following questions:

1- The analysis of the problems identified in Retrospective Ceremony is important to improve the problem solving;

2- I consider important to give my improvement suggestions to the team;

3- I think that is important to prioritize the problems;

4- I think that is important to define the Root Cause of the problems, so the team can plan the solutions better;

5- I think that is important to monitor the progress of the problems;

6- I think that is important for me to be involved in the improvement process of the team.

Figure 62 - Cumulative Sum of Responses from A&V and Algo Team

Figure 62 depicts the results of the cumulative sum of the answers given by every team member from the A&V and Algo Teams. As can be seen in the figure above, the results are very similar between teams, and both very positive, leading us to believe that both teams agree with the practices and values of the Continuous Improvement process proposed during this dissertation project.

Below can be seen some team members feedback testimonials regarding the CI process from A&V team:

Member 1 - *"Good work!"*;

Member 2 - "*Good measures to identify problems in faulty processes, although bureaucratic processes put an halt to their resolution/improvement.*";

Member 3 - "*Pretty good feedback, with potential for improvements.*";

Member 4 - "*The ceremonies and interaction between team members created a great environment to work in. In the future retrospectives we should focus more (steer the team) to the problems we can actually solve and spend less time with the ones we can't do much about (ex. resources).*";

Member 5 – "*The review and retrospective meetings are important and has a clear impact on team working. However, we have too much syncs meeting. We could replace some sync meeting by technical meetings. It is important to have elements dedicated to prepare and moderate the sync, review and sync meetings. These elements are important in the problem solving process (identify and track solutions).*";

Following it is presented the feedback from the Algo team:

Member 1 – "*The solutions proposed on the feedback meeting are very appealing and interesting, being a shame that they weren't pursued. Overall the worked done in the previous months was good, however some improvements should be performed.*";

Member 2 – "*Shorter and more concise meetings. More feedback from the management regarding the problems pointed in the retrospective. But overall scrum brought more structure to the team making it easier to organize the context of the meetings.*";

Member 3 – "*Little exposure time to scrum to provide feedback. Nonetheless, from the little time I was able to observe I can say that I like the scrum process, since it can improve team dynamics and idea/problem discussion. In my experience this is a very important point as it may allow problems/ideas to be resolved or improved. Furthermore, it improves team spirit because it serves as an integration process for new members. Finally, in respect to the OPL it seems to me a good strategy since it allows to further breakdown problems, at least from what I could gather also considering that I don't even know how to implement it.*";

Member 4 – "*I think that overall SCRUM is ok. Although the problem-solving aspect of it was pretty lackluster when it comes to the retrospectives. Most of the retrospectives did not end with the problems identified and possible solutions, this aspect was improved on later into the project. The lack of a clear method in how to approach the identified problems was apparent, since the meeting ended with just a listening of the complaints and not a single action to solve the problem or reason to those complains identified. In that aspect the approach presented today could have helped and seems a good starting point towards a more effective retrospective. But it does need to be adapted to the team and project in question. Luckily the approach appears to be flexible enough to be adapted, which is a good thing.*"

Member 5 – "*Scrum has brought transparency and visibility to the development process. We have been encouraged by the scrum masters to get involved in many topics, such as identifying plans for future feature development and identifying pain points. Overall, the productivity has increased thanks to this. The presented process has potential to provide transparency and track easily the status of the black points, ensuring the process improvement topics are not forgotten.";*

4.3.3 Scrum Master

In the interest of perceiving the team member's opinion regarding the performance of the Scrum Master during this project, the teams were asked in the first questionnaire: "How effective was the Scrum Master in stimulating the team participation during this project?".

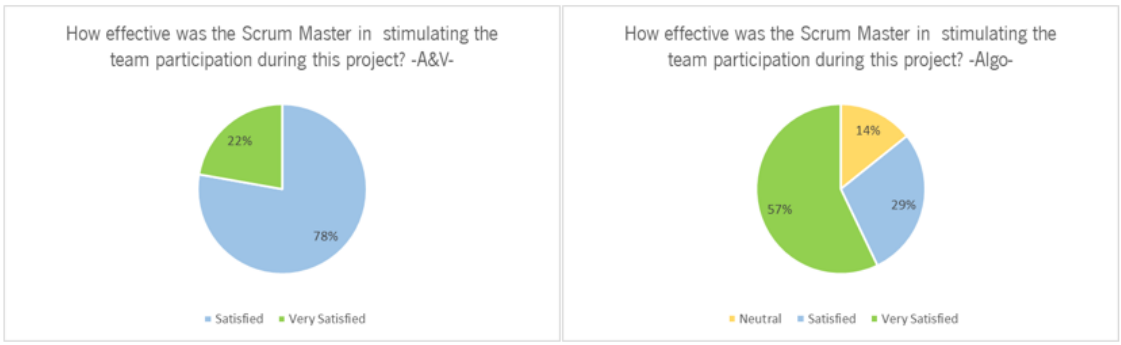

Figure 63 - Scrum Master Evaluation by Teams

The analysis of the results of these questions shows that both teams had very good feedback regarding the effectiveness of the Scrum Master during this project. Below can be consulted some feedback testimonials from the team members regarding the role of the Scrum Masters.

Member 1- "*Overall, all the work done by the scrum masters yielded positive results and added value to the team. The team members were not only more focused, organized and aware of each other's work, but also had a system to diagnose pain points and improve. Their work consisted in teaching the team the scrum's development cycle, supporting/planning all scrum activities and keeping meetings on track. This was key to make sure the process was well implemented and positive for the team.*";

Member 2 - "*In respect to the scrum masters I think they have correctly implemented the scrum process and played a major role in how the team responded to the scrum strategies, either by simplifying the process or by motivating the team to follow the scrum policies.*";

Member 3 – "*As for the Scrum masters, very good performance and enthusiasm in difficult tasks that crossed their path. A lot of patience performing a series of tasks that, by some, would be considered mind-bogglingly dull and they took it like champs! Nevertheless, they were incredible in all matters!*".

5. Conclusion and Lessons Learned

Nowadays, if technology companies aspire to sustain their competitive advantage, and follow the fast-changing market, they must pursue perfection and although being impossible to reach it, the effort to get there provides the drive and inspiration necessary to accomplish improvements along the way.

This dissertation project explored the implementation of a continuous improvement process on an agile software development team from a company that aims to become a world leader in IoT. The previous company's experience with Scrum recognized that most teams were inefficient in taking advantage of the improvement opportunities that arose during the Sprint Retrospective ceremonies and transform them into improvements for the team. The CI process was designed to be performed together with the Scrum cycle, and its implementation was incremental and gradual to be sustainable over time. The final proposal of the CI process provided the necessary conditions for the teams to be able to identify emerging problems, or opportunities for improvement, give and implement suggestions to address problems, monitor problems-resolution processes over time, identify root causes of those problems and assess their impact, and all these responsibilities were performed during the daily work of the team.

The results showed that the adoption of the CI process by the A&V team, after the Scrum implementation was finalized and standardized, doubled the suggestions implementation rate when compared with the previous results obtained by only performing the Retrospective ceremony, contributing for the increasing of the problem-solving capability of the team. Besides that, the accumulation of small improvements turned into big improvements throughout time: the level of motivation, involvement, empowerment and teamwork within the team increased; There is a better visual, task and time management; The communication improved, and a mindset of continuous improvement was built within the team. A more detailed analysis allowed to conclude that on average, one developer spends only 7,5% of his working time performing Scrum activities and about 1,7% in the extra CI activities. The percentage of hours required to perform the Scrum and the extra CI activities is very low, but enough to represent huge gains for the team without a big investment. These results support the idea that through its people, a company can be continuously better and get higher results.

During this dissertation project, many challenges were faced, not only by the Scrum Master but also by every software development team. As this project was developed in a very big company, even having total support from the project leader, it wasn't always easy to do and try new things in a practical way. The internal bureaucracies of the company made it difficult or worst, impossible, to change some proceedings

that needed improvement. As in every continuous improvement implementation, breaking the traditional mindset and the belief that "not everyone is creative", besides the challenges to acquire new behaviour patterns, the management of the short-term expectations and getting active participation of all team members were the big challenges experienced during this project. Although we tried, with resilience and teamwork, to overcome all the obstacles.

The fact that the project had understudy software development teams from the innovation company's department, made it very difficult to quantify the results obtained with the implementation of the CI process. As innovation teams, their biggest motivation was the exploitation of existing and new knowledge, and the qualitative nature of this area is hard to quantify. One sprint was never repeated so it was very difficult to compare the result between Sprints. Besides this, the A&V team decided not to punctuate their User Stories, so in this case, it wasn't possible to monitor the team's velocity.

During this project, the researcher worked together with the team to always find the best options and design a process that the team would accept and enjoy working with. In addition to performing the role of Scrum Master, the researcher participated actively in the process of problem-solving with the team.

The implementation of the CI process developed during this dissertation project received very good feedback from the team members. Beyond increased team performance, with all demos made by the team receiving excellent appreciations from the clients and stakeholders, according to a testimony of one team member, Scrum increased the team's happiness, and there is no more gratifying result than that.

"Scrum increased my productivity and Team happiness!" - Team Member 6

5.1 Future Work

Although the CI process showed good results, it is always possible to improve. So, the next step to take is to spread this continuous improvement culture through the department and implement the CI process in other teams. Adjusting the CI process to the needs of each team could bring big improvements to the department. The next implementations should previously define the key performance indicator, to be easier to quantify and monitor the results along time.

Referências Bibliográficas

Al-baik, O., & Miller, J. (2016). Kaizen Cookbook : The Success Recipe for Continuous Learning and Improvements. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5388–5397. https://doi.org/10.1109/HICSS.2016.666

Alves, A., & Leão, C. (2015). *ACTION, PRACTICE AND RESEARCH IN PROJECT BASED LEARNING IN AN INDUSTRIAL ENGINEERING AND MANAGEMENT PROGRAM Anabela*. (recommendation 4), 1–10.

Apcer. (2015). *Guia do Utilizador Iso 9001:2015*.

Baskerville, R., & Wood-Harper, T. (1996). A critical perpective on action research as a method for information systems research. *Journal of Information Technology*, *11*, 235–246.

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development.

Bengtsson, M. (2018). Transforming systems of consumption and production for achieving the sustainable development goals : moving beyond efficiency. *Sustainability Science*, *1*. https://doi.org/10.1007/s11625-018-0582-1

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

Bessant, J., & Francis, D. L. (1999). *Developing Strategic Continuous Improvement Capability*. (April 2016). https://doi.org/10.1108/01443579910291032

Bititci, U., & Nudurupati, S. (2002). Driving Continuous Improvement. *Manufacturing Engineer*.

Bosch Group. (2009). *Robert Bosch - His life and work*.

Caetano, T., Caroli, P., & Ramos, G. (2016). *Fun Retrospectives Activities and ideas for making agile retrospectives more engaging*.

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for Development: Guidelines for Process Integration and Product Improvement* (3rd editio; A.-W. E. P. Inc, Ed.). New Jersey, United States: Pearson Education (US).

Crosby, P. B. (1979). *Quality is Free: The Art of Making Quality Certain: How to Manage Quality - So That it Becomes a Source of Profit for Your Business* (1st editio). Michigan: McGraw-Hill.

Deming, E. (1993). *The New Economic for Industry, Government, Education*. Cambridge.

Deming, W. E. (1993). *Quality Productivity and Competitive Position*. Massachusetts.

Denner, V. (2015). *No Title*.

Denning, S. (2010). *The Leader's Guide to Radical Management* (I. John Wiley & Sons, Ed.). San Francisco.

Dinis-Carvalho, J., Ratnayake, R. M. C., Stadnicka, D., Sousa, R. M., Isoherranen, J. V, & Kumar, M. (2016). *Performance Enhancing in the Manufacturing Industry : An Improvement KATA Application*. 1250–1254.

Dyba˚, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(3), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Ferreira, A. L. (2016). *Methodological Approaches for Software Process Improvement in Multi-model Environments*.

Gallie, D., Zhou, Y., Felstead, A., & Geen, F. (2012). Teamwork, Skill Development and Employee Welfare. *British Journal of Industrial Relations*, *50(1)*, 23–46. https://doi.org/10.1111/j.1467-8543.2010.00787.x

Haddas, M. A., Asiri, M. H., Mukhalid, R. F., Alahmari, S. S., Al-qathtani, S. A., & Hasan, S. H. (2017). *Continuous Improvement – Development with Time*. (November). https://doi.org/10.5120/18935-0360

Halloran, P. (1999). Organisational Learning From the Perspective of a Software Process Assessment and Improvement Program. *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*, 11. https://doi.org/0.1109/HICSS.1999.772898

Haug, M., & Enz, V. (2011). *What is the relationship between coaching interventions and team effectiveness ?* (5), 89–101.

Hines, P., & Taylor, D. (2000). *Going lean*. Cardiff, UK: Peter Hines & David Taylor 2000.

Imai, M. (1986). *KAIZEN: The Key to Japan's Competitive Success* (McGraw-Hil).

International Standard. (2017). *ISO/IEC/IEEE International Standard - Systems and software engineering*. https://doi.org/10.1109/IEEESTD.2017.8016712

Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, *30*(10), 1–24. https://doi.org/10.1002/smr.1954

Kerth, N. L. (2001). *Project Retrospectives - A Handbook for Team Reviews*.

Khalifa, M., & Verner, J. M. (2000). Drivers for software development method usage. *IEEE Transactions on Engineering Management*, *47*(3), 360–369. https://doi.org/10.1109/17.865904

Khan, I. A. (2011). *KAIZEN : The Japanese Strategy for*. *1*(3), 177–184.

Kišš, F., & Rossi, B. (2018). Agile to lean software development transformation: A systematic literature review. *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018*, *15*, 969–973. https://doi.org/10.15439/2018F53

Kozak-Holland, M. (2011). The History of Project Management (Lessons from History). *Journal of Product Innovation Management*, 528. https://doi.org/10.1111/j.1540-5885.2012.00969_3.x

Kumar, S., & Harms, R. (2004). Improving business processes for increased operational efficiency: A case study. *Journal of Manufacturing Technology Management*, *15(7)*, 662–674. https://doi.org/10.1108/17410380410555907

Liker, Jeffrey. (2004). *The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer* (Vol. 2004; I. CWL Publishing Enterprises, Ed.). New York: McGraw Hill.

Liker, JK. (2005). *The Toyota way : 14 management principles from the world's greatest manufacturer*. McGraw Hill.

Linderman, K., Schroeder, R. G., Zaheer, S., & Choo, A. S. (2003). *Six Sigma : a goal-theoretic perspective*. *21*, 193–203.

Marx, A. (1995). Management commitment for successful suggestion systems. *Work Study*, *44*(3), 16–18. https://doi.org/10.1108/00438029510085357.

Ming Huo, Verner, J., Liming Zhu, & Babar, M. A. (2004). *Software quality and agile methods*. (June 2014), 520–525. https://doi.org/10.1109/cmpsac.2004.1342889

Moica, S., Harea, C. V., & Marian, L. (2018). Effects of Suggestion System on Continuous Improvement : A Case Study. *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 592–596.

Morien, R. (2005). Agile management and the Toyota way for software project management. *2005 3rd IEEE International Conference on Industrial Informatics, INDIN*, *2005*, 516–522. https://doi.org/10.1109/INDIN.2005.1560430

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78. https://doi.org/10.1145/1060710.1060712

Ohno, T. (1998). *Toyota Production System - Beyond Large-Scale Production*. Productivity Press.

Paasivaara, M., & Lassenius, C. (2014). Deepening our understanding of communities of practice in large-scale agile development. *Proceedings - 2014 Agile Conference, AGILE 2014*, 37–40. https://doi.org/10.1109/AGILE.2014.18

Petersen, K., Wohlin, C., & Baca, D. (2007). The Waterfall Model in Large-Scale Development. *Product-Focused Software Process Improvement*, (May). https://doi.org/10.1007/978-3-540-73460-4

Popova, N., Kryvoruchko, O., Shynkarenko, V., & Zéman, Z. (2018). Enterprise management in VUCA conditions. *Economic Annals-XXI*, *170*(3–4), 27–31. https://doi.org/10.21003/ea.V170-05

Project Management Institute. (2017). *A Guide to the project management body of knowledge (PMBOK guide) sixth edition/ Project Management Institute.* (Sixth edit). Newtown Square, Pennsylvania: Project Management Institute, Inc.

Ribeiro, F., Tereso, A., Deborah, D., & Andrade, P. De. (2017). *Development of a grooming process for an agile software team in the automotive domain*. Universidade do Minho.

Robert Bosch GmbH. (2015). About the Mission Statement "We are Bosch." Retrieved July 3, 2019, from Robert Bosch GmbH website: https://inside.bosch.com/irj/portal/?NavigationTarget=HLPFS://wcms_c_Bosch_20GlobalNet/wcms_c_01_20About_20Bosch/wcms_c_mission_statement_we_are_bosch/wcms_c_01_about_mission_statement_we_are_bosch/wcms_c_about_mission_statement_we_are_bosch&NavigationCon

Robert Bosch GmbH. (2019a). About CM - Car Multimedia. Retrieved July 17, 2019, from https://inside.bosch.com/irj/portal/?NavigationTarget=HLPFS://wcms_cm_Bosch_20GlobalNet/wcms_cm_02_20Organization/wcms_cm_Automotive_20Technology_20UBK/wcms_cm_organization_bbm/wcms_cm_bbm_units/wcms_cm_CM/wcms_cm_cm_about

Robert Bosch GmbH. (2019b). The beginnings 1886-1905. Retrieved July 2, 2019, from Robert Bosch GmbH website: https://www.bosch.com/stories/1906-1925-globalization-and-new-beginning/

Robert Bosch GmbH. (2019c). The Bosch Group at a glance. Retrieved July 3, 2019, from Robert Bosch GmbH website: https://www.bosch.com/company/our-figures/

Rother, M. (2010). *Toyota kata*.

Schmidt, M. T., Elezi, F., & Lindemann, U. (2014). *Towards Recursive Plan-Do-Check-Act Cycles for Continuous Improvement*. 1486–1490. https://doi.org/10.1109/IEEM.2014.7058886

Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide ™*. (November).

SCRUMstudy. (2017). *A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK ™ GUIDE) Third Edition A Comprehensive Guide to Deliver Projects using Scrum Includes two chapters about Scaling Scrum for Large Projects and the Enterprise*. Retrieved from www.scrumstudy.com

Seymour, T., & Hussein, S. (2016). *The History Of Project Management*. (September 2014). https://doi.org/10.19030/ijmis.v18i4.8820

Singh, J., & Singh, H. (2015). *Continuous improvement philosophy – literature review and directions*. (May). https://doi.org/10.1108/BIJ-06-2012-0038

Susman, G. I., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly*, 582–603.

Sutherland, J. (2014a). *Scrum: The Art of Doing Twice the Work in Half the Time*. Penguin Random House.

Sutherland, J. (2014b). *SCRUM - The Art of Doing Twice the Work in Half the Time* (R. House, Ed.). 9781847941107.

Sutherland, J., & Heitz, H. (2011). *Scrum and Lean: How a Lean Scrum Can Improve Your Performance*.

Tonini, A. C., & Spinola, M. D. M. (2006). *Six Sigma and Software Development Process : DMAIC Improvements Six Sigma and software development process : DMAIC improvements*. (November 2014). https://doi.org/10.1109/PICMET.2006.296875

Tripp, F., Armstrong, J., & J., D. (2018). Agile Methodologies: Organizational Adoption Motives, Tailoring, and Performance. *Journal of Computer Information Systems*, *58*(2), 170–179. https://doi.org/10.1080/08874417.2016.1220240

Vasanthapriyan, S. (2018). Agile and Scrum in a Small Software Development Project- a Case Study. *Proceedings of 7th International Symposium*, 7. Sri Lanka.

Womack, J., & Jones, D. (1996). *Lean Thinking - Banish Waste and Create Wealth in Your Corporation* (2003 Editi). New York: A Division of Simon & Schuster, Inc.

Womack, J., Jones, D., & Roos, D. (1990). *The Machine that Changed the World*. New York: Macmillan Publishing Conipany.

APENDIX I – QUESTIONNAIRE 1

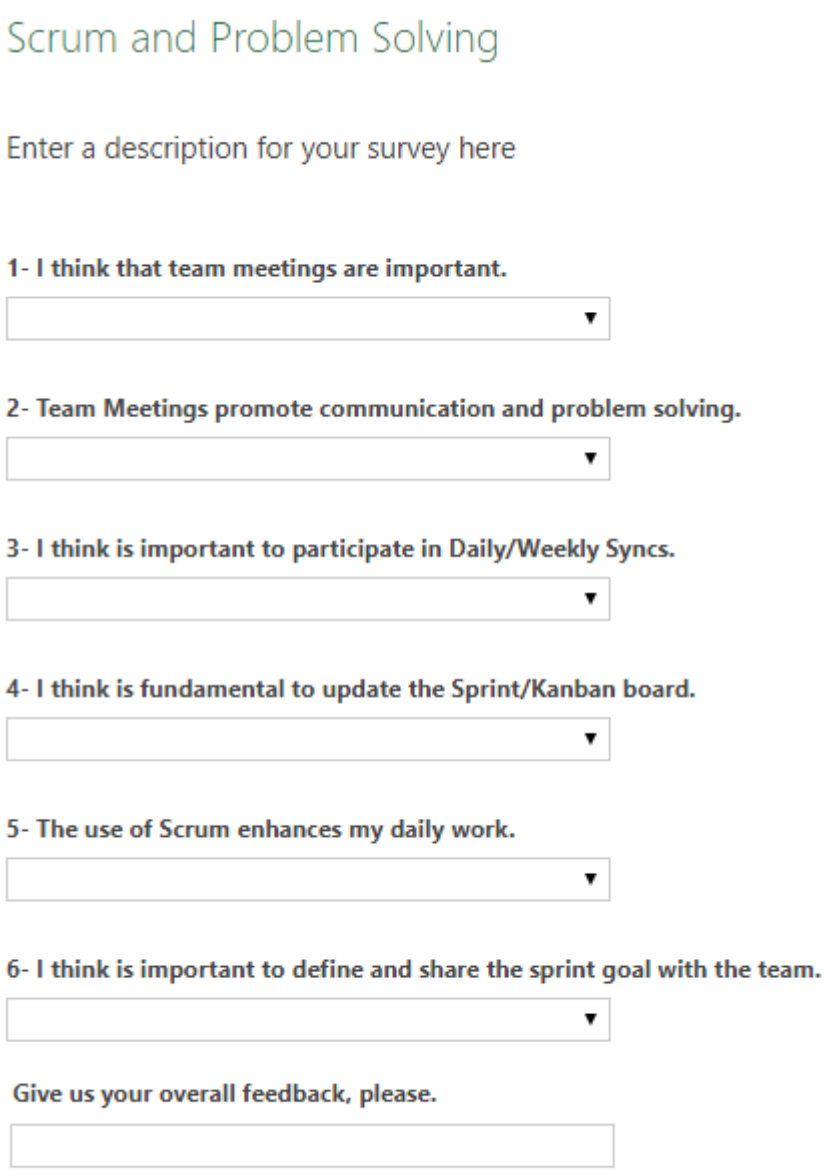

Figure 64 - Questionnaire I

Member 1 – "*Overall it was a good experience and a nice strategy for see and present the work we were being done during the week to the responsible (product owner). Something to use in future!! Well done!!"*

Member from A&V – "Better practices for an improved organization and Planning of the team";

Member 2 – "Demos, shelf labelling, focus on board and update task and Vicon room";

Member 3 – “Organizations of the closets to improve the time management when looking for material; Flying tasks have been reduced. In the beginning, these tasks disturbed a little the development of the main user stories.”

Member 4 – “Guide the team to the problems we can actually solve; Several were implemented - the more meaningful one was getting everybody to show up to meetings on time (when they were already in the office).”

Member 5 – *“The Scrum Team have been managing the Development Team to stay focus on discussing topics. Some problems leads to correction measures.”*

APENDIX II – QUESTIONNAIRE 2

Please give us your feedback about the Retrospective meeting, so we can become better together!

1 - How do you feel about attending the retrospectives?

Rate questions on scale from 1 to 5, assumig 1 as being very unssatisfied and 5 as being very satisfied.

2 - Did the scrum master explain to you the purpose of this meeting and present you the rules, steps and the objectives?

3- Are the issues discussed being relevant and meaningful to the team?

4- Are retrospectives helpful to you?

5- How effective was the Scrum Master in stimulating the team participation during this meeting?

6- Are the meetings generating prioritized and actionable items for improvement?

7 - Are you actively engaging this meeting?

No

Figure 65 - Questionnaire II (Part I)

8 - Were you honest and open in the meeting?

No

9 - Do you feel that your teammates were honest and open in the meeting?

No

IMPROVEMENT SOLUTIONS: Were the Improvement Solutions...

1 - Meaningful?

No

2 - Specific?

No

3 - Quickly actionable?

No

4 - What actionable improvement solutions have been implemented?

Figure 66 - Questionnaire II (Part II)

5 - What actionable improvement solutions have not been implemented?

Do you have any suggestion for improve this meeting?

Figure 67 - Questionnaire II (Part II)

APENDIX III – COMPLEXITY X VALUE MATRIX TEMPLATE

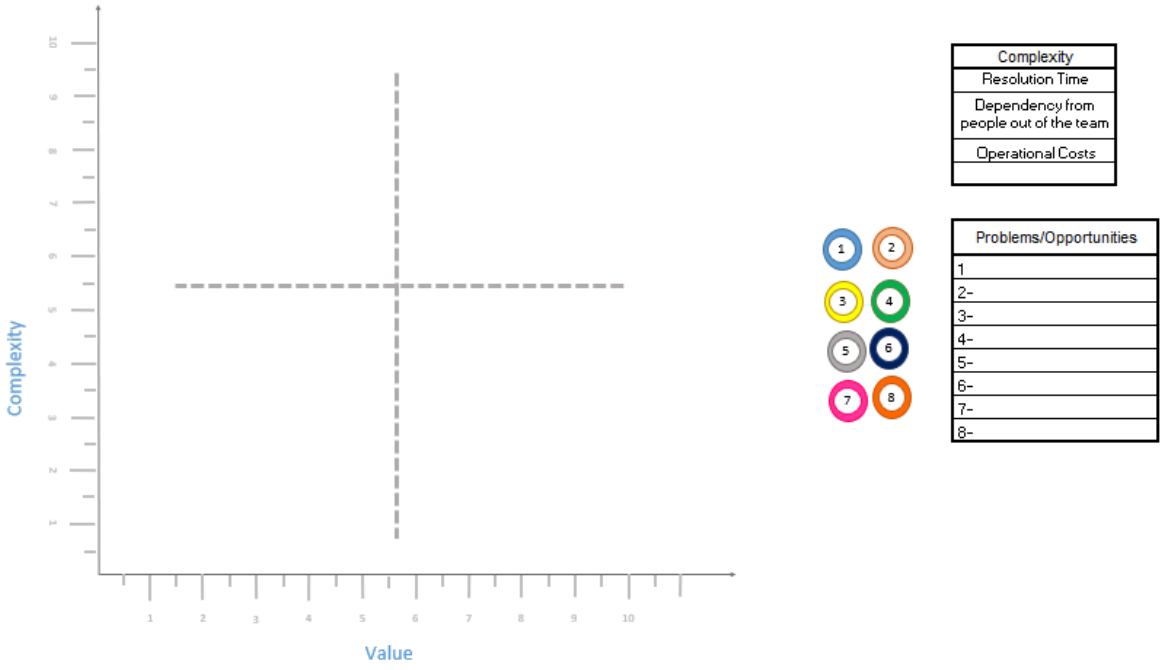

Figure 68 - Value x Complexity Matrix Template

APENDIX IV – FISHBONE DIAGRAM TEMPLATE

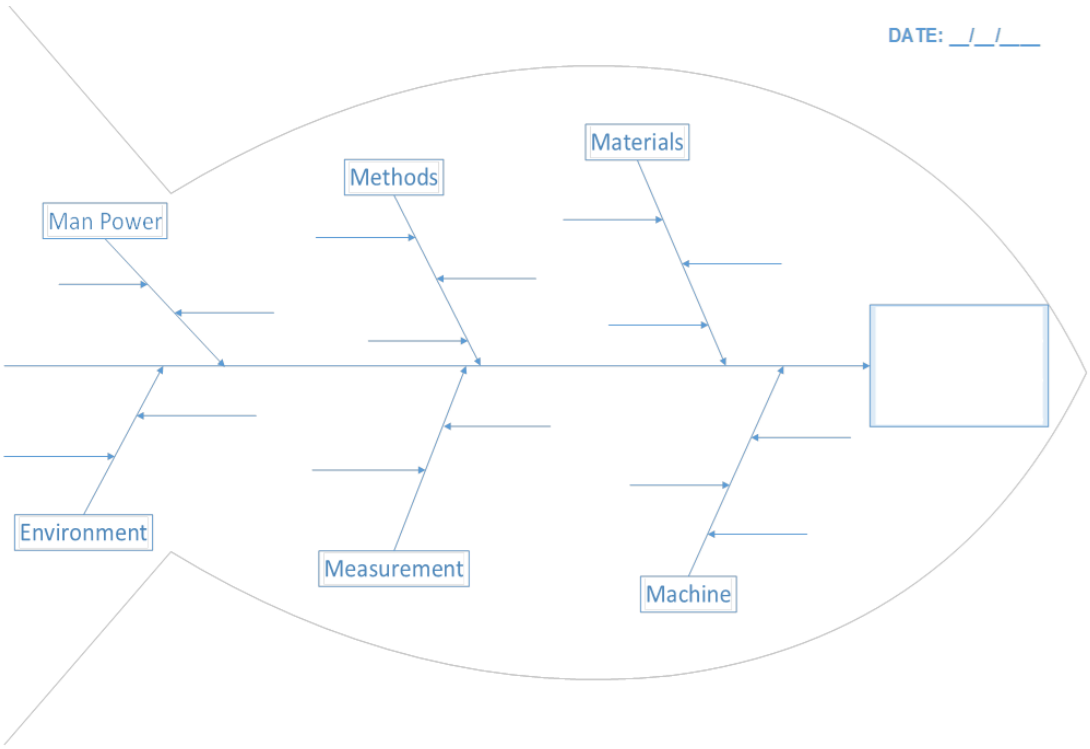

Figure 69 - Fishbone Diagram Template

APENDIX V – 5W2H TEMPLATE

5W2H

What?	Why?	Where?	When?	Who?	How?	How much?/ How many?

Figure 70 - 5W2H Template