MESG

MESTRADO EM ENGENHARIA DE SERVIÇOS E GESTÃO

Improving Agile Project Management Processes in a Global Software Development Company

Ricardo Alexandre Pessoa Nunes

Master Thesis

Supervisor at FEUP: Professor Doutor José António Faria

Supervisor at Company: Engenheiro Miguel Ferreira

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

2023-06-30

Let Everything Happen to You.
Beauty and Terror.
Just Keep Going.
No Feeling is Final.

Ranier Maria Rilke

Abstract

Agile project management has emerged as a popular and effective approach in response to the dynamic and ever-changing nature of software development. Despite its widespread adoption, there are gaps and challenges present in agile project management practices, with the adding impediment of global software companies facing additional challenges brought by virtual work.

This research project had as an objective the improvement of the agile project management processes of a global software company, market leader in its field. Adopting a case study approach and a service design methodology, the project investigated and aimed to solve the research objective, identifying major pains and barriers present in agile methodologies. It was important to understand the existing gaps in scientific knowledge by conducting a literature review. A characterization of the company, and interviews conducted with relevant and experienced stakeholders, validated the barriers and causes identified. The first interviews identified that problems mostly occur in three phases (sprint planning/data management, ticket management, and the setting of distributed teams), supported afterwards by the study of prioritization of problem fixing.

A proposal of possible solutions was then presented and displayed to interviewees, getting an assessment of the perceived impact and effort required for the solutions, and identifying metrics of success. A metric of prioritization for the implementation of solutions, called Implementation Score (IS), was proposed, relating the different factors gathered in the interviews. Finally, a decision support system for agile project managers was defined, utilizing several important factors as the variables, providing reliability, traceability, and trust to decisions of project managers.

The company has internally implemented more than half of the solutions, as well as having installed the IS and the methodology of problems prioritization, utilizing the established metrics to assess the success of those measures. The decision support system is also being put in work and evaluated in several teams.

This research project concretely identified main areas of agile management problems, while also providing a metric to analyze and prioritize the solutions to be implemented when faced with constraints. It also created an artifact that can aid project managers in decision-making regarding different teams' typologies, mitigating doubts, insecurities, and lack of trust by clients in agile project management. The fact that this decision diagram can be refined and worked on subjectively also goes accordingly to the agile methodology school of thought.

Keywords: Agile (Methodology); Project Management; Distributed Teams;

Resumo

Gestão de projeto Agile tem emergido como uma abordagem eficaz para responder à natureza dinâmica e de constante mudança do desenvolvimento de software. Apesar da sua adoção generalizada, ainda existem lacunas e desafios presentes nas práticas Agile de gestão de projeto, com o impedimento adicional de empresas de desenvolvimento de software globais encontrarem desafios trazidos do trabalho remoto.

Este projeto de pesquisa tinha como objetivo melhorar os processos de gestão agile de uma empresa de software global, líder de mercado na sua área. Adotando uma abordagem de caso de estudo e uma metodologia de designs de serviço, o projeto investigou e tentou resolver o objetivo da pesquisa, identificando dores severas e barreiras presentes nas metodologias agile. Foi importante entender as lacunas existentes no conhecimento científico ao fazer uma revisão de literatura. Uma caracterização da empresa, e entrevistas conduzidas com partes interessadas relevantes e com experiência, validaram as barreiras e causas identificadas. As primeiras entrevistas identificaram que os problemas ocorrem maioritariamente em três fases (planeamento de sprint/gestão de conteúdo, gestão de tickets e o formato de trabalho distribuído), apoiados depois pela priorização de resolução de problemas.

Uma proposta de possíveis soluções foi então apresentada e demonstrada aos entrevistados, obtendo uma análise do impacto e esforço perspetivado como requerido para as soluções, e identificando métricas de sucesso. Uma métrica de priorização para a implementação de soluções, chamada o Valor de Implementação (IS), foi proposto, relacionando os diferentes fatores obtidos pelas entrevistas. Finalmente, um sistema de apoio à decisão para gestores de projeto agile foi definido, utilizando diferentes fatores cruciais como variáveis, providenciado fiabilidade, rastreabilidade, e mais confiança nas decisões dos gestores de projeto.

Este projeto de pesquisa identificou concretamente as principais áreas de gestão Agile que originam problemas, assim como providenciou uma métrica para analisar e priorizar soluções a ser implementadas quando encontradas limitações, sejam temporais, económicas ou técnicas. Foi também criado um artefacto que poderá ajudar os gestores de projeto nas suas decisões conforme as tipologias das diferentes equipas em que se inserem, que mitiga as dúvidas, inseguranças e falta de confiança por parte dos clientes em gestão de projeto Agile. O facto deste diagrama de decisão poder ser melhorado e trabalhado subjetivamente também vai de acordo com a escola de pensamento das metodologias Agile.

Palavras-Chave: (Metodologias) Agile; Gestão de Projeto; Equipas Distribuídas;

Acknowledgements

This project was achieved through the contribution of several people and, as such, I would like to acknowledge their attention, availability and dedication to me and the research project.

First of all, to msg insur:it Iberia in general. During the time that I had to do my research, everyone had their door open to me, providing me with valuable insights, helps, tips, recommendations, or, overall, a friendly chat. Although I value the entire company, I would like to highlight my mentor in the company, Miguel Ferreira. He made sure I felt at home and welcome, integrated, and with every possibility and information I should need. In periods of uncertainty, the stability was important. Thank you. I would also like to especially name and thank Márcio Marques, Luís Marques, Jorge Miranda, Paulo Sousa, Luís Braga, Sara Oliveira, Ana Saraiva, Miguel Azevedo, João Carvalho, Miguel Castro, Mariana Mesquita.

I would like to thank my dissertation advisor, Professor José Faria. Starting a research project of this dimension is always tough and he was an immense help in showing me the way that needed to be paved.

I would also like to thank Professor Jorge Teixeira. His expertise, focus, and dedication to me and the entire student body is something that should be commended. Day or night, he was always available for a helping hand.

To my girlfriend, Luísa. Your confidence, pride, and comfort in me made this process much better through these months. My partner in crime and the light in the night. Thank you.

To my father, Carlos Nunes. He provided me guidance and certainty in times of need, and always let me explore. The path to knowledge begins since a baby, and the path that you put me on brought me to where I am today. Thank you.

Last, but not least, and since I am not allowed to write an entire dissertation (and I could) on everyone that supported me and helped, I want to send my greetings and thanks to my friends, to those that believe in me, that support me, that know what I am capable of sometimes better than myself. Although I do not usually know how to express it, my gratitude is ever-present.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

APMDM - Assistance to Project Managers Decision-Making

CCC - Core-Country-Customer

DSR - Design Science Research

ERP - Enterprise Resource Planning

HR - Human Resources

IID - Iterative and incremental development

IS - Implementation Score

IT - Information Technology

KPI - Key Performance Indicator

OKR - Objectives and Key Results

PT - Portugal Team

QA - Quality Assurance

Q&A - Questions and Answers

RQ - Research Question

SeTaK - Selenium Test Automation Kit

UW- Underwriting

UWB - Underwriting Workbench

XP - Extreme Programming

1. Introduction

1.1. Project Context

One of the methodologies surfacing and becoming more frequent in software development is the Agile methodology (Alsaqqa et al., 2020). Agile methodology is "a conceptual framework for software engineering that begins with a starting planning phase, following the road toward the deployment phase with iterative and incremental interactions throughout the life cycle of the project" (Alsaqqa et al., 2020). Agile methods intend to reduce dispersed or exaggerated efforts "in the software development process with the ability to adopt the changes without risking the process or without excessive rework" (Alsaqqa et al., 2020). As such, agile methodologies are more focused on improving the simplicity of processes, while devaluing extensive documentation gathering and tracking. Agile is, then, more commonly associated with projects with fast changing requirements and turbulent environments (B. Boehm, 2002).

Project management decisions in an agile setting are different than those from traditional settings, as one is dealing with a volatile environment, flexible requirements, and a developmental and iterative process (Augustine et al., 2005).

This project intends to understand how the methodology and principles of Agile development function in a multicultural and geographically dispersed team, all the while also understanding how these methodologies and principles function in different types of software projects: Software development, implementation of first party software, and/or implementation of third-party software. A further study on project management and project managers, especially in an agile setting, is then needed.

Msg insur:it Iberia, previously msg life Iberia, a branch from msg insur:it, the company where the project was introduced, has experienced a rise in workers and involvement in projects. They are being compelled to create new teams and to bring new members at a faster pace, trying to maintain or improve the company's productivity and efficiency regarding the quickly moving market's expectations and standards.

It is important to maintain quality of delivered software, even in the face of tighter timeframes and strict demands. The changes that the company is facing are, then, more forced by the nature of the market, more so than the gradual or organic cultural development. This was seen as an opportunity by the company to revamp the practices and tools being utilized in succeeding with their objectives, while maintaining the Agile methodology.

This project has the initial research objective of understanding the current state of the Agile methodologies in the software development and project management processes. The following research objective consists in using the knowledge in processes and project management in tandem with the knowledge acquired of the company, identifying the areas of bigger concern, and providing solutions to the identified problems, evaluating them on severity and frequency, while also creating a framework of decision-aid for project managers in this company and companies of a similar nature (software development). Besides the strategy of implementation, there is also a necessity to define metrics or results that permit the claim of success or failure of the measures implemented.

1.2. Problem Description/Motivation

Software quality is notoriously difficult to pin down. Every software development activity, from drawing up the requirements all the way through to deployment and maintenance, can at some level contribute to (or detract from) the quality of the final software system (Walkinshaw, 2017). Software quality is recognized differently depending on each perspective, including that of the clients, maintainers, and user (Laporte & April, 2017).

Recent years have seen the geographic distribution of software development. This distribution allows team members to be in various remote sites during the software lifecycle, thus making up a network of distant sub-teams (Jiménez et al., 2009). Although this distribution allows for a communication between greater distances (there are distributed teams dispersed throughout different continents), it can also lead to problems directly correlated to those greater distances (Morrison-Smith & Ruiz, 2020). Discussing the optimization of project quality in a distributed environment is a two-fold problem: How and what tools are used to optimize a project and how are distributed (e.g., in geography, demographically, technically) teams projects managed (Walkinshaw, 2017).

Not only are companies facing the benefits and drawbacks of distributed teams, but they are also incorporating agile methodologies and project management in their functioning. There are several Agile methodologies (Scrum, Kanban, Extreme programming (XP), among others) that preach simple concepts, but, dealing with people, can lead to complex events (Augustine et al., 2005). Sometimes, in fast-paced settings and teams of multiple typologies, there is a generation of a "lack of shared understanding of the project's goals", which might lead to the diminishing of the quality of the product delivered (Augustine et al., 2005).

In an agile setting, the strong definition of roles brings the advantage of combining individual strong aspects of workers into a solidified package in the form of the product (Cervone, 2011). Nonetheless, this dependability that each role has on the workers might bring some problems, especially if fast processes lead to lack of documentation or lack of oversight from the part of the project manager (Cervone, 2011).

Agile project management is also harder to track down and measure, as one must deal with constantly changing requirements and environments of work, in opposition to the traditional development strategies (Fernandez and Fernandez, 2008). If a project manager can eliminate the bureaucratic struggle of constant and strict documentation that are associated with the development models used in the last century and closely manages the process and the team, the productivity of the project, and, subsequently, of the company, should increase, leading to a higher probability of success (Cervone, 2011).

Since Agile methodologies are still in their early stage (a few decades of development), the studies and understanding of project management in an agile setting, especially when introducing the remote setting, can still be rudimentary (Augustine et al., 2005). There are some simplistic frameworks defined for project management, but they are still under-developed (Fernandez & Fernandez, 2008).

Software quality assurance is a multi-phased and multilayered process, and the combination of projects of such nature with a distributed team on different levels might incite some challenges and roadblocks that need to be dealt with.

1.3. Research Questions

The aim of the research questions is to guide the project through a logical framework, with the intending to reach the end of the project with all of them answered. They are structured in a way that relates to the project and its objectives. Throughout the project, it was noted that, although some problems were identified in the general field of processes and project management, rarely were the causes properly addressed or solutions presented. The research in this dissertation is meant to understand the issues involved in project management, their causes, and what can be done to fix them. As we move forwards with that knowledge, it would also be important to delineate a framework for project managers to follow when they need guidance or specific measures.

As such, at the end of this dissertation, these research questions (RQs) should be answered:

RQ1 - What are the problems affecting software development global companies in their processes?

RQ2 - What are the causes of the identified problems?

RQ3 - What are the solutions that could be implemented to avoid or mitigate those problems/causes?

RQ4 - What are the strategies of implementation of the identified solutions?

RQ5 - How can a Project Manager decide on team and project management implementations and strategies?

1.4. Study and Project Development at msg insur:it

Msg insur:it Iberia, the Iberian branch of the msg insur:it company, a software development company, mostly focuses in providing services to healthcare and insurance companies all over the world, spanning its projects through many continents, focusing mostly on Europe and the United States. The company prides itself in being a market leader of information technology (IT) insurance solutions.

Providing software and consulting solutions for insurers in an ever-changing world poses its challenges and requires flexibility, which is why the company employs Agile methodologies throughout their projects. In the last two years, due to a large increase in projects and opportunities, this branch has doubled in size, going from 40-50 employees to over 80. A turbulent market coupled with a big inflow of new workers needs to be dealt with carefulness and precision.

The project conducted is meant to address several factors that the company wanted studied and delved deep, with the research objectives of the company being:

1. Exploring the different Agile tools.
2. Exploring the effect of new workers in a distributed setting and how a big influx of workers affects company culture.
3. Identifying improvement opportunities in company processes.
4. Studying how to do maintenance of quality in a distributed setting.
5. Structured project management decision making.

To accomplish these objectives, the company provided the opportunity of an internal study, allowing for the careful and close observation of processes, relationships between teams and

the overall structure and functioning of the company. The process to identify and solve problems was then defined:

1. Characterization of the relevant processes and company structure.
2. Identification and prioritization of problems.
3. Conceptualizing, prioritizing, and proposing possible solutions.
4. Provide strategies of implementation and measure the output.
5. Implement the solutions into an integrated framework of project management.

At the end of the report, the factors should have been well-defined and studied, giving rationality, credence, and a bigger probability of validity, compatibility, and reliability to the strategies and solutions presented.

1.5. Report Outline

In this section, the structure of the report will be outlined and explained. Throughout the project, the workflow shown by figure 1 will be followed:

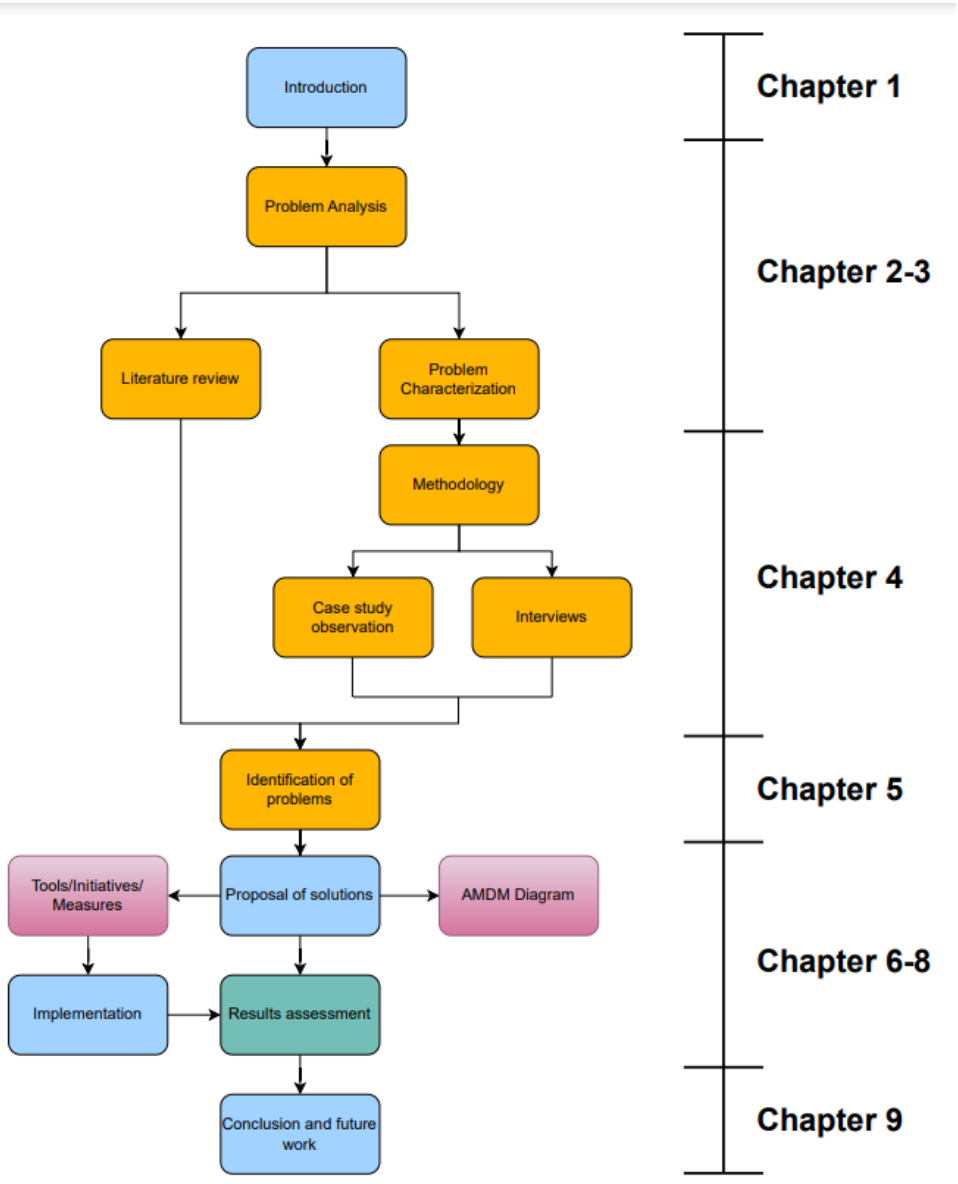


Figure 1 - Report outline with the corresponding chapters

This sub-chapter concludes the introduction portion. The following chapter (Literature review) is meant to study the state of the art of the relevant areas of the project. In it, a more in-depth understanding of the theme at hand will be provided, as well as showing some of the advantages and disadvantages of the topic of study chosen. This study and comprehension is vital, as it shows what others gathered previously. A dissertation is meant to bring something new towards the field of research, and, as such, providing the literature review grants reliability to the topic and ensures that there is no repetition of unnecessary information throughout the project.

In the "problem characterization" chapter, the company, their understanding of their problems and objectives will be defined. Not only that, but also the gaps identified in the literature review and the intentions of innovation that the dissertation should follow.

The methodology chapter intends to provide a comprehensive explanation and comparison of different methods that could have been chosen, and what was chosen. For projects of different typologies, different methods and procedures are necessary, so, it is important to understand why this one chose the specific ones it did.

The literature review and the methodology will permit us to identify problems that will be listed in the following chapter. After describing and understanding the problems, solutions can be proposed.

The "implementation and measurement of solutions" chapter tries to provide a sense of accomplishment in the solutions proposed, in the form of results. In it, the study of the implementation of the thought solutions and the metrics required to define the success of such implementation will be laid out.

The second to last chapter will introduce the main artifact of this dissertation, a decision support system that will permit for project managers to have a dedicated framework that helps in agile project management decision-making.

In the final chapter, the conclusions that we can infer from this work will be stated and there will be suggestions of future work to be done. It should be clear if the research objectives were completed, what was the innovation brought by the dissertation, and what is the contrast between what existed in the previous literature and what was now brought forward by the student.

2. Literature Review

This chapter will approach the theoretical background scientific content consulted and studied to develop this project. General notions needed to be specified and understood for a deeper and proper exploration. In it, three different, but key aspects, of the project will be studied and defined separately, with a final sub-chapter studying the existing literature combining two or more of them.

In this research, the sources utilized were mainly books, articles, and other available dissertations. Academically, it is very important to ensure that every information consulted and cited is trustworthy, with this project focusing mostly on works that are peer-reviewed and cited often. As we are dealing with fast-paced and everchanging topics, it was also important to note the contemporaneity of the sources cited.

2.1. Software Development

One of the first models used to visualize and approach software development was called the Waterfall Model (Royce, 1970). According to the model, a linear path is followed, going from as early as establishing the system requirements, building through the software requirements and analysis to coding and testing, following towards the final stage of an operating system. The process treats every phase separately and does not allow for a return to a previous stage. As such, implementation of the waterfall model implies a tremendous emphasis on documentation (Walkinshaw, 2017). Every process must be carefully studied and performed, as a failure of a step implies the failure of the process.

Royce follows it up with an alternate, iterative waterfall system. This model would focus less on a linear path and more on an iterative path, following a return from a successful step to the previous and next one once it was finished, with an intermediate testing step between advances. This intended for the changes to be scoped down to manageable limits (Royce, 1970). Although the waterfall model is considered a traditional approach and is still used frequently, many software engineers are opting for other methodologies in recent times (Benediktsson et al., 2006).

While Royce suggested the waterfall model, other models of software development were being constructed, such as the iterative and incremental development (IID) (Larman & Basili, 2003). IDD first appeared from the work of Walter Shewhart, suffering improvements along the twentieth century, having hundreds of books and papers promoting it as their main or secondary theme (Larman & Basili, 2003).

The advances in IID were nonetheless aided by Royce's creation, as the refinement of the waterfall model led to models that are still in use today, like Barry Boehm's Spiral model (B. W. Boehm, 1988). B. W. Boehm (1988) stated that the spiral model could "accommodate other models as special cases", and that his model would be divided in four phases. Determining objectives, alternatives, and constraints, followed by evaluating alternatives, identifying, and resolving risks, followed by the development and verification of the next-level product, culminating in the planning of the next phase (B. W. Boehm, 1988). The cycle would then progress until the software was deemed acceptable and was then tested and implemented. At the time, Boehm utilized his model, to develop a Software Productivity System, stating that "all the projects using this system increased their productivity by at least 50 percent" (B. W. Boehm, 1988).

Besides incremental methods, other methodologies surfacing and being developed are evolutionary methods and overall "agile" methodologies (Benediktsson et al., 2006).

2.2. Agile Methodology (Agile software development)

Software has progressed largely throughout the 21st century, with large organizations adapting even higher quality standards and the business market, combined with new technologies and shorter deadlines requiring a diligent and agile software production process (Almeida, 2017). The agile mindset is more flexible than the waterfall model, breaking down the software development process in smaller chunks, permitting changes in requirements (Almeida, 2017).

The Agile methodology in software development is different from traditional approaches as the "continuous tests and iterations occur during the entire software development lifecycle", in a dynamic fashion (Srivastava et al., 2017). B. Boehm (2002) faces the debate between the defenders of the traditional waterfall model and the progressives of the agile movement with moderation, defending that a "combined approach is feasible and preferable", and that companies should "evolve towards the best balance of agile and plan-driven methods that defend their situation". An agile perspective tries to focus more on product quality and on its consistency of processes than its repeatability (Lycett et al., 2003).

Agile methodologies value and prioritize testing (Nerur et al., 2005). This so-called test-driven development makes the code more understandable and "facilitates the introduction of new coding or the changes of the existing one" (Nerur et al., 2005).

In 2001, 17 supporters of the agile methodologies gathered and discussed among themselves, originating between them what they called the "Manifesto for Agile Software Development" (Fowler & Highsmith, 2001), a document that was meant to value:

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

Furthermore, the Agile Manifesto provided a set of 12 principles that meant to encapsulate the main ideas of the Methodology. (Fowler & Highsmith, 2001). All in all, the agile way is useful in the sense that it deals with uncertainty and tackles emerging or possible crisis head-on, with little down time between stages of development (Fowler & Highsmith, 2001).

There are several Agile techniques, such as "Extreme Programming (XP), Scrum, Crystal Methods, Feature Driven Development, Lean Development, Dynamic Systems Development Methodologies, and others" (Cohen et al., 2003). They differ, but their similarities come from the agile framework characteristics: Iterative development and focus on interaction, communication, and the reduction of intermediate stages (Cohen et al., 2003). Out of those, SCRUM has gained popularity (Srivastava et al., 2017), influencing work experiences and having the key agile characteristics (Cho, 2010).

2.2.1. SCRUM

The first mention of "SCRUM" in literature is an article by Takeuchi and Nonaka (1986) in the *Harvard Business Review*. In it, the authors present a new process alternatively to the traditional and linear ones, comparing the new process to a team of rugby passing the ball back and forth:

a multidisciplinary team with members working in tandem towards a specific goal. Takeuchi and Nonaka (1986) state that this process is "essential for companies seeking to develop new products quickly and flexibly". Scrum comes, then, appropriately, from a rugby strategy, and it can also mean "crowd".

It is the most popular agile methodology, being an "iterative, incremental, and empirical process", presenting defined roles such as scrum master, product owner, and the team itself (Lopez-Martinez et al., 2016). The scrum approach accepts that a problem cannot be "fully understood or defined", and that it is best to have a flexible and reactive approach to the new requirements than having the traditional rigid approach (Lopez-Martinez et al., 2016).

The Scrum process is divided in three stages/phases (Abrahamsson et al., 2017). The pre-game phase includes the planning of the system, where a product backlog list with all the requirements is stored (Abrahamsson et al., 2017). The product backlog list receives regular updates and at every iteration is reviewed, and in the architecture phase is the design of the system based on the product backlog (Abrahamsson et al., 2017). If it demands a change of an existing system, the relevant backlog items are identified and reviewed for potential future problems (Abrahamsson et al., 2017).

The development phase is where the agility comes into play, where "the unpredictable happens" (Abrahamsson et al., 2017). There are several techniques and tools used in the sprint section, that lasts 2 to 4 weeks and is planned by the product owner and the team (Sprint planning), although that might change during iterations (Lopez-Martinez et al., 2016). The tools used might be:

- Daily scrum: Inspection of the progress towards the sprint goal and adaption of sprint backlog if necessary. Usually a 15-minute meeting at the same time every working day (Schwaber & Sutherland, 2011).
- Sprint review: Inspection of the outcome of the sprint and adapting for the future sprint, with the possibility of adjusting the sprint backlog if necessary (Schwaber & Sutherland, 2011).
- Sprint Retrospective: Planning of ways to increase quality and effectiveness, by inspection of the elements involved in the sprint (individuals, interactions, processes, tools), being the final step of the Sprint (Schwaber & Sutherland, 2011).

The post-game phase is the final part of the project and happens when the all the necessary requirements are agreed to be satisfied, no more issues can be found, and no more items can be added (Schwaber & Sutherland, 2011). The system will be released, leaving only the integration, the system testing and documentation to be done (Abrahamsson et al., 2017).

Alas, Scrum is not a universal tool and is able to be used only in the right context, with Abrahamsson et al. (2017) suggesting that Scrum should only be used by teams of less than 10 engineers. Scrum will also only work if the organization accommodates to it, as resistance to chance will hinder the adoption of agile methodologies and traditionalism might impede progress (Lopez-Martinez et al., 2016).

2.3. Agile Project Management

2.3.1. Project Manager

The project manager role is extremely important, as "the financial stability of a project (and a company) is entirely dependent on the management of said project", as there are certain

resources allocated to achieve a goal (Schmid & Adams, 2008). In general, "performance is a critical concern", and a project manager should strive to understand critical success factors for the project they are inserted in, much like customer relationships, customer care, cost, and risk assessment (Gunduz & Almuajebh, 2020).

As the main source of power or drive for a project, project managers should focus on how to motivate their workers (Schmid & Adams, 2008). Gagné & Deci (2005) state that the ideal motivation journey of a worker comes first from a "state of amotivation" (lack of intention and motivation to do a task whatsoever), leading to a "state of extrinsic motivation" (that requires some consequence and stimuli outside of the activity project managers intend to motivate the worker to do), concluding in a "state of intrinsic motivation", where the employee has "an interest and enjoyment of the task and is inherently autonomously motivated to do so".

Other problems identified in empirical studies about project management are financial problems, administrative aspects, and authorities' approval mechanisms (Gunduz & Almuajebh, 2020).

2.3.2. Traditional vs Agile methods of project management

As time passes and volatile environments are ever more present from a business perspective, so are the organizational environments in those businesses (Augustine et al., 2005). As discussed in chapter 1.2, traditional formal software development methodologies may be seen as too strict, and too linear of processes, and the existing management approaches might only be effective with stable and known consistent requirements, while turbulent environments with flexible requirements need a more dynamic project management approach (Augustine et al., 2005).

Traditional project management is characterized by planning in advance, focusing on stability and a long-term perspective, while an agile project management is "incremental, iterative, flexible, and short-term oriented" (Thesing et al., 2021). The advantages of both approaches to project management are more accentuated in areas where the methodologies exhibit significant differences, with traditional planning management wanting "clear roles, responsibilities, and systematic, stable documented planning", while the focus of agile project management is the "ability to recognize the need for changing requirements in a short period due to regular feedback from the customer" (Thesing et al., 2021). Given that, traditional project managers are concerned with "budget, schedule, and scope, wanting to reduce risk and loss of time and money", while the agile project manager is more preoccupied with "deliverables and increasing business value", being trained to deliver a product, not having a fixed process (Fernandez & Fernandez, 2008).

Thesing et al. (2021) states that there are certain criteria that can help decide between procedures models, with projects not being suitable for agile methods if the following apply:

- Inability of decomposing a project into smaller, separate deliverables.
- Changes are not supported by nature of legal, technical, or budgeting points.
- Operations in real time or safety critical systems should not have a turbulent environment and might not be suited for agile project management.
- The nature of the organization or of the project manager do not suit the conditions: if management does not understand agile philosophy, project management will be poor and lacking.

If one or more of these occur, the project will most likely not be suited for agile methods and should be substituted by more traditional models, e.g., the classical waterfall approach (Thesing et al., 2021).

A hybrid approach might be the most valid approach in some cases (Fernandez & Fernandez, 2008), with Cunha and Gomes (2003) stating that "traditional engineering roots of management processes should be complemented with a more organic and adaptive view".

2.3.3. Risks in software project management

Agile project management has a solid concept behind it; however, it can still cause an impact on the factors associated with it (Coram & Bohner, 2005):

- People: multiple people, from developers, testers, customers, and other stakeholders that are expecting all kinds of results. The customers look for a product while executives look for profit (Coram & Bohner, 2005).
- Process: Agile project management affects planning, documentation, and development processes, with an informal planning, sparse documentation, and constant changes in development process (Coram & Bohner, 2005).
- Project: Agile methods affect project types, business factors and other project characteristics. As mentioned, not every project type suits agile methods (Thesing, 2021), and some projects might have strict deadlines and expectations, concerning time span, features, and/or capabilities (Coram & Bohner, 2005).

Some challenges presented by this methodology can be eased by adding back some formality (Coram & Bohner, 2005), giving some more credence to the hybrid approach discussed by Fernandez and Fernandez (2008).

An agile project management might be described as "lazy" when it comes to eliciting requirements, as it is only done superficially, vaguely, and delays efforts/expenses to finishing stages, which might also induce cost savings (Paetsch et al., 2003). This mentality can be risky, though, as it is highly dependable on the high level of expertise of developers, trusting that they understand the agile techniques and that they will develop and implement them well, even with a low level of structure and information (Paetsch et al., 2003).

Some or even "most software projects", even with improvements in methodologies, "still use more resources than planned", are timelier than expected, and are less functional, providing "less quality than expected" or intended (Barros et al., 2004).

2.4. Distributed/Virtual Teams

Distributed teamwork has become a common theme in organizations and in business in general, being a solution to faster time-to-market needs, and a less costly solution to complex organizational problems (Connaughton & Shuffler, 2007).

Virtual teams can be defined as a set of members connected in a work arrangement with spatial distance, limited physical interaction and working mostly between each other using predominantly electronic media to communicate between them (Dulebohn & Hoch, 2017). A virtual team is functional, disperse, and function on a spectrum of virtuality, having members in direct contact and others that never once speak face-to-face, managing their communications

only electronically (Liao, 2017). Multicultural distributed teams have been popularized in numerous organizations (Connaughton & Shuffler, 2007).

2.4.1. Leadership in Virtual Teams

Virtual teams create several leadership challenges (Bell & Kozlowski, 2002). The lack of face-to-face contact and the spatial (and probably time) dispersion, combined with the technological nature of communication, makes it more difficult for leaders to perform their natural tasks (Hoch & Kozlowski, 2014). Team leaders must be competent and able to manage virtual teams, providing training and guidance, as well as being able to monitor the professional and emotional needs while at a distance (Ford et al., 2017).

Although leadership is important, it is also of paramount importance for the leader to be able to empower his team, as more power and independence in project decisions might reduce the likelihood that the spatial distance will affect the project results (Chinowsky & Rojas, 2003).

"Electronical" leadership might bring some challenges, however, if leaders analyze and understand them, they can turn them into opportunities by adapting their behavior to the virtual settings, learning new skills, and applying them (Lilian, 2014). The best leaders ensure their teams have trustworthy relationships, communicate well, and solve conflicts properly (Ford et al., 2017).

2.4.2. Trust

Ford et al. (2017) define trust as "the willingness of one to be vulnerable to another based on the expectation by a trusting party that party being trusted will perform a particular action important to the trusting party, regardless of the ability to monitor or control the other party". Not only the leader, but all parties involved must foster techniques and ways to improve trust (Ford et al., 2017). Pavlou et al. (2007) state that "trust is the belief that the other party will fulfill the expectations and promise of competence, integrity, and benevolence."

Hsu et al. (2007) believe that teams that are well aligned in terms of interests, goals, and objectives, will share their knowledge more easily and be more trustworthy. In some contexts, though, people might only want to share knowledge if there is an expectation of reward or something beneficial to them (Hsu et al., 2007).

Cultural, linguistic, work style, and time zone differences might be challenges that come along with the virtual team management, which might increase the cost of hiring and training a proper project manager that is equipped to deal with such (Ford et al., 2017). Separation of individuals makes it harder for them to form bonds and create relationships (Chinowsky & Rojas, 2003). Employees of different cultures coming together under the same organization can offer some reluctance in knowledge sharing (Ardichvili et al., 2006).

Employees should be encouraged, increasing their levels of motivation (Gagné & Deci, 2005), ensuring independence and determination when faced with difficult tasks (Ford et al., 2017). Teambuilding has been shown to increase trust to an extent (Crisp & Jarvenpaa, 2013). Virtual teams should be conducted in an amiable atmosphere, allowing members to better know each other and have an easy communication, thus leading to more trust between them (Hsu et al., 2007). One company, for example, introduced a "virtual break room as a place for casual,

informal interaction for team members", to combat the lack of physical interaction (Ford et al., 2017).

Virtual teams that do multiple projects increase their level of trust in each other, as they perform their tasks, interact frequently, and accomplish results (Saunders & Ahuja, 2006). Hiring a manager with previous experience in virtual teams might reduce the presence of problems and increase success (Ford et al., 2017).

2.4.3. Communication

Traditionally, teams had no need for complex linking technologies, or they only used it sparsely, as most interactions were done face-to-face. Since virtual teams are divided by distance, communication technologies are crucial (Bell & Kozlowski, 2002). Companies should be mindful when choosing the technology used for communication purposes, ensuring that they are as reliable, rich, and fast as adequate for the necessities of teams. (Ford et al., 2017). Possible technologies used might include telephones, e-mail, videoconferencing systems, private networks, and other systems either of an intranet or extranet nature (Saunders & Ahuja, 2006).

Higher frequency of communication is not necessarily a good sign, and companies should focus on the *quality* of communications (Connaughton & Shuffler, 2007; Marlow et al., 2017). A high level of quality communication leads to a better understanding, what can be called "shared cognition" (Marlow et al., 2017). A bigger focus on technology as the main mean of communication without proper coordination can generate confusion and/or information "overload" (too much information without any proper use is passed between members) (Lilian, 2014). Virtual teams remove time and space barriers from potential skilled candidates from another part of the world who would otherwise not be able to perform the job (Bell & Kozlowski, 2002). They offer high flexibility to workers, and faster answer times (Bell & Kozlowski, 2002). Such flexibility may allow for employees to have a better work-life balance, increasing indexes of happiness (Liao, 2017).

2.4.4. Conflict in virtual teams

Conflict is something that must be managed in virtual teams (Connaughton & Shuffler, 2007). Hinds and Mortensen (2005) state that there are two types of conflict: Interpersonal conflict, motivated by the relationship between members, and task conflict, where team members have differing views on expectations or process of achieving a task. Members tend to breed more conflict in virtual teams as they do not share the same context (environment, culture, technologies) (Hinds & Bailey, 2003). The study by Hinds and Bailey (2003) affirms that affective (interpersonal) conflict is more common in teams with friendlier and familiar connections. While interpersonal conflict is almost always seen as negative, task conflict might have some positive side effects, by helping the team to find alternative methods to a solution (Hinds & Bailey, 2003).

(Chinowsky & Rojas, 2003) defend that conflict can be avoided if measures are established early or if sensitive areas get more focus, concern, and deliberation before any decision. Studies indicate that in environments of conflict performance suffers (Saunders & Ahuja, 2006).

Some state that a team is better to be fully distributed (every member in different locations) than partially distributed, as subcultures and relationships between closer members may differ

from the rest of the team (Chinowsky & Rojas, 2003; Connaughton & Shuffler, 2007), creating unforeseen troubles to leadership and team management (Saunders & Ahuja, 2006).

Virtual teams include closer maintenance and possible restructuring, which increase complexity and number of work relationships, as well as increasing the possibility of information leaks (Ford et al., 2017) and a lesser sense of connection (Breu & Hemingway, 2004; Chinowsky & Rojas, 2003).

2.5. Agile Methodology in Distributed Teams

Management of software development in distributed teams has challenges that are not associated with local teams (da Silva et al., 2010). Although this is widely recognized, most distributed teams still use traditional method processes (da Silva et al., 2010).

In major companies with plenty of distributed sites, one can sense lack of synchrony, and volatile requirements brought by agile methodologies need a well-supported structure to face these challenges (Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001). Technological and communication issues, such as failures in systems or outdated software might cause some delays as well (Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001). Different types of tasks generate interdependencies between teams, requiring different types of coordination (Espinosa et al., 2007). These dependencies affect the entire organizational structure, whether teams are involved in the process or not, and should be carefully studied before solutions are implemented (Espinosa et al., 2007).

Agile methods are well-adapted for turbulent environments, which may be the case in some instances for distributed teams (Coram & Bohner, 2005). When rightly applied, the costs are lower, the quality, efficiency and productivity is better, and change is quickly addressed (Coram & Bohner, 2005). If necessary, to face some specific challenges pertaining to the Agile methodology, managers can employ some formality, characteristic of more traditional methods (Coram & Bohner, 2005).

An Agile methodology in software quality assurance with distributed teams offers an alternative path to software development and methodologies that support ill-defined requirements, and even some questionable projects, showing itself as a valuable tool in the search for effectiveness, efficiency, and better products and processes (Coram & Bohner, 2005).

The literature gaps found in the literature review were mostly focused on:

- Lack of depth on the problems specific processes encounter. There are problems stated and identified, but not exactly where in the process and the causes are not very well-specified.
- Lack of suggestions for solutions provided. There is not enough information in the current literature regarding suggested solutions, initiatives, and/or measures to tackle the problems or causes that were identified.
- Lack of an established framework for decision making by part of the project manager. There are statements and inferences regarding what a project manager must do, his responsibilities, and what his skillset must be, but there is no research on what factors affect his decisions and how to choose project management strategies.

3. Analysis and Diagnosis of the Company

This chapter will introduce the general problem and the objectives of this work of research. The company that is the basis of the case study will be introduced in more detail, explaining the different opportunities that they brought to the possibility of finding problems (and solutions) to this theme. This intends to accomplish the research objectives of exploring the different agile tools, how to identify problems and solutions, and also studying the influx of new workers and their onboarding. These identifications will permit a structured project management decision.

3.1. Definition of the Company

Msg-Insur:it Iberia is a branch of the German multinational msg insur:it. The msg group was founded in 1980, currently employing 8000 people, with offices in 23 countries. They intend to provide life insurers and overall partners in the Health, Property, and casualty insurance with software development of the utmost quality.

Msg insur:it follows a Core-Country-Customer (CCC) approach. A CCC structure combines a standardized software approach with some benefits of custom software development, allowing for a modern infrastructure combined with the custom features that insurers need to stand out from the competition.

Msg Insur:it employs flexible working hours and locations. As long as the employees attend the obligatory meetings (more on that on later sections of this work) and fulfill their working hours, a worker can perform his or her tasks at home, at the office, or in a hybrid setting, with a few exceptions. The company is composed of several different teams and projects (Appendix 1), all working along some form of agile methodologies. The size, scope, and dependencies of projects are different and the way each team works varies based on that, even if they share some similarities.

The company has been receiving many new projects and required the entrance of several dozens of new employees, practically doubling in size in the last 2 to 3 years. The entry of this much “new blood” carries out some possible problems associated that the company wanted to assess, much like the changes in culture established by the offset of the ratio between experienced members and juniors, as well as the difficulties associated with the onboarding process of workers that are mostly working remotely.

Besides that, the fact that a lot of new projects have appeared also brings some concerns regarding the methodologies used so far. The company intended to have an external and unbiased looks on the way the projects, teams, and the company is organized and run, to perform the necessary refinements to properly accommodate the new clients’ requirements.

For that, the company was very open and straightforward in providing extensive documentation on processes, roles, and organizational structure (more on that on the following sub-chapters). In turn, this report intends to provide, in the same vein, an extensive output detailing the observations conducted during the timeframe of this report. This unbiased perspective intends to showcase the flaws discovered in the present functioning of the company, evaluating the causes, presenting solutions and strategies of implementations, defining, and justifying metrics (quantitative and qualitative) that can be a mark of accomplishment.

To properly assess the problems the company faces, one must do an external and internal analysis. The external analysis will be done in this chapter, with the description and breakdown

of the processes that were thought to be the most important and problem-inducing. The internal analysis will be performed according to the methodology (chapter 4), provided from the critical assessment of the external factors presented in the following sections.

3.2. Roles within the company

There will be mention of stakeholders and several roles throughout this report. To give a better sense of understanding and to align the report with what's the norm in the studied company, one can define the roles found within the company:

Developer – Responsible for developing features required for the project (can be a new functionality, a bugfix, or unplanned support).

Buddy – Team member responsible for the onboarding of new members.

Agile Coach (Usually called Scrum Master) – Helps make sure the team/company is being responsible and effective in adapting the Agile methodology. His roles can include coaching the team in Agile/Scrum methodology, discovering, and resolving obstacles in a team's functioning.

Tech Lead – Responsible for verifying and inspecting the overall technical and software aspects of the team.

Delivery Manager – Responsible for backlog management and ensuring deadlines are accomplished during the sprint.

Business Analyst - Responsible for analyzing stakeholder needs and business processes to identify, write and validate business/user interface requirements for large software projects.

Platform Support – Working inside and outside of the team, updating, and verifying technical documentation.

Modeler – Team member responsible for compiling and organizing large quantities of data.

Product architect – Team member focused on the design and the UI of the applications.

Product owner – Team member focused on bridging the gap between the team and the customer, gathering, and delivering customer and market's requirements to the team, thus increasing the value of the product created. The Product owner also inserts these requirements into a backlog.

Project Manager – Team member responsible for directing and managing member and the project, running analysis on several processes and key performance indicators.

Quality Assurance Tester – Team member most responsible for finding and testing errors found during the normal functioning of the applications.

3.3. Company Structure MSG Iberia

The company is divided between multiple teams inside it. The departments are divided by nature and by region. FJA is a north American insurance technology company focused only on that continent, while there are other teams, or departments, like VHV and the Product Machine and Portugal (PT) team that are focused on Europe, namely Germany and Spain, as can be identified in the organizational chart developed and presented in Annex 1.

During the present work, it was possible to be involved on several different projects inside of the FJA and the VHV team, having acquired comprehensive and extensive knowledge of tools, operations, and interpersonal relationships. To better understand and study the company, several diagrams were composed. A model of entities and relationships in the company was built, to understand the composition of the connection between company (top) to a team (bottom). As will be discussed later, backlog management is the process of gathering, processing, and managing the requirements the client has for a product.

3.4. Global model of entities and relationships in the company

As the company is focused on a Core-Customer strategy, it is mostly dedicated to serving individually and in a custom fashion to their clients. MSG Insur:it, as a company, has services with multiple clients in the healthcare service consulting area, requiring an application that connects the customer and the company's database.

As seen in the model of entities and relationships in the company present in Appendix 2, the team, consisting of multiple sub-teams, such as the QA (Quality Assurance), Support, Business, and DevOps, gathers and manages the requirements in the process of building this application, being responsible for the feasibility and development of the features required and managing the backlog. The Human Resources (HR) department is responsible for receiving, analyzing, and providing the feedback reports provided by staff. The business team gathers requirements from the customer, placing it in the backlog to be managed.

3.5. Model of entities and relationships of backlog management

The model of entities and relationships of the backlog management present in Appendix 3 was made to understand both how the team itself works in tandem with the client to build the application and the functions each role performs. On one hand, the product owner is focused on communicating with the client and gathering the requirements necessary. Then, he digests this information and proceeds to pass it in an understandable and marketable way to the team, inserting these requirements in a backlog. On the other hand, the DevOps (a software engineering methodology which aims to combine the work of development and operations teams by facilitating a culture of collaboration and shared responsibility) and support team develop multiple features, features that are to be tested.

Additionally, the QA team is responsible for finding bugs in these features or in the system, performing unit testing on each feature, as well as release tests and smoke tests overall periodically. Each feature developed and bug found is turned into a "ticket", a storage of the data relevant to the ticket, like estimated time of delivery, assignee of work, user stories, etc., that is then placed into the management tool used by the company. Finally, the tech lead inspects the roughness and technical aspects of the tickets, while the delivery manager, since we are in an agile setting, ensures that the sprint is going according to the planning.

3.6. Sprint planning

Sprint planning is a fundamental phase in the functioning of the company, and overall in Agile methodologies. The organization works in sprints, which can vary from 2-4 weeks and are intended to work on a specific and planned set of features. Before each sprint, as Appendix 4 shows, and usually while the previous sprint is still active, the client and the product owner

engage in multiple conversations, discussing and fine tuning the requirements that will be worked on for the next one. In a sprint planning meeting, the product owner discusses with the team if the requirements are logical, possible, and feasible during the time available to them. As the team estimates ticket time completion and is more aware its own capacity, this is, the working hours available for each member of the team, the backlog items can, then, be properly selected. The product owner is aware of the priority that the client places on each feature and, as such, he will define it for the team. If, by some reason, there is not enough time for every ticket, this ensures that the most important requirements are satisfied for the sprint.

When a ticket is being defined, it is important that it is also understandable as well. If a ticket has multiple tasks, usually more than 7, sub-tickets are created that relate to the original. The tickets are then associated in the creation of a sprint backlog, that is placed in the total backlog as a sub-section. The Agile Coach plans for agile commitments (next sprint planning, stand-ups, sprint reviews and retrospectives) and ensures the dates are fulfilled.

3.7. Ticket Creation

The process for ticket creation has a standardized workflow shown in Appendix 5 and should be fulfilled each time, to ensure consistency across the backlog. The product owner defines the priority, as the result of the conversations with the client. Afterwards, the tech lead of the team verifies if the ticket is valid, this is, if the user story is well-defined and understandable. He then analyses the ticket and assigns it to the team or team member that is better equipped to handle it, depending on if it is each a feature or a bug, and if the ticket is ready to be developed.

Meanwhile, testers are working in parallel with the team, trying to find bugs in the systems or in features that were developed. If a tester finds no bugs with the developed feature, he sends it for review for the product owner, who compares the ticket/feature to the company's standards for being done. He then deploys this feature into the system and closes the ticket. The client verifies if the feature is according to specification, and, if not, the ticket is reopened and worked on once again.

3.8. Onboarding Process

The onboarding process of the company is extensive and layered in multiple phases:

1. Welcome session with the regional manager, where the worker gets aligned with what consists the group of msg, what is the mission, vision, and values of the company. Additionally, the worker gets to know how the company is organized and guidance in becoming a team member.
2. IT session with a DevOps member to setup every technical aspect of the new worker.
3. A HR/Administrative Session for the signing of the contract, to sign some internal documentation, explaining of benefits, salary, so on and so forth.
4. Marketing Session for the company, clients, and products presentation. Introduction of the worker to the perceived and intended company culture, providing a welcome kit, and a series of guidelines on how to represent and promote the company, internally and externally.
5. Team session with an appointed senior member (10+ years of experience in the company) and the buddy. In it, the project where the member will work on is introduced,

showing the tools utilized in the process (more on that in the next sub-chapter). The workspace, the resources and online courses are presented.

6. After these 5 sessions, there is a round of initial feedback after 1 week and another one after a month of working in the team. The rest of the support is supposed to be conducted by the buddy, who will accompany the new member throughout his experimental time at the company.

3.9. Benchmarking the involved teams in the case study

Although the teams and projects might be different, there are some similarities that should be considered in the regular proceedings of such:

- Agile methodology: Be it SCRUM, KANBAN, or SCRUMBAN, every team employs an Agile methodology in their workforce.
- Mandatory days at the office: Every team has a mandatory Office Day, where each member should attend the Porto Office in person, allowing for the team to do the Retrospective, the Sprint Review, and planning sessions in a collaborative, in-person way.
- Confluence: The documentation tool for the team. Be it guidelines for processes, information on tools or previous iterations, or just general data should be stored and found here.
- Enterprise Resource Planning (ERP): The tool used to track what each worker worked on. Every worker action has a code that can be introduced in this tool, which allows for the company to better track the usefulness of the worker's actions and to have a billing explanation to the customer.
- Microsoft Teams: Tool used for communication between workers. In it, meetings and scheduling of meetings and events is conducted.
- Service Store: Tool used for acquirement of software or hardware. In it, a worker inputs a request and, after approval, gets the receival of the order in due time.
- Outlook: Tool used for the exchange of e-mails between workers. More formal requests or communication to and from external parties are conducted through Outlook.

3.9.1. Benchmark FJA

The FJA division of the company is comprised of four teams, as shown in the Company organization diagram (Appendix 1). Some teams have a smaller workforce, with around 8 members, while others are composed by more than 20. Each team's objective is to deliver service to the US-based companies.

In general, the FJA teams work on a SCRUMBAN basis dealing with 2-weeks sprints. The sprint begins with a planning session by the senior member of the team. The due date of the tickets for the current sprint are set, with the expected date recommended to be 1 day ahead of the real estimation, to employ flexibility and security in the delivery of each ticket. The start dates for the tickets and the assignees should also be decided. Complex tickets are divided in smaller, sub-tasks. At least 1 hour per sprint is used for review of code quality.

A KANBAN board like the one present in Appendix 6 is then set for the sprint, with different phases being defined, from an "open" ticket to a "closed" ticket. A ticket can only be started worked on if the type (bug, task, test) is defined and if the user stories (the description of the steps taken by the user to achieve the goal of the ticket) are well-written and explanatory. The

technical details and requirements should also be detailed to ease the understanding of a developer/tester.

Software engineers then begin resolving tickets. They test the developed and implemented features in one environment, before they pass it down to the QA testers. The QA testers have the obligation of testing each ticket and focusing on the most important areas to test, most known as release tests or smoke tests. The backlog (Appendix 6) is then worked on.

The release tests are documented and have a series of steps, detailing the procedure of several functionalities that are required to work for the system to be considered "releasable". Smoke tests are tests carried out with the user in mind. A QA tester carries out a series of tasks from one end of the process to the other, in a somewhat exploratory way, attempting to find errors and validating the effectiveness and well-functioning of the system. Testing is done in two different locations (Iberia department and India department).

The FJA applications are, in general, stable. They suffer less severe updates and are just incrementally built upon, leading to a steady and gradual process. Despite having 3 environments, only one is mostly used to develop and release patches, with deploys being mostly scheduled for a time where less people are working (even if that window might be lessened by the fact that there are teams in 3 very different time zones present).

3.9.2. Benchmark VHV

The VHV team works in a different setting, opting for 2 separate teams: Planned and unplanned. The planned teams work in prepared items, with less flexibility and pace required, utilizing SCRUM. The unplanned team works in last minute issues of development and support, requiring a quick turn-around, hence working in Kanban.

Like stated before, a team day in the office is required every 15 days for each team, coinciding with the Retrospective and sprint planning. There is an additional suggested day of presence, which coincides with the first Friday of the sprint. This is meant to improve connection, knowledge, and lessen the learning curve.

The team works within scrum framework guidelines, and if the sprint/iteration is not going well, extra help from the unplanned/support team might be requested. Team shows burndown chart in the middle of the sprint, and analyses if further actions are needed.

There is real-time error analysis, where a sudden blocking issue in the application/environment is identified and becomes a priority issue to be fixed urgently, analyzed immediately by the support team. If the problem is not easily fixable and might take a while to solve, a ticket should be created. This will cause visibility to the work done and sets an archive if the issue turns up again.

The daily meeting is where each team member explains their plan for the day, that might change depending on other factors. When a team member goes on vacations, he adds a comment on the tickets he was working on prior to the vacations, explaining the status of such and what must be done by the next assignee. At the beginning of the second week of the sprint, a reevaluation is done, to assess if the development team needs help from the support team. If a ticket has an estimated work time of more than 2,5 days, it should be divided into sub-tasks. Clear tickets are sent to the planned team, uncertain tickets are sent to the unplanned team. The sprint review

presenter is responsible for making sure that the environment for demonstration is ready, up, and stable during the sprint review frame.

As the team works with several different environments, the software engineers should deploy the tickets to the environment explicitly asked to. In turn, QAs should always add videos or screenshots of identified bugs or issues, facilitating the Software Engineers' work.

The planned board is "recycled" every sprint, meaning that no ticket should remain at the end of the sprint. The ones that do, should either transfer automatically to the next sprint or be placed in the backlog (in the example of a ticket being blocked by another). The support/unplanned board, working in a Kanban setting, does not suffer this process of recycling and is more flexible and tolerable with timings.

The VHV application consists of 5 active and testable, independent environments that are in constant development and state of release. This type of development accrues benefits and disadvantages. On one hand, if one of the servers fails or encounters some issues, one can easily take its place in development and release testing. Changes can also be implemented without the client or elements even noticing, not even having to consider working hours, like in other teams. Different environments can be utilized by different teams (planned and unplanned). On the other hand, it is harder to synchronize information and to keep focus on what environment you are and if the changes are taking effect.

Ultimately, each team in the company, although working with similar tools, has a means of functioning that differs from each other. There is not a universal standard that each of them must follow, so, as the team members and leaders differ, so do the measures that each team employs. Since they all follow a fast, flexible, and iterative strategy from the Agile methodology, the processes are easily and frequently changed. This is useful for productivity and efficiency but can lead to confusion and efforts wasted.

Now that the company and their objectives of research are characterized and that the research gaps in the literature review are identified, the problems must be studied, identified, and solved. In the next chapter (Methodology), there will be a short comparison between existing methods for the study of phenomena. Subsequently, the chosen methodologies for this research project will be identified and explained.

Key Takeaways of this chapter:

- A comprehensive observation and characterization of a company and a company's processes is essential for its understanding.

- The company is comprised of several teams with similarities in functioning, but key differences that also change the alternate management strategies that each project manager employs.

- The process areas that one can focus for the methodology and posterior chapters are: 1) backlog management; 2) sprint planning; 3) ticket creation; 4) onboarding process; 5) distributed environment of a global company.

4. Methodology

The methodology section of this work will first present the most common research tools and strategies that exist. In sum, the methodology of a project/dissertation is the strategy chosen to better understand and answer the research questions, guiding the research work (Long, 2014). It is where a researcher follows the process of collecting, analyzing, interpreting data, and applying it to the research objectives, with the intent of finding answers (Leedy & Ormrod, 2001).

4.1. Existing Methodologies

There are three main and popular approaches to a methodology: The qualitative approach, the quantitative approach, and the mixed-methods approach (Williams, 2007). Quantitative research is mostly focused on the measurement, the gathering and analysis of statistical, concrete data (Long, 2014), while qualitative studies tend to approach the work from an exploratory aim (Corbin & Strauss, 2008).

The mixed-methods approach occurs when qualitative and quantitative are combined and occur simultaneously in a project. To choose between methodologies, a researcher must analyze the research questions and predict the type of data that will be appropriate to answer them (Williams, 2007).

Quantitative Research:

Quantitative research is classified as a deductive approach, in the sense that it intends to be a logical and objective top-down method that tries to expand existing theories (Leedy & Ormrod, 2001). The research is meant to be "independent of the researcher" (Williams, 2007). It follows a simple guideline of setting the research objective and questions, defining the dependent and independent variables, the analysis of the data collected, and a statement/conclusion deducted from the analysis (Williams, 2007). This approach is only utilized when the studied subject is measurable (Watson, 2015).

There are different types of measurements used in quantitative research, from the measurement of primary information (e.g., experiments and surveys done by the researcher) to the analysis of secondary information (e.g., census, previously gathered statistical data) (Williams, 2007). One can approach the quantitative method from the survey design perspective, where one samples "data from respondents that are representative of a population" (Williams, 2007), or one can approach the research from an experimental design purpose, where one (or more) independent variable is systematically manipulated and the effect it brings on one (or more) dependent variable is studied (Watson, 2015).

Quantitative research is advantageous in the sense that it gives us an experimental context of real-life, being systematic and possible to reproduce (Williams, 2007). However, there are two issues that one must consider when choosing the quantitative approach: The possibility of statistical error (either for human or measurement error) (Watson, 2015) and the possibility of confounding correlation for causality (Williams, 2007). Just because a variable is related to the phenomenon that one is studying, it does not necessarily mean that changes associated to that variable will cause a significant impact on another related variable. As such, one must not only identify variables, but study the validity correlation between them before proceeding with the gathering and analysis of data (Williams, 2007).

Qualitative Research:

Since the qualitative approach is intended in more of an exploratory sense, it can be considered an inductive approach, as it aims to be a bottom-up method that draws general conclusions from specific observations (Long, 2014). It is a methodology that occurs in a natural setting and allows the researcher to posit himself as observer (and sometimes an active participant), giving him a "high involvement" in the experience (Creswell, 1994).

A qualitative methodology "involves purposeful use for describing, explaining, and interpreting collected data" (Williams, 2007). A qualitative researcher does not believe the world has a fixed standpoint, but states that, since the natural system that we observe is composed of social being and interactions, reality and the meaning of experiences will be fundamentally built upon those factors, developing multiples "interpretation of reality" (Merriam, 2002). A qualitative approach might have "five areas: case study, ethnography study, phenomenological study, grounded theory study, and content analysis" (Williams, 2007).

A case study is focused on "a person, program, or event" (Creswell, 2003). The data collection is "extensive" and can come from multiple sources, and the researcher must have a hands-on approach, with the conclusions being "lessons learned, or patterns found that connect with theories" (Williams, 2007).

An ethnography study is different from a case study, in which the research studies "an entire group that shares a common culture" (Leedy & Ormrod, 2001). Ethnography tries to understand culture and culture change, as well as the causes for change, and it might lead to over-specified conclusions that might not be generalized (Williams, 2007). For ethnography purposes, one must justify the study, describe the group and the method of study, as well as the evidence gathered to support the conclusion and the answers to the proposed research questions (Williams, 2007).

The ground theory approach intends to induct a general theory "grounded" from the data analyzed (Merriam, 2002). A phenomenological study is focused on "understanding an experience from the participants' point of view" (Leedy & Ormrod, 2001). It is meant to understand why a participant describes or perceive the events the way he does and how his awareness of "intentional" consciousness alters his connection and what he feels towards the experience (Merriam, 2002).

One can also argue that all of qualitative study, being subjective, exploratory, and perspective-based, is of the phenomenology nature (Merriam, 2002). At last, one can perform a content analysis study, which can be described as "a detailed and systematic examination of the contents of a particular body of materials for the purpose of identifying patterns, themes, or biases".

Qualitative research provides a more described understanding than quantitative research and is more flexible, especially in understanding causality and different perspectives (Merriam, 2002). However, subjectivity brings the constraint of being hard to generalize the findings, as well as enhancing the probability of biased results by part of the researcher (Hood, 2006).

Mixed-Methods Research:

It appears, then, the mixed methods approach, the combination of "quantitative and qualitative research approaches in a single research study" (Creswell, 2003). Researchers "analyze not only numerical data (…) but also narrative data, which is the norm for qualitative research in order to address the research questions defined" (Williams, 2007). The mixed methods approach does not eliminate the other two, it only intends to "draw from the strengths and minimize the

weaknesses of the quantitative and qualitative research approaches" (Johnson et al., 2007). It is important, though, that the researcher has logical flow and outline of the methodology, and that he does not choose the mixed method without a proper justification, providing it to the reader to increase validity and transparency (Halcomb & Hickman, 2015).

The researcher must also identify his skillset and be critical in assessing if it is appropriate to conduct mixed research (Creswell & Clark, 2011). The mixed methods approach is valid and brings a combination of components that might bring stronger results, but only if well sequenced and justified (Almalki, 2016).

Design Science Research:

Design Science Research (DSR) (March & Smith, 1995; Hevner et al., 2004; Kuechler & Vaishnavi, 2008) is research that "often begins by identifying and representing opportunities and problems in an actual application environment", (Hevner, 2007).

DSR intends to create artifacts, that can be constructs, models, methods, and implementations that are innovative and valuable in such a way that they contribute to advance the field where they are inserted (March & Smith, 1995; Hevner et al., 2004;).

Constructs are problem-solving languages, and languages are utilized by models to "represent problems and solutions" (Winter, 2008).

Methods are processes that guide problem-solving processes (Winter, 2008). For this to be valid, the artifact produced should be connected to the problem definition and research objectives, and the relevance to the real-world practice should be stated (Gregor & Hevner, 2013).

The DSR Guidelines (Hevner et al., 2004) can be found in Appendix 7. Not only relevance, but rigor is necessary (Winter, 2008). Choosing DSR, a researcher should go through the following process shown in figure 2, as identified by Peffers et al. (2007):

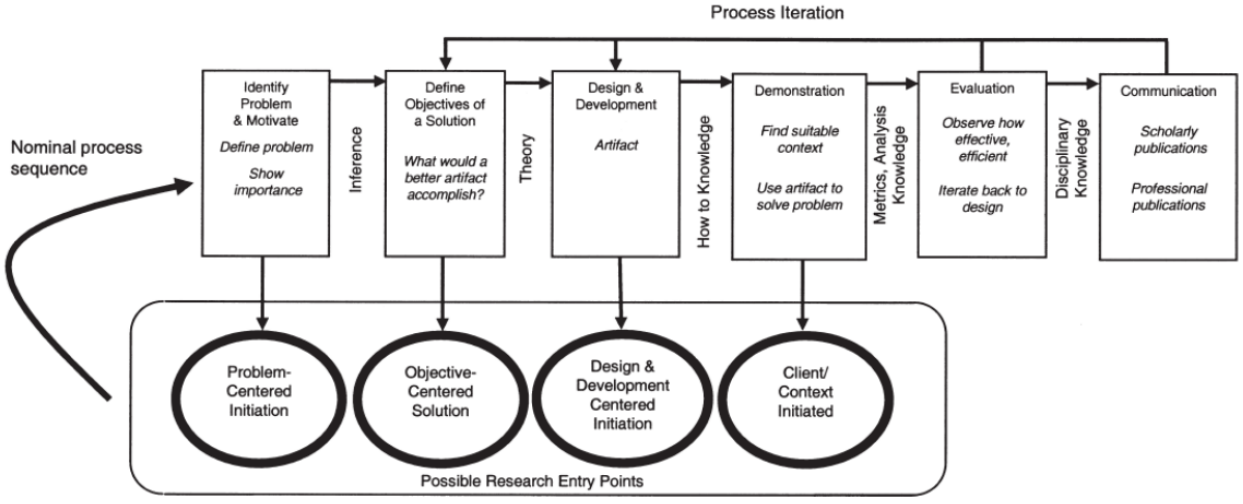


Figure 2 - Design Research Science Process (Peffers et al., 2007)

4.2. Chosen Methodology

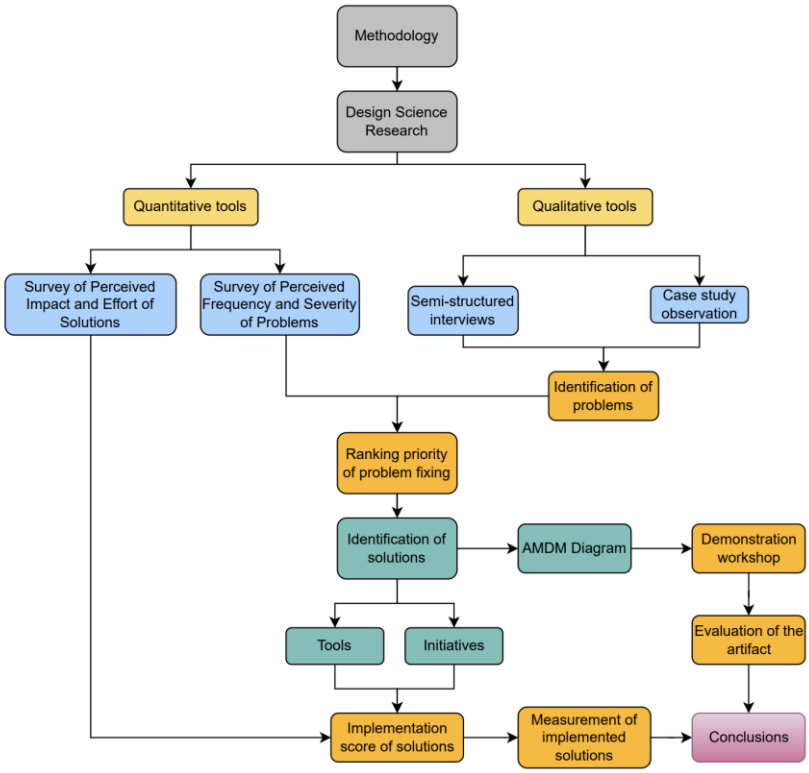


Figure 3 - Outline of the methodology chosen for the project

The methodology chosen to accomplish the research objectives and answer the research questions, in regard to the context presented in chapter 3 and the literature gaps identified in chapter 2 was the methodology represented by figure 3 which can be considered a case of Design Science Research.

As identified by figure 2, when employing Design Science Research, one must take several different steps. An internship was being done in parallel with the writing of the dissertation, and, as such, there was plenty of opportunity of immersion in the culture of the company, letting one be a careful observer and an active observant, studying the company as a case study.

The first step is to identify the problem, its causes, and the weight they have in processes. This will be conducted in a mixed research fashion, with the qualitative aspect of the case study observation of the company and the interviews conducted with the appropriate consent (Appendix 8) to the relevant stakeholders (Appendix 9), combined with a quantitative tool based on surveys answered by the interviewed stakeholders, ranking the frequency and severity of the problems identified (Appendix 10 and 11). Appendix 12 shows roles of the stakeholders in the company, providing credibility to their opinions, the duration of the interview and the main insights that were able to be extracted. This allowed for the understanding of the flow and functioning of the processes in the company, starting to identify areas of improvement and fundamental areas that need to improve for a better efficiency and effectiveness.

The semi-structured interviews conducted allowed for an even better understanding and validation or disproval of the assessments, while also providing empirical and trustworthy information from experienced and knowledgeable individuals to back the findings. The qualitative process permitted the identification of several problems in chosen areas (more on that in chapter 5, table 1, 2, and 3) and the causes of it. As proven by the semi-structured

interviews, that allow for a better grasp of personal and social matters (DiCicco-Bloom & Crabtree, 2006), different team members in different areas of expertise and responsibility noted different problems and stated different levels of being affected by them.

Solutions are, then, studied, defined, and developed. The proposal of solutions came in two ways: First, tools and initiatives that were meant to solve each cause, with the purpose of mitigating or eliminating the problems found in the company (chapter 6). After a workshop where solutions proposed were introduced and discussed, the workers filled out a survey (Appendix 12 and 13) with what they felt would be the impact of the solutions and effort required for these solutions (More on chapter 7). The different solutions had metrics devised (qualitative or quantitative) associated with them and the perspective of success associated with it (Chapter 7 and Appendix 23).

This combination of tools and initiatives allowed for the design of the development of the final artifact in this research project, the Assistance to Project Managers Decision-Making diagram (Chapter 8). This decision support system built as a decision tree with several different essential factors in project management, focused on the case study setting, intending to solve issues of confusion, lack of transparency or trust, and intending to implement a culture of traceability, consistency, and standard behavior. The artifact was then presented, once again, to the relevant stakeholders in the setting of a structured workshop that was meant to analyze and refine the artifact and study its validity to the company (demonstration stage).

Although this artifact was designed with this specific company's research objectives and flaws, there are possibilities of expansion or generalization in the field of project management (Chapter 9). At the moment of submission of this dissertation, the artifact was at the "evaluation" stage, with the company implementing it in certain teams and observing the results. The evaluation stage explores "the performance of a solution technology in its real environment" (Pries-Heje et al., 2008), which, in this case, means the studied company. This "naturalistic observation" takes time and reflection, with the metrics for success having to be defined *a priori* of the implementation, while the conclusions of success (or failure) can only be stated *a posteriori*, which means that building an artifact with DSR is an iterative and time-consuming process.

In a research paper, the methodology is the backbone of all further research and conducts the search for questions, the findings of the inserted context, and the answers/solutions to the research objectives and questions (Kothari, 2004). The methodology defined in this work was meant to be extensive, comprehensive, and holistic, going from the social and individual analysis of a worker to the breakdown of several important processes in the company.

Key Takeaways of this chapter:

- There are several different existing methodologies, comprised of qualitative, quantitative and mixed-methods tools for research.

- The methodology chosen for this project will be Design Science Research, with a mix of qualitative and quantitative tools, used in alternative stages (figure 3). These will allow for the identification of problems and development of solutions.

- Design Science Research will be employed to create an artifact (decision support model) that gives project managers a framework of team management strategies.

5. Problems found

The following chapter will present the results brought forth by the methodology, as known as the sectors that were more important to study, the problems found, and the causes related to it. As a result of the information acquired through the literature review, the study of the characterization of the company, direct observation, mainly through the semi-structured interviews and informal conversation, one can encounter problems mostly in 3 main areas.

The problems identified have different causes that must be studied so they can be individually resolved to mitigate the negative effects of the problems. The tables below translate into a visual meaning what transpired from all the aspects of the methodology, with a great focus being put in the insights from the interviews, as they were the main source of customized information and expert knowledge.

5.1. Sprint planning/data management

Table 1 - Problems and causes found in sprint planning/data management

P1	**Lack of sense of direction**	
	C1.1.	Long-term roadmap not connected to sprint planning
	C1.2.	Daily meetings with focus on only one part of the team
P2	**Improper Documentation standards**	
	C2.1.	Outdated information
	C2.2.	Lack of organized information
P3	**Poor Client-Company connection**	
	C3.1.	Inaccurate requirements by the customer part
	C3.2.	Estimation of hours is imprecise
	C3.3.	Backlog accumulates unassigned and unworked tickets

Sprint planning and data management is an important phase of the process that might generate problems regarding the connections between information. As a product owner must build a bridge between the client and the team, one must treat information as a privilege and as fragile. Requirements can be easily misconstrued.

Since the company is focused on select customers and intends to harbor them for multiple years, one project/application can suffer only small increments in a large interval of time. Refinement, or optimization, is harder to visualize in a grand scheme, and when a team works in 2-week sprints, usually, the overall vision of the project might be lost. This leads to confusion or lack of sureness on why this feature should be implemented, or what the overall goal of the project is (C1.1.).

Since the focus of the company is in *developing* an application, sometimes, the daily meetings (that are mandatory for the team as a whole) feel too focused on the developer's side, with little to no connection with other roles, like testing, for example (C1.2.).

In the specific case of this company, the repository of documentation/information is outdated, with pages dated years back without information (C2.1.).

There is also difficulty in finding the proper places where information is stored, even if it was regularly updated (C2.2.).

The client usually does not give complete or accurate requirements. This might be because they do not have technical knowledge, they do not understand how the application functions, or do not know how to express themselves. In the same vein, the client also retains all the power in this exchange, so, lack of or improper communication is usually blamed only on the team (C3.1.).

Improper communication of requirements makes it harder for the team to define estimation of working hours necessary for completion of a ticket, which might lead to incomplete work, poor estimates, or too many hours estimated for the reality (C3.2.).

In this specific case, the backlog also gets flooded with tickets that end up going unassigned. This might be because they are blocked by the client for some reason, are only being planned for further iterations, or for lack of communication. This results in dozens of tickets sitting idle without any work being done on them for days, weeks, possibly even months, clogging the backlog and making its management even harder (C3.3.).

5.2. Ticket management

Table 2 - Problems and causes found in ticket management.

P4	**Disperse efforts**	
	C4.1.	Multiple environments for development
	C4.2.	Multiple environments to be tested
	C4.3.	Repetitive and long manual testing
P5	**Difficulty in communication**	
	C5.1.	Lack of synchrony between team members and environments being used
	C5.2.	Tickets added mid sprint without criteria
P6	**Conflicts in ticket management**	
	C6.1.	Tickets being worked on by people with different time schedules
	C6.2.	Same person doing the same role in different teams

The ticket management phase of the process is the natural following of the sprint planning, where the team puts in action what they planned, focusing in resolving the tickets according to their priority and/or impact.

Most teams work on several different environments for the application. Although this works well to reduce disruption of services and the increases readiness of the servers, it also leads to disperse efforts. A developer working on an environment might have to employ every solution on every environment (C4.1.).

Too many environments also make communication between teams difficult, especially in a bigger team. One must always communicate which environment he is developing/using/testing. One must always warn the team if the server will be temporarily down and, if any communication error occurs, once again, time and efforts can be wasted (C5.1).

A tester might have to test a feature or a bug in several environments to understand if the team is dealing with a local or global bug, resulting in time wasted, ranging from minutes to hours (C4.2.).

Flexibility brings many advantages, but the fact that there are several different working hours (especially accounting for distributed teams), makes ticket management harder. For example, if a person works from 07h-16h and another works from 14h-23h, the time window where they can communicate in synchrony about a ticket that might be going back and forth (for example, developer and tester) is 2 hours, which might cause delays in ticket completion (C6.1.)

There are also people working the same role on different teams. A developer is not necessarily attributed to just one team. This might cause two issues: If a developer gets assigned two different tickets on two different teams with two different delivery managers, which one does he start first? The conflicts arising might be hard to solve. More, there are different boards for each application, and a person that is used to having tickets in alternate boards might not notice that they had a new issue assigned. (C6.2.)

There are, occasionally, tickets added mid sprint, as they become necessary. They could be new features, bugs found, or regular fixes. If they are not accounted for, the working hours for the sprint will accumulate and might lead to incomplete work. (C5.2.)

There is also little to no automation in the testing of tickets. Most of the brunt force of testing that takes time are release and smoke tests, which are mostly the end-to-end process of the application. These usually have plenty of documented steps, and are repeatable, unlike spontaneous bugs that might occur in the day-to-day work. Manually doing release tests and smoke tests is time consuming, tiring, and might be prone to human error. (C4.3.)

5.3. Distributed teams

Table 3 - Problems and causes found in distributed teams.

P7	**Disconnect from the project**	
	C7.1.	Lack of sense of long-term direction of the project, team, or company
	C7.2.	Value of the work hard to gauge
	C7.3.	Isolation, burning out, increased stress
P8	**Lack of cohesion**	
	C8.1.	Lack of sense of union in team
	C8.2.	Tension between departments
	C8.3.	Harder onboarding process
P9	**Communication**	
	C9.1.	Communication takes longer in remote settings
	C9.2.	Culture and standards are different between teams/Countries
	C9.3.	Harder security of information

Distributed teams face several problems, some of which have already been found and described in the literature review (more on that in chapter 5.4).

Focusing on this case study in particular, distributed teams might lead to several problems:

Distributed teams might have a sense of loss of direction in the long-term goal/vision of the project or the company, as they are not as aware of the project or connected to it as they could be in a physical environment. (C7.1.)

There's also the loss of the sense of union in a team, as they do not see each other regularly, they tend to communicate less and, when they do, they do it through a screen. (C8.1.)

There is also less connection between departments. If a person does not meet often with their teammates from the same project, they do it even less with other teams. To reduce office occupancy, mandatory team days are spaced throughout the week, thus reducing contact between departments, which might lead to depersonalization between teams, with tensions and rivalries occurring. (C8.2.)

There is also a mismatch between people who work from home and those who work in the office. Those who work from home might not connect as much as people who work in the office, who might bond in the sense that they choose the same working method and are more easily communicating. (C8.1./C8.2.)

Communicating in a fast fashion can also be easier in an office setting. One can dislocate himself from a section of the office to another quickly, while, on the other hand, virtual communication might be harder with schedules having to match and online tools having to be used. (C9.1.)

Dealing with such a multifaceted company such as this one, that works with people from several countries and time zones, like Portugal, Germany, India, the United States, not only the schedules are different, but the working culture is also not the same and that might lead to

conflict or a mismatch of information between teammates. Standards of procedures are different, and it might lead to lack of trust and collaboration. (C9.2.)

When someone is working remotely, it is also harder to monitor work. It is not easy, without individual metrics, to understand if someone is working or not if someone is at their station or not. On the other hand, the work that is effectively done might also be devalued, as without individual metrics, the success of someone's work is not as measurable as someone who management sees in the office. (C7.2.)

Remote working also brings along a harder process of onboarding. Workers that are virtually distributed take longer to meet their colleagues, to talk and connect socially with them. They also need some resources that are only available physically (e.g., working material, like company laptop), and software that might take more time to be obtained/installed in a distributed setting. (C8.3.)

All of this contributes for harder security of information, as it travels through many more channels than usually would locally. (C9.3.)

Isolation, burning out, stress, uncertainty… these and more factors associated to distributed work could lead to lack of productivity and distractions increasing. (C7.3.)

5.4. Results Discussion

After identifying the problems, it can be important to establish how they were defined and through which tools. There were problems that were asserted through the observation and interview approach and corroborated by the literature review performed in chapter 2, while others were only found after proceeding with the observation and interviews, not being presented in the literature review stage and showing the presence of literature gaps.

Problems identified through the methodology and corroborated by the literature:

- C3.1. – Internally, it was stated that requirements are not frequently supplied with accuracy or completeness. The literature confirms this issue, since agile is identified as a turbulent environment amid flexible requirements, sometimes requirement gathering is called as "lazy", as identified by Paetsch et al. (2003), as well as delayed and not as well-structured (Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001).
- C5.1. – Internally, difficulties in synchrony in communication, technology and environment used were identified, with effort and time waste concerns. The literature aligns with this observation, by stating that an uncoordinated focus in technology might generate confusion between members (Lilian, 2014) and that different types of tasks affect coordination between co-workers (Espinosa et al., 2007).
- C6.1. – In this global software company, zone differences or flexible working hours might pose a problem if workers focused on the same task are not matched. Ford et al. (2017) pose this as a possible problem as well.
- C7.2. – Through the interview process, mostly with HR professionals, there was a sense of difficulty in valuing someone's work while they are working remote. People also felt that the work that they did was not being appreciated. The literature states that this can lead to a lack of motivation (Gagné & Deci, 2005).

- C7.3. – There was a higher possibility of isolation and/or distractions faced by remote workers identified in the interviews. Chinowsky & Rojas (2003) agree and state that, to reduce the effect of spatial distance, workers have to be empowered.
- C8.1. – Different configuration of workers state that they feel less connected to each other, as well to their team as a whole. Remote workers also only see or interact with other people through a screen. Breu & Hemnigway (2004) also identified this possible lack of connection between remote workers.
- C8.2. – The 8.1. problem is magnified when dealt in comparison between teams. Connaughton & Shuffler (2007) argue that conflict must be dealt with between teams.
- C8.3. – Although the company has a well-structured onboarding process, as described in chapter 3, there were still some observations and declarations from interviewed stakeholders showing interest in improving the process. The literature identifies the higher costs of hiring and training individuals in a remote setting (Ford et al., 2017)
- C9.1. – The interviewed stakeholders showed some frustration towards remote communication. They stated that office communication was faster and intuitive. The literature shows a different outlook, stating that, barring technical difficulties, communication is easier and quicker in remote settings (Connaughton & Shuffler, 2007; Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001; Marlow et al., 2017)
- C9.2. – Dealing with people from so many different time zones, workers in the company felt difficulty in connecting to every different work culture and standards. Ardichvili et al. (2006) believe that people originating from different cultures have more difficulty in knowledge sharing, an outlook shared by Hinds & Bailey (2003) and Ford et al. (2017).
- C9.3. – Security of information is a crucial concern in the company, but, if not kept in check, it might generate potential leaks or mishandling of information. Ford et al. (2017) identifies virtual teams as data security dangers, with high possibility of information leaks.

The remaining identified problems were barely or not even mentioned in the studied literature, providing a confirmation of the existing literature gap in identifying concrete and specific problems and the reasons for them to happen in a global software development setting.

Key Takeaways of this chapter:

- One can mostly identify problems in three main areas:

    1) Sprint planning/Data management.

    2) Ticket management.

    3) Distributed teams.

- Each problem can be constituted of different causes that must be individually studied, as they have possible different solutions.

- The problems identified came from multiple sources: Literature review, case study observation, and the conducted interviews.

- RQ1 and RQ2 are answered in this chapter.

- Although the literature identifies some problems, there is still an existing gap in identifying specific problems.

6. Proposal of Solutions

The identification of problems in the past chapter guides the structure of the rest of the report, answering RQ1 (“What are the problems affecting software development global companies in their processes?”) and RQ2 (“What are the causes of the identified problems?”). From now on, the focus is on answering the remaining research questions, starting by identifying possible solutions.

6.1. Frequency and Severity of the problems found

There is a limited timeframe for this project, which means priorities for solutions must be analyzed and defined. Due to natural observation, conversations, and the informal interviews performed, as well as a survey conducted to interviewees (Appendix 10), a notion of frequency in which one evaluates how often a cause created a problem and the severity of each cause in the natural functioning of the day-to-day work of a worker was established (Figure 4). For better visualization of the results (Appendix 11), this table is represented in the following graph:

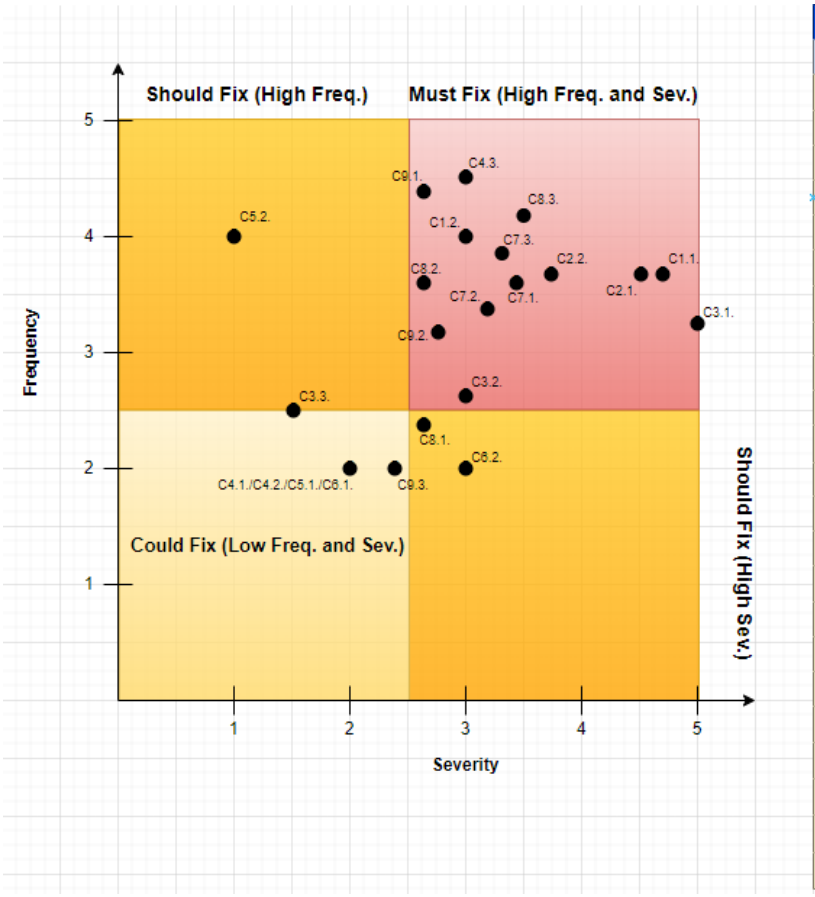


Table of Causes	
ID	Description
C1.1.	Long-term roadmap not connected to sprint planning
C1.2.	Daily meetings with unbalanced focus
C2.1.	Outdated information
C2.2.	Unorganized information
C3.1.	Innacurate requirements by the customer part
C3.2.	Estimation of hours is imprecise
C3.3.	Backlog accumulates unassigned and unworked tickets
C4.1.	Multiple environments for development
C4.2.	Multiple environments to be tested
C4.3.	Repetitive and long manual testing
C5.1.	Lack of synchrony between teams and environments in use
C5.2.	Tickets added mid sprint without criteria
C6.1.	Different time schedules between teammates
C6.2.	Same person doing the same role in different teams
C7.1.	Lack of sense of long-term direction of the project
C7.2.	Value of the work hard to gauge
C7.3.	Isolation
C8.1.	Lack of sense of union in a team
C8.2.	Tension between departments
C8.3.	Harder onboarding process
C9.1.	Communication takes longer in remote settings
C9.2.	Different culture and standards between teams/countries
C9.3.	Harder security of information

Figure 4 - Frequency and severity of the problems found (With label)

The following graph is divided into 4 sections:

1. Must fix: The causes in this section occur in a high frequency and with a high impact on the functioning of the worker throughout the day, being a priority for fixing and solution identification.
2. Should fix (High Frequency): These are causes that are identified to occur often, with low to mid-impact to the worker during the day. These are problems that should be fixed, however, are not the priority.

3. Should fix (High Severity): These are causes that, although not as often happening, have a high impact in the workers' day when they do occur. Once again, to lessen this impact, these causes should be fixed, not being the main priority.
4. Could fix: These are the causes that were identified as problematic, not happening often and with much impact in the natural functioning of the work. They should be looked after and solved, but only after the other problems are looked at in more depth. As such, the cases identified in this section will not have as much focus as the rest.

6.2. Solutions Identified

The following step for this case study was to use the same resources (state of the art, observation, informal and unstructured conducted interviews) appropriate in identifying problems to identifying possible solutions. The solutions were identified in the three main areas chosen to study (Sprint planning/Data management, ticket management, distributed teams). The condensed version of the solutions identified and proposed can be seen in the table 4 below:

Table 4 - Summary of solutions found

ID	Description
I - Sprint Planning/ Data Management	
S1	Tool of alignment between the project roadmap and the sprints planned
S2	Protocol of liaison between Client, Product owner, and the domestic team
S3	Periodic meetings for alignment between the project roadmap and sprint planning
S4	Review of the daily meetings functioning
S5	Tool of data management
S6	Implementation of a homepage for the repository of documentation
S7	Creation of a template document for requirement registry
S8	Development of a tool for backlog management
II - Ticket Management	
S9	Establish a central display of environments
S10	Establishing a documentation of norms for environment management
S11	Development of a tool for automated tests
S12	Incentive for individual ticket centralization
S13	Establish documentation for ticket management
III - Distributed Teams	
S14	Establishing specific goals, Objectives, and Key Results
S15	Initiative of projects focused on inclusion and team-building activities
S16	Betterment of the onboarding process
S17	Standardized information page regarding a distributed company's environment
S18	Creating distributed environment training workshops

6.2.1. Sprint Planning/Data management:

S1 - Tool of alignment between the project roadmap and the sprints planned: Development of a tool that compiles and displays the long-term roadmap of the project in several small stages that can be connected to the different planned sprints. Features, expectations, and results expected can be inserted. This tool aims to connect the overall arc of the project with the sprints. This solution can also help the employee to track back with his implementations, to connect the dots between future features and what could block them from previous iterations.

S2 - Implementation of a protocol for a liaison between Client, Product owner, and the domestic team: In the beginning of a project, the requirement gathering process and communication between teams can be difficult. In early stages, the idea is to implement a protocol for physical connection between a relevant stakeholder of the domestic team, the client, and the product owner. This solution aims to lessen the communication barriers in distributed teams, to facilitate early requirement gathering, and to provide a stronger connection between teams, the clients, and the project.

S3 - Implementation of periodic meetings for alignment between the project roadmap and sprint planning: Depending on the project and the team, defining a periodic meeting where the Project Manager, in tandem with the Product owner, demonstrates and compares the overall project roadmap with the previous and future sprints. This solution aims to provide a structure between the long-term vision and the short-term vision.

S4 - Review of the daily meetings functioning: Since the meetings are almost always entirely focused on the development side, the sprint planning should also have a dedicated time slot for stand-ups where team members review the backlog and define the need for certain roles to be present during dailies. This solution aims to keep the daily meetings to the duration and members necessary, taking the most advantage of the time for each member of the project.

S5 - Tool of data management: Development of a tool that allows for a better management of the data. This tool will implement routine checks to identify outdated information, will assess the priority for update and maintenance of the information provided in general, and will assign a member of the project (previously appointed) to update the repository. A scheduled and frequent review and organization of the information will also be appointed in this tool. This solution aims to stop outdated information from accumulating on various topics and will organize the information.

S6 - Implementation of a homepage for the repository of documentation: Per the observation and interviews done during the time of this case study, one of the biggest concerns and complaints were regarding the "mess" caused by the unorganized content in different facets of the company. The consensus was that, even if the documentation was up to date, it still would be hard to access. As such, one of the solutions proposed is to implement a centralized index page for the documentation management tool of the company. This allows for members to access information with ease and faster. It creates a standard for documentation that is easily maintained and controlled. It is also more intuitive to add new content without affecting the already existing one.

S7 - Creation of a template document for requirement registry: Development of a template presented to the customer with several different pre-set attributes. This will then be discussed with the product owner, which will later discuss it with the team. This solution aims to provide

a registrable and storable document for which the team will be able to look for and reassess their vision of the ticket and will reduce uncertainty in communication and improve traceability.

S8 - General backlog refinement meeting: Although some of the teams already have backlog refinement meetings during the sprint session, the suggested solution would be to periodically (this timeframe is project dependent) have a meeting for the refinement of the entire unassigned backlog, conducting regular check-ins, prioritizing tickets, and assigning team members to work on them. Any tickets that are not worked on for too long and are not prioritized should not be cluttering space. This solution intends to ease access to pertinent information, and to organize the tickets for validity and pertinence.

6.2.2. Ticket Management:

S9 – Establish a central display of environments: Establishing a global and easily accessible tab focused solely on environments currently being worked on, being deployed, or being tested will allow for the team to work in tandem easily. This solution aims to synchronize team members and the different environments used, as to allow the existence of multiple development and testing grounds without the waste of time or lack of communication associated with it. This tool might have the status of each environment, version control, metrics, and change logs. This solution aims to increase ease of communication and synchronization between teams without the need for direct communication every time a change is required in an environment.

S10 – Establishing a documentation of norms for environment management: Establishing a tab with norms and rules necessary for environment management. Steps to be followed before proceeding to developing/testing on an environment, as well as the information necessary to retrieve or provide before and/or after developing/testing. This solution aims to ease the process of environment management, to standardize it, and to turn it more understandable for experienced and new members of the team.

S11 – Development of a tool for automated tests: Testing, in general, can be repetitive and strenuous. This can lead to time wasted, burn-out, more probability of human error, and are limited to the capabilities of the team available, as well as the general costs of labor. There are certain tests, e.g., smoke tests or released tests, that have well-documented and repeatable steps. In these cases, a development of a tool that will conduct them, when necessary, will be helpful for the company. This solution aims to increase the scope and scale of testing, to reduce time wasted, costs of testing, and to reduce the likelihood of human error.

S12 – Incentive for individual ticket centralization: Although the process of using SCRUM, Kanban, or SCRUMBAN is widely recognized, successful, and useful, the fact that there are users working with each other on different time schedules, or users working on different teams might lead to confusion in ticket management by an individual. The aim is to take advantage of Jira platform and to develop a ‘personal area’ for each worker, where they have the information required only for what is concerning to them. This should allow the user to track ticket status, to filter tickets by several categories, to make changes to the tickets that will be translated in the main board and should have the user’s individual history easily accessible. This solution will allow for people working on different teams to have a centralized board where they can access everything, reducing time wasted, disperse information, and provide easier communication.

S13 – Establish documentation for ticket management: Establishing a certain standard for ticket management, like ticket creation guidelines (Definition of Ready, Definition of Done), who to assign tickets, how to prioritize it, how to document it, test it, and communication throughout the entire workflow. There should also be a section on uncertainties, like time estimations gone awry, new tickets added, sickness, vacations, among other factors. This solution aims to give a structure to the defined workflow and reduce communication barriers, improve certainty and direction in the process.

6.2.3. Distributed teams

S14 – Establishing specific goals, Objectives, and Key Results: Gauging work is not necessarily done via Key Performance Indicators (KPIs) or constant monitoring and measurement. As a team, there are certain goals that should be specified. There are also objectives and key results for a company and for a team, which should be the metric through which success is measured.

S15 – Introducing an initiative of projects focused on inclusion and team-building activities: Establishing an initiative of different social projects, with focus on team-building activities and the fostering of communication and interaction between team members. This solution aims to fight isolation for people who are in a virtual setting and to increase the union between teams and departments, building a stronger and cohesive company.

S16 – Betterment of the onboarding process: Although the onboarding process at this company has a solid foundation, it is still noted that there are some problems with the current configuration of it. The culture of the company is introduced to a new member in this stage of the organizational behavior. As such, it is vital that for a better alignment between company and worker that the onboarding process is improved.

S17 – Creating a standardized information page for concerns and solutions regarding a distributed company's environment: Entering a company with focus on multiple countries and continents might feel overwhelming. This solution intends on the creation of documentation and standards that should be followed globally by each member. This page can focus on multiple subjects, such as communication channels and expectations, working hours, composition of teams, and security concerns. This solution aims to condense all the necessary information into a findable and organized page, reducing cluttering and confusion of information, informing the workers in a proper manner, and establishing a standard of conduct.

S18 – Creating distributed environment training workshops: This solution focuses on creating and distributing workshops focused on training for the reality of a distributed environment in today's world. This type of workshop can include many tools and objectives, such as cultural awareness, tips for a better work strategy and environment, among others. This solution intends to reduce isolation, to improve communication and understanding between members from different backgrounds, and to improve union and interaction in a distributed company.

The relation between what cause (tables 1, 2 and 3, in chapter 5) each solution intends to solve or mitigate is translated by the table 5:

Table 5 - Relation between problems, causes and solutions

Sector	Problem	Cause	Solution
Sprint Planning/Data management	P1	C1.1	S1, S2, S3, S5, S14
		C1.2.	S4
	P2	C2.1.	S5, S8, S17
		C2.2.	S5, S6, S7, S8
	P3	C3.1.	S2, S7, S8
		C3.2.	S7, S8
		C3.3.	S7, S8
Ticket Management	P4	C4.1.	S9, S10
		C4.2.	S9, S10, S11
		C4.3.	S11
	P5	C5.1.	S9, S10
		C5.2.	S12, S13
	P6	C6.1.	S12, S13, S17, S18
		C6.2.	S12
Distributed teams	P7	C7.1.	S1, S14
		C7.2.	S14
		C7.3.	S15, S17, S18
	P8	C8.1.	S15, S16, S17, S18
		C8.2.	S15
		C8.3.	S16, S17, S18
	P9	C9.1.	S17, S18
		C9.2.	S18
		C9.3.	S17

These solutions were found with the chosen methodology for this project/company, but they were stated in a general frame. The next chapter will propose strategies of implementation of the initiatives, tools, and measures inside the company.

Key Takeaways of this chapter:

- The interviewed persons considered that each cause of the problems had different levels of severity and had different levels of frequency. This creates different priorities for the solving of each.

- Different solutions can be identified for the three main areas studied in the previous chapters. The identified solutions can impact and solve more than one cause of problems.

- RQ3 is answered in this chapter.

7. Implementation and Measurement of Solutions

The last chapter provided a theoretical framework of solutions to be implemented, providing the answer to RQ3 ("What are the solutions that could be implemented to avoid or mitigate those problems/causes?"), considering the causes and problems identified beforehand. To study the effectiveness of these solutions, one can utilize the company for the empirical study, discussing the implementation strategies and the metrics necessary to assess if each solution was adequate and provided benefits, defining a roadmap for the company in specific.

7.1. Cost-Benefit Analysis

Due to the limited time frame of the project and the internship, it is not feasible to implement every single solution presented. During the informal interviews, there were several discussions regarding the problems identified, the proposal of solutions and the evaluation of the impact/effort needed for implementation. A survey was also presented to interviewees (Appendix 12), with the results (Appendix 13) translated in figure 5:

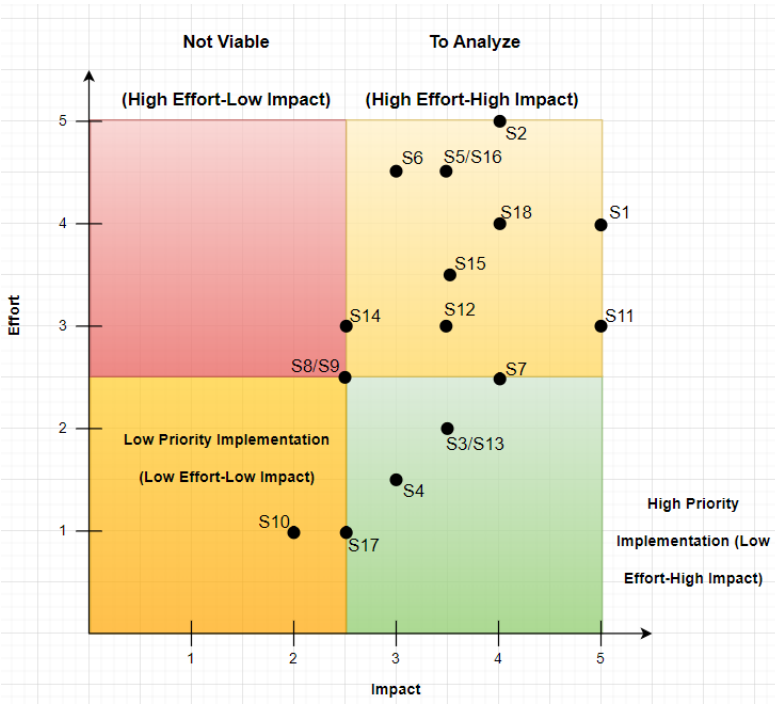


Table of Causes	
ID	Description
S1	User story mapping
S2	Protocol between client, product owner, team
S3	Meetings of alignment project roadmap - sprint planning
S4	Review of the daily meetings functioning
S5	Tool of data management
S6	Homepage for repository documentation
S7	Creation of a template document for requirement registry
S8	Backlog refinement meeting establishment
S9	Establish central display of environments
S10	Documentation of norms for environment management
S11	Development of a tool for automated tests
S12	Incentive for individual ticket centralization
S13	Establish documentation for ticket management
S14	Establishing specific goals, objectives, and key results
S15	Projects focused on inclusion and team-building activities
S16	Betterment of the onboarding process
S17	Standardized information page of distributed company environment
S18	Creating distributed environment training workshops

Figure 5 - Cost-Benefit Analysis of Solutions (with labels)

7.2. Implementation Score

Although the Cost-Benefit analysis is important and straightforward in deciding some of the solutions to be implemented, since a lower effort yields a higher impact (e.g. S3, S13), there are some solutions that, while impactful, are also costly, and/or time-consuming. A solution affects multiple causes of different problems. As such, an equation was designed to study the priority each solution should have in being implemented:

$$\mathrm{IS} = \frac{I * \mu S * \mu F}{E}$$

This was called Implementation Score (IS), with I being the impact estimated for the solution in cause, E being the effort that such solution will take, μS being the calculated average of the

severity of the causes affected by the solution and μF being the calculated average of the frequency of the causes of problems affected by the proposed solution. This implementation score had a relation of multiplication between the three factors based on the fact that if one of them is 0, the implementation score should automatically be zero, as no impact, severance, or frequency, meaning that either the cause to be solved has no effect on every day-to-day basis or the solution has no impact. Conversely, if a solution requires 0 effort it will have an "infinite" implementation score, which means it should be implemented immediately, as it is easy to deploy. In the end, the Implementation Score for the solutions is summarized in the following table:

Table 6 - Summary of the implementation scores of solutions

Solution	IS
S1 – User story Map	18,3
S2 – Protocol between client, product owner, team	13,5
S3 - Periodic meetings for alignment between the project roadmap and sprint planning	30,4
S4 - Review of the daily meetings functioning	24,0
S5 - Tool of data management	11,9
S6 – Homepage Confluence	9,1
S7 - Creation of a template document for requirement registry	16,0
S8 – Backlog Refinement meeting	11,1
S9 - Establish a central display of environments	4,0
S10 - Documentation of norms for environment management	8,0
S11 - Development of a tool for automated tests	13,5
S12 – Incentive for individual ticket centralization	6,2
S13 - Establish documentation for ticket management	7,9
S14 - Establishing specific goals, objectives, and key Results	9,6
S15 - Initiative of projects focused on inclusion and team-building activities	9,2
S16 – Betterment of onboarding process	7,8
S17 - Creating a standardized information page for concerns and solutions regarding a distributed company's environment	21,3
S18 - Creating distributed environment training workshops	9,2

Since there are plenty of solutions proposed, this metric can be used to assist in the decision of priority and urgency in the implementation.

Due to the limited timeframe of the project, not all the solutions could be implemented presently. The implementation score is a good metric on how to decide in priority for the implementations, in general, however, some of the solutions propositioned required some previous work done before being able to be put into practice. Some of this previous work also implied timeframes that were not supported by the project. As such, the following sub-topic will present the implemented solutions in the present and the strategies/notions of implementation/insertion for the solutions that are yet to be in place.

7.3. Sprint Planning and Data Management

Implemented at the moment:

S1 - One of the most frequent and severe problems identified in the study of problems was the lack of requirements or the inaccuracy of requirements made by clients, as well as a lack of connection between sprint planning and the overall long-term view of the project. To establish a better clarity and alignment between short-term plans and long-term plans, a user story mapping protocol was designed.

In it, following a set of steps (Appendix 14), a product owner and a client define a user story map, a tool that allows a visual display of everything regarding the project, from epics and stakeholders to a minimal visual detail added to complement a feature (Appendix 15). This iterative process will allow for everyone involved in the development of the project to gauge the steps the team has taken so far and where the project is headed, adding a fluency and connection to the overall project. This will also enable for an easier conversation between product owner and client regarding what is wanted or viewed as necessary.

The requirements will also be placed somewhere where they can be easily accessed, tracked, and documented. If an upcoming feature is blocked or is dependent on past features, a review and tracking of such will also be easier to identify and study. As with mostly everything in agile methodologies, this process is not stagnant. For effects of simplicity, the user story map shown in Appendix 15 is small, but not only will the user story roadmap of a project be bigger, but it will also be interchangeable. As a team follows the workflow defined in appendix 14, the roadmap can change, whether it be priorities of tasks, the timeline of the tasks, or the timeline/budget of the project.

It is important to have a constant and iterative process where the needs and wants of the client change, keeping the team and the short burst nature of sprints in connection with the long-term vision, however, more of that is guaranteed by the S2 and S3 solutions.

S2 - The S1 protocol is important for an overall alignment between the early stages of a project throughout plenty of phases until the project is reaching its maturity. However, it is prone to change and flexibility and is a general view of what, how, and when the things should be done. To complement the map and to ensure the synchrony and connection of the project between every moving part of the project, a protocol between the client, the product owner, and a senior member of the team (most likely a Senior Developer), where they have extended meetings (preferably in person) to understand the bigger tasks (epics) of the project, the requirements, the functional capacity of the team, the estimation of timeframes, and the adjustments necessary **(S2)**.

For this protocol to work, it is important that every participating member prepares beforehand for their duties, to be flexible and attentive, and to have proper understanding and documentation of the decisions made. This protocol is meant to be intensive and extensive at first, with lower intensity as the progress goes on (Appendix 16). This protocol is meant to provide accurate, realistic, attainable, and smart requirements.

S3 - After implementing S1 and S2, it's important to understand that the "User Story Map" is not rigid. Since we are dealing with agile methodologies and an everchanging environment, the necessities and requirements are prone to change. In fact, it's almost guaranteed that they are going to. People who are engaged in learning and staying on top of the project will keep tabs on the long-term project by themselves, however, it's not the case for everyone. To make sure that everyone is still connecting the short bursts of iterations (sprints) with the long-term vision of the project, periodic meetings will be implemented to go over the changes made during that period.

The periodicity of the meetings will be defined by the project managers, depending on the scope of the project, the size of the team, the volatility of the client, among others, however, it's not recommended to have the meeting more than once a month and less than once every two months.

S8 - The backlog refinement meeting is being implemented only in VHV now, with monthly meetings scheduled for the planned features team. There were over 60 tickets that were not being worked on, had no estimates, and had no assignees. Now, over 40 of those tickets were either scrapped, defined, or resolved, saving time, improving efficiency, and avoiding future confusion.

Implementation roadmap of the remaining solutions:

S4 - One of the problems most identified throughout interviews is that there was a feeling of unnecessary or too much time-consuming meetings. This happened because, often, people with little relevancy to the meeting were invited to it and rarely contributed. For example, if a meeting of 15 minutes has 3 people that are not contributing, but are obligated to be there, that's already 45 minutes of effective time wasted.

Agile methodologies are focused on flexibility and adaptation. As such, a suggested implementation is a 10-minute window at the end of each sprint planning where the daily meetings' functioning is reviewed and decided for the next two weeks. Which members are relevant and should be there every day, and which ones are called punctually, when necessary. The duration for each meeting is also established and what are the most important points to focus on each meeting, as well as the points of communication need to maintain a solid flow and understanding between participating and non-participating members.

S5 - At the moment of conclusion of this project, it was established that Confluence, the wiki tool for the company, as a platform is not developed enough to invest in a tool for documentation management for it. The compatibility issues found are causing a bigger effort

than anticipated and the implementation score would then suffer because of it, making it not viable or worth it for implementation.

The alternatives proposed in the future would be to change the documentation application to others with dedicated and integrated documentation management tools, or for the company to provide an extensive training in Confluence, focusing on its features, like spaces and pages, the collaborative editing and the available document templates that exist, as well as the possibility of searching and organizing the content. This session can also be helpful with the introduction of the next possible solution, the centralized page index (S6).

S6 - A problem frequently identified during the interview phase was the fact that information in the company's data repository was outdated and/or unorganized. People found it hard to find what they wanted for. As such, a revamp of the documentation is proposed. This alteration would take shape with a new, updated, relevant central page index.

Due to the limited timeframe of this case study and the efforts required for this solution, the implementation was not yet done. However, the steps to proceed with this implementation would be to replan the structure that the company intends to have with the documentation already present, considering the different teams, roles, and departments of the company. The index page would then be created with an intuitive visual and allowing users to access the relevant spaces.

The content would be well-organized, maintained and updated by the tool (S5) of data management, being able to be conserved and updated with ease. New members would be able to access and use information with less difficulty, members with more experience would be able to access and use information faster.

This solution brings consistency to the overall company, enhances the navigation and discovery potential for members, all while saving time and space.

S7 - At this moment, requirement gathering is done in an intuitive way. A conversation is had between the client and the Product Owner, and they decide in an informal manner what to register or proceed to do. As such, the requirement gathering is done in an open and improvisational communication matter. To avoid confusion, misinterpretations, and to facilitate the registry and gathering of requirements, as well as to give a common sense of understanding before the issue/ticket is even created, a template document for requirement registry was created (Appendix 17).

As shown, a client and a Product Owner will decide, and register, in the document, plenty of relevant factors that can then be attached to the ticket at hand (properties like User Stories, dependencies, constraints, etc.).

7.4. Ticket Management

Implemented at the moment:

S11 - In a software development company, especially in one that chooses to use agile methodologies, variations and little tweaks are in a daily fashion, sometimes hourly. Iterations and new features imply testing.

Constant manual testing is time-consuming, it is tiring, and it is also prone to human error. Besides the testing of the normal added features, at the end of every sprint, a release test is conducted. This implies the testing of all the products and all the features in the application. Most of these tests are replicable and have repeated steps. There is, then, the possibility of implementing a tool for testing automation. This tool is called SeTAK.

SeTAK (Selenium Test Automation Kit) is a tool to create and run Selenium based user interface tests. It was designed to support testers in writing and organizing test cases, in pure XML and without the need to use any programming language. Complex test suites can be structured in multiple ways and can be run against a web application from the command line, inside of Eclipse or by automated Maven builds, with the following workflow described in figure 6:

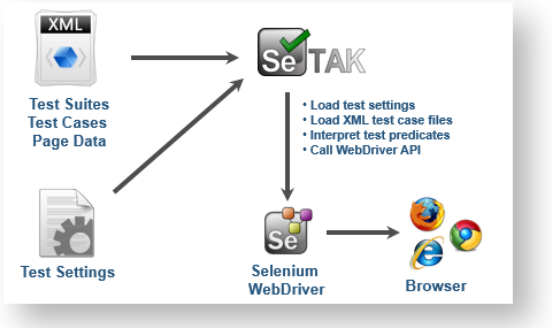


Figure 6 - Functioning of SeTaK

SeTAK can be used in every kind of repeatable testing, being especially indicated for release and smoke testing. More information about the metrics measured and the improvements made in the testing processes in the "Results" chapter.

S9 - Workers might have some difficulty in being synchronized with several environments. To facilitate their job of consulting specifications and state of each, a centralized page was created with several different properties, such as version, technical details, observations, and configurations (appendix 18). This is meant to help visualization, consultation, and updating of environments in a centralized display.

S13 - Although the company already has some fundamental understanding and implementation of ticket management, such as standardized templates imbued in JIRA with some required fields (due date, description, assignee, etc.), the definition of a ticket workflow (Appendix 14), notions of estimation and acceptance criteria, as well as dependencies and relationships, there are still some other features that can be implemented in the process, namely a strong and standard definition of done, a strong and standard definition of ready, and a workflow to easily identify the priority that should be given to tickets, solutions that were prepared and implemented as can be seen in Appendix 20.

Implementation roadmap of the remaining solutions:

S10 - The existence of multiple environments can bring, as stated before, confusion, lack of synchrony, too much effort for little results, among other negative side effects. The implementation of a standard for environment management is then being devised as a solution.

In this standard, naming guidelines for environments would be established, general concerns and recommendations would be stated. A redirection for the central display (S9) would also be implemented, to increase coherence. This standard allows for workers to be on the same page regarding the environments and their management, causing more efficiency and productivity with less work.

S12 - For people that are working on more than one project, keeping the organization and the flow of work might be more difficult, as there are more concerns regarding capacity and estimation, priority of tickets, and filtering of issues. JIRA offers the ability to users to create a centralized board where you can combine different projects in a way that suits what the worker needs. The board (which can be KANBAN or SCRUM) can then be customized with different columns, filters, and other properties. A workflow is proposed for a worker participating in multiple projects (Appendix 19), allowing him to save time with a better visualization of his workload. This, in combination with the standard project boards, permits an overall top to bottom and uncluttered view for the company member.

7.5. Distributed Teams

Implemented at the moment:

S14 - Implementing a framework of identifying Objectives and Key Results (OKRs) is a solid measure to align the company, the teams inside the company, and the workers in understanding and synchrony. These metrics measure progress and success, that is, the results achieved in comparison to those that were expected.

This framework consists in five key steps, as shown in appendix 22: First, the establishment of the company's key objectives, which are long-term and high-scale goals of the company. They are focused on qualitative definitions and should be conductive of the vision of the company. Second, the key results are measurable metrics (quantitative fashion) with a set timeframe.

These are meant to show the level of progress the company is making in achieving their objectives. The same process is done for the teams, with statements and metrics in a short to medium term. It is crucial that the OKRs of the teams are aligned with the OKRs of the company. Finally, and since this is meant to be an iterative process with constant evaluation, periodic meetings for refinement of the OKRs should be conducted.

S15 - People tend to have a higher interest in what they are doing if they feel invested and included in the project and in their team. With that in mind, a protocol of initiatives for team building and inclusion was developed. In it, there are several measures suggested for better connection between teammates:

1. In a software development context, problem-solving and quick analysis is important. Knowing how to do so in tandem with your co-workers is a valuable skill. Puzzles, escape games/rooms, team challenges and such are good ways of developing those skills. This can be done in a remote setting or physical setting, to accommodate for every type of worker.
2. Virtual coffee breaks: In a physical setting, it is often that workers have a natural coffee break where they relax, hang out, and talk about different subjects unrelated to their

work. In a distributed environment, the implementation of weekly windows for a coffee break is a way to combat the distance. The coffee breaks are not mandatory and would not have a structure.

3. Team lunches: Whenever there is an office day, where most of the team is going to be in the same physical space, encourage the members to have lunch together, either in the office or off-location.
4. Individual discussion sessions: Every month, a worker is paired with a teammate at random. After the pairing, they would have a couple of meetings where they could discuss their role in the team, what is their opinion on the project overall and what are some struggles and strong points they have faced that month. This allows team members to connect with a person they might not have talked much with before, can lead to an identification of problems and strong points in the knowledge that they each possess, and allows for an alignment between individual and team. This can be a controversial measure for some, so, it could be important to establish some guidelines, like the frequency and duration of the meetings, an anonymous environment for feedback and identified challenges, and the continuous gathering of feedback on the initiative.

S16 - Although the onboarding process of the company has some solid grounds, the fact that the buddies have no adjustment to their schedule and are expected to maintain the same work rate while guiding someone with little experience in the company's technical aspects might be problematic.

As for a solution to improve the onboarding process and engage the new workers more, the buddies should have a reduction of capacity hours allocated to others project, hours that should then be used to focus on the training, development of skills, and meshing of the new worker with the team and project. The buddy system is important, but the new member should be introduced to more teambuilding events, as well as more regular feedbacks and check-ins.

A closer relationship between buddy, team, and worker will lead to a better connection and results, all while reducing distractions, isolations, or growing pains.

S17 - One of the solutions that takes less effort to implement and could be done almost immediately was the introduction of a centralized page of frequently asked questions, frequently met concerns, and solutions.

Being a company focused on healthcare, it deals with privileged and delicate information. Information like this is always important to maintain under control, however, under the distributed team nature of the workers, it can be more difficult to ensure it. So, there was also a page displaying the security measures in place to guarantee a secure and compliant management of information (Appendix 21).

<u>Implementation roadmap of the remaining solutions:</u>

S18 - Building awareness on how to better work together in a distributed environment is key for success. In that sense, and if the timeframe of the project was more extended, the idea would be to implement several workshops focused on the realities and challenges of working remotely and/or with different cultures/time zone. The different workshops would consist of multiple

themes, such as communication, productivity from home, dealing with isolation, distractions, and insecurities, work-life balance, among others.

There would be a mix of virtual workshops and hands-on workshops, to make them compatible with the different modalities of work. People from inside the company and invited speakers would be able to share their experiences, gather around in a like-minded mindset and discuss good practices for better results as workers and teammates. There would be continuous feedback, and this would be an iterative process with the constant betterment of this initiative.

7.6. Measurement of Solutions

When proposing solutions in the form of tools or initiatives, it is important to establish some form of measurement. This serves multiple purposes, allowing for the evaluation of the impact of each solution implied. It allows for monitoring and refinement of the solutions, and compares the solutions implemented with the past realities. Although the metrics that were identified tried to be specific and measurable, the fact that some of the solutions are dealing with more social/ emotional factors implies that the measurement will be subjective and people-based, meaning that the measurement and metrics had some form of quantitative and/or qualitative nature (Appendix 23).

At the point of completion of this dissertation, there are only measurable results for S8, the backlog refinement meeting, and S11, the automated test tool. For S8, over 66% of the backlog was "cleaned" and improved on, reducing cluttering and outdated ticketing. For S11, the quantitative results show that a manual regression test on a single product takes 7-8 minutes manually and the tool achieves it in 3 to 4 minutes, achieving the same results in half the time. A release test in the VHV team took 5 to 6 hours depending on complications and complexities of testing, while it took the tool 2 to 2 hours and a half to achieve the same result.

As the project progresses and the implementations are observed and tracked, the measurements described in Appendix 23 will be checked and the solutions will be reevaluated and refined.

Key Takeaways of this chapter:

- There are different cost-benefit ratios for each solution identified.

- When one has a limited timeframe, one can implement an "Implementation Score" metric to help the process of prioritizing solutions to implement.

- Due to the limited timeframe of the project, the solutions that were not able to be implemented had a suggested implementation roadmap.

- One can define measurements (qualitative or quantitative) for each solution proposed.

- RQ4 is answered in this chapter.

8. The Decision Diagram

Chapter 7 answered the fourth research question (RQ4: "What are the strategies of implementation of the identified solutions?"), structuring the remainder of the project in the sense of observing and measuring the empirical effects of the proposed solutions in mitigating or outright eliminating the problems/causes that were stated in chapter 5.

However, the gap presented in the decision-making process of a project manager (RQ5: "How can a Project Manager decide on team and project management implementations and strategies?") is still not answered. A project manager in a software development company still must decide for himself what he views as the best possible management strategy.

This can be difficult, especially in a company with varied teams with multiple typologies. An absence of framework might generate lack of consistency, lack of communication and risk management, as well as increased difficulty to evaluate the effectiveness of management decisions. Stakeholders might perceive the decision-making as arbitrary or opaque, leading to doubt or dissatisfaction.

Throughout this case study, there were several factors that were expressed multiple times regarding their weight in management, be it as a possible advantage or cause of issues. To decide how to answer the last research question, the mind map of figure 7 analyzed the previous steps utilized in the report and identified the final steps necessary was to devise this framework of decision-making:

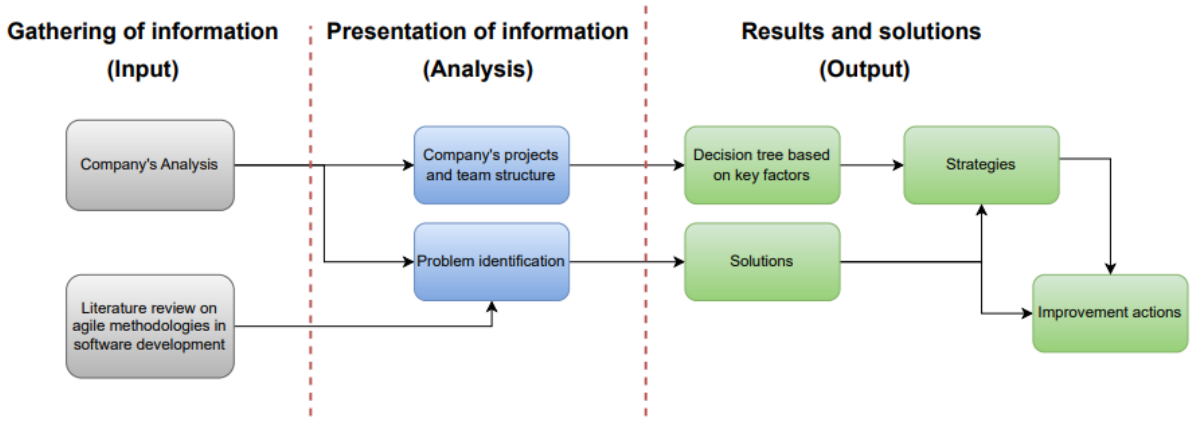


Figure 7 - Mind map for the creation and connection of the diagram with the remaining report

Utilizing them as guidelines for a better project management decision-making, a decision support system for project managers was constructed, called the Assistance to Project Managers Decision-Making (APMDM) diagram, present in figure 8:

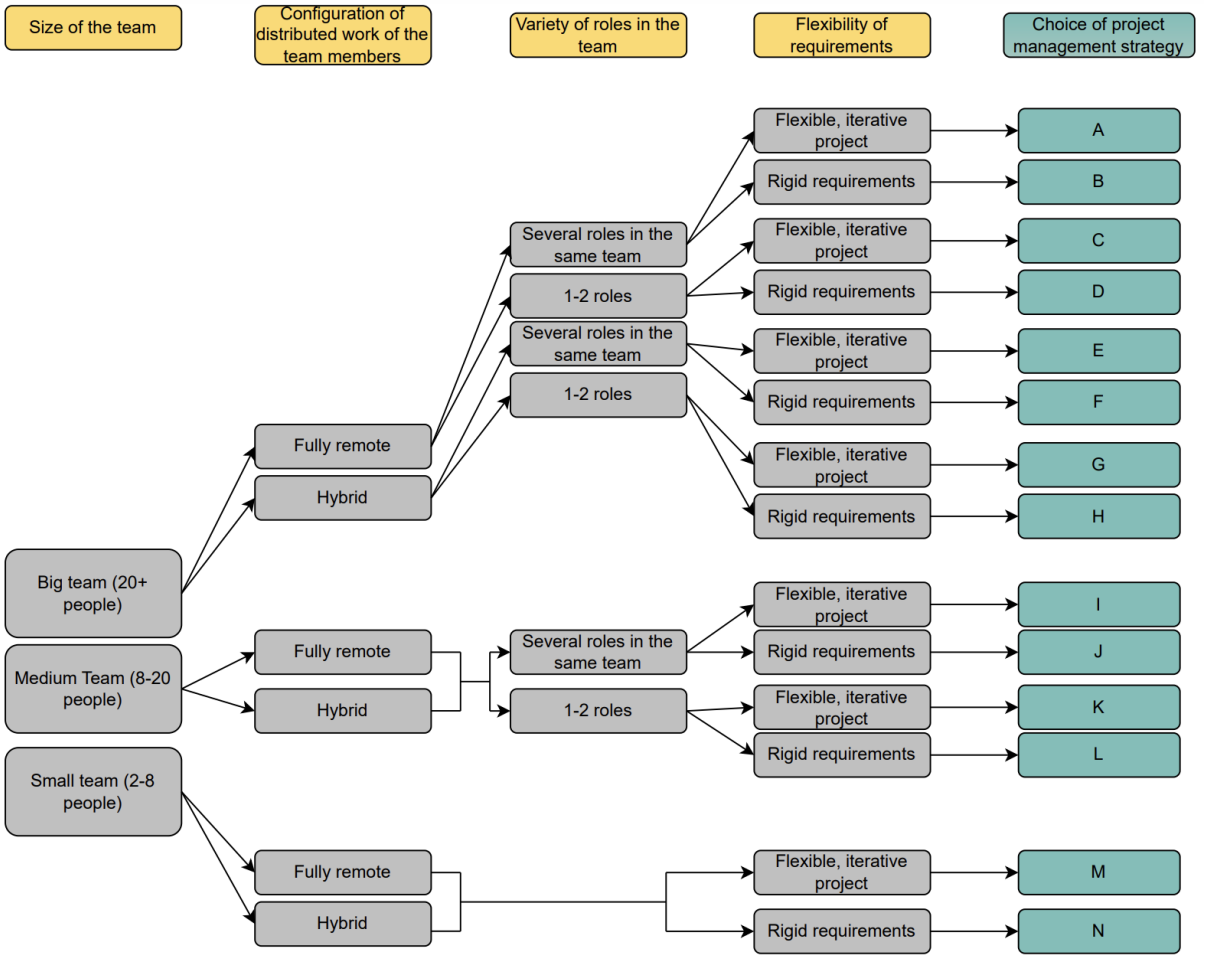


Figure 8 - The APMDM Diagram

As it can be seen, a project manager will follow the factors of major influence in projects of software development (size of the team, the type of distributed work, the variety of roles in team, and the flexibility of the requirements in the project) to arrive at several project management strategies, ranging from A-N (fully described in appendixes 24 through 37). These factors were chosen out of the information gathered from all the phases of the methodology.

This decision support system can be very useful for project managers, as it will display a sense of transparency and consistency, as well as provide a better quality of decisions, supported by a structured framework. The project managers will personally recognize their decisions with the rigor that they expect to maintain. Their peers will have more trust in the project manager's decisions, with results improving in smaller timeframes. In case of errors, the cause and possible solution will be easier to identify and work on, with each phase of decision outlined. There will be no shame or guilt in having committed a mistake, as it can be worked on and solved quickly.

It is also important to state that, like with all the other solutions and artefacts presented in this dissertation, this is not an inflexible framework. There are certain cases where, for other external factors not considered in the APMDM, this decision support system is not applicable or not appropriate. Even when applicable, it might get subdued to changes or adaptations. Flexibility, common sense, and the ability to insert increments and better iterations is key for employing a solid framework for improving quality management in a case-by-case basis.

Going back to the DSR process described in chapter 4, there are still two phases left to employ empirically: The demonstration and the evaluation. Artifacts created by DSR should be evaluated and validated by relevant stakeholders, being showcased by the creator of the artifact. To achieve this purpose, a demonstration workshop was conducted with 5 people inside of the company (An Agile Coach, a Product Owner, the Regional Manager, a Tech Lead, and a Delivery Manager), where the APMDM diagram was explained and demonstrated. Afterwards, a period of questions and answers (Q&A) was proposed, where stakeholders tried to better understand the artifact and remove any doubts. This period was then finished with general feedback from the stakeholders regarding the validity of the diagram, its application in real world and the possibility of implementation in the company.

All the stakeholders found the APMDM diagram interesting and relevant to the studied field and to the company, studying the possibility of implementation on certain teams (what would be, in a longer timeframe, the evaluation phase of the DSR methodology), and, eventually, in general. Although they agreed on the factors and believed that this was the correct selection of them, they pointed out that, in other companies, some other factors or an expansion on the already presented ones could happen, e.g., although this company has no teams working fully locally, there are companies that have teams that work only in-office settings.

Another possible factor identified in the feedback that could be an input for a diagram in another company would be the seniority of the team. This can be divided in two types: the seniority of the team members (plenty of experience or little experience) and the maturity of a team (new team, established team, reforming team, etc.), which can impact decision-making.

As the company is mostly focused on teams that have mixed seniorities and are, in general, in the “established team” phase, the diagram having this input does not provide a significant difference in a project manager’s decision process. This demonstration phase was important, as it evaluates and validates a theoretical artifact with real life experience and user feedback, allowing for a refinement and improving stakeholder involvement and engagement in the work conducted.

By agreeing on the validity and potential of the APMDM diagram, the stakeholders brought forth a strong possibility of practical relevance and effectiveness of the artifact, acknowledging that the company’s objectives were aligned with the research and that there is value brought to the target audience or to the realm of the problem.

The APMDM diagram still has to be evaluated, and the implementation of it in new teams is currently being studied in msg insur:it Iberia. For a proper evaluation, metrics had to be defined of success regarding the different factors in the strategies (Appendix 38). Given a solid timeframe, a relative analysis of the implementation of the diagram can be done by acquiring and assessing the results.

The APMDM diagram provides an answer to RQ5, as it gives project managers an established and constant framework of decision of implementation of strategies for a swift management of teams, in view of multiple different weighted factors.

9. Conclusion and Future Research

This project was focused on conducting research on the agile project management of a global software development company, intending to identify the main areas where one could find and try to solve problems. The main areas of relevance for this process were the sprint planning/data management/backlog management, the ticket management stage, and the overall state of the distributed teams in the context of software development.

This chapter will present the main conclusions drawn from the project, demonstrating the relation between the research objectives established in the first chapter and the output achieved at the end. There will also be presented possibilities of improvement or new research that may arise after this project.

9.1. Main conclusions

Overall, the research objectives that were defined in the first chapter were accomplished and all the set-out initial research questions were answered.

The process began by an extensive characterization and study of the processes that were involved in a global software development company, trying to determine the current state of such. The large number of projects and influx of new members in the studied company allowed for a possible identification of key similarities and differences between the different teams, and the way they handled the overall processes.

The objectives of the company were the analysis of processes and to obtain suggestions of changes, developments, or restructuring necessary to increase efficiency, productivity, connection between team and project and reducing the time to market of an emerging product, while reducing development times in general for current products/projects. The company was focused on Agile methodologies, opting mostly for SCRUM and KANBAN tools.

The first steps in the methodology allowed this project to answer the first research question (RQ1): “What are the problems affecting software development global companies in their processes?”, simultaneously answering the second research question (RQ2): “What are the causes of the identified problems?”. These answers came in the identification of problems in three main areas of the company’s functioning: the sprint planning and data/backlog management, the ticket management state of the company, and the overall conditions attributed to the distributed team configuration.

After identifying the problems and the causes, the project followed the natural course of trying to identify solutions and strategies of implementation for those identified, attempting to answer the RQ3 (“What are the solutions that could be implemented to avoid or mitigate those problems/causes?”) and RQ4 (“What are the strategies of implementation of the identified solutions?”). For that to happen, there was first an analysis of the severity and frequency of the problems/causes of problems in the functioning of the company, aided by interviews and surveys conducted to several relevant stakeholders in the company. This analysis brought forth several solutions, providing the answer to RQ3.

The theory of the identified solutions was then put to practice, with the implementation of 11 out of 18 solutions (~61%). The remaining are still on the implementation roadmap phase, either due to the impact, effort, cost associated with the solution, or because of the limited timeframe of this project. Therein lies the answer to RQ4, and metrics of success were defined to gauge

the efficacy of the implemented solutions and of the future ones to implement. There was also the implementation of a metric that allowed for the relevant stakeholders to prioritize solutions given limitations, called the implementation score (IS), which related the severity and frequency of problems and the effort and impact of installing a solution.

Finally, this project generated an artifact in the form of a decision support system, consisting of relevant factors present in teams as the variables of decision, such as the size of the team, if teams worked on hybrid or total remote setting, if they had multiple roles in the team, and if the project was flexible. This decision tree was meant to provide a structured framework and a practical approach to decision-making for project managers, allowing them to have a bigger sense of transparency and consistency, as well as to provide a better quality of decisions.

This artifact is meant to improve reliability and traceability. This diagram, called Assistance to Project Managers Decision-Making (or APMDM), is the encompassing of the whole project, with the strategies employed being possible by the previous analysis of the processes of the company and the implementation of some of the solutions discovered and suggested. This artifact answers the fifth research question (RQ5): "How can a Project Manager decide on team and project management implementations and strategies?".

This artifact was presented to relevant stakeholders in the company, to understand and validate its functionality and usability. The feedback from this demonstration, in the form of a workshop, was positive, with statements of confidence in the practical application of the decision diagram in the real world and a show of interest in the empirical establishment of it in the company.

The openness of the relevant stakeholders and the constant participation and rigor of both the researcher and the participants requested for this work were essential for a well-aligned and performed research work, following the steps established *a priori* and accomplishing the objectives stated.

Concluding, and in sum, this project provided results/answers in four different outcomes:

1) Suggestion of appropriate solutions for the identified problems.
2) Establishment of factors of importance in problem and solution characterization and definition of a metric (IS) for prioritizing implementation of solutions.
3) Establishing an implementation roadmap for each solution, and metrics of success.
4) The decision support system/decision tree for project managers.

This project intended to fill the literature research gaps identified in the early chapters, providing an advance and contribution to the previously established literature, and inviting the future research and development of investigation by companies, workers, and scholars.

9.2. Limitations

Although the project was able to fulfill its research objectives and provide an answer to the research questions, there are always some hurdles that are encountered throughout or retroactively identified in the research project.

One of the main limitations to be identified is the generalization of the findings. Since the methodology was mostly focused on the case study of the company, without studying other samples and realities, it is difficult to assert that the inferences can be generalized.

Although the methodology was consisted of mixed tools (qualitative and quantitative), the main basis of interviews posits some subjectivity to the findings. There is a necessity of the critical

analysis of the context of the company and of the moment in time. Opinions, answers, and identifications might change with time. People might also not feel comfortable expressing negative opinions about their place of work, even if anonymity was insured and safety of protocol established, which can generate some biases.

The timeframe of the project limited the implementation of solutions, the usefulness of the implementation score, or the extensive study of the APMDM diagram, which means that, at this moment, it is not yet possible to establish the diagram as field-tested.

9.3. Future Research

As stated in the sub-chapter above, the timeframe available for the execution of this project made the implementation of every solution proposed unfeasible. Although more than half of them were/are implemented, not all of them were able to be measured properly and assessed. We can, then, divide the possibility of future research in two ways: Internal, and external.

Internally, the company can follow through with three perspectives: The first one, by measuring and assessing the current solutions in place. This project management optimization approach, especially in an agile setting, is an iterative process, so, it is important that the solutions that were once right/appropriate continue to being validated, and refined, if necessary. The APMDM diagram can be useful if proved to work, which should be validated against the metrics defined.

Secondly, the solutions that were not yet implemented should follow the implementation roadmap, with careful consideration for the steps designed. If faced with constraints, it would be good practice to analyze the IS and deciding on that basis what and when to implement certain solutions.

Third, the company can use this project as a steppingstone for further study in good project management practices and strategies, propelling the investigation and research to higher levels, attempting to achieve better and faster results. This latter perspective is currently being deepened, as the company is also studying the possibility of establishing the methodology of interview and survey approach to assess the frequency and severity of problems encountered and perceived impact and effort of solutions.

Externally, the future research is based on validating the conclusions of the methodology and the outcomes presented more generally. This project followed a DSR approach with the use of mixed-methods tools with a bigger focus on qualitative data, due to the limitation of data available and the timeframe of the project. The conclusions and solutions here presented are a good jumpstart for a scholar or worker studying good practices of agile management to apply to other companies or case studies, permitting the analysis of the possible generalization of the strategies, tools, initiatives, measures, and artifacts created in here to other companies in the sector. The fact that the APMDM artifact was created, evaluated, discussed, and refined in a specific setting proves that it could be applied to the company, but researchers can attempt to implement it in other environments and test the generality of the created artifact, while also validating it against the proper contextual metrics.

This analysis, especially if backed with quantitative data, can be a strong step in the direction of applying the established frameworks and solutions to other companies of the similar nature, creating a new standardized and refined service design for companies in the software development area.

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile Software Development Methods: Review and Analysis*.

Almalki, S. (2016). Integrating Quantitative and Qualitative Data in Mixed Methods Research—Challenges and Benefits. *Journal of Education and Learning*, *5*(3), 288. https://doi.org/10.5539/jel.v5n3p288

Almeida, F. (2017). Challenges in Migration from Waterfall to Agile Environments. *World Journal of Computer Application and Technology*, *5*(3), 39–49. https://doi.org/10.13189/wjcat.2017.050302

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (IJIM)*, *14*(11), 246. https://doi.org/10.3991/ijim.v14i11.13269

Ardichvili, A., Maurer, M., Li, W., Wentling, T., & Stuedemann, R. (2006). Cultural influences on knowledge sharing through online communities of practice. *Journal of Knowledge Management*, *10*(1), 94–107. https://doi.org/10.1108/13673270610650139

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management. *Communications of the ACM*, *48*(12), 85–89. https://doi.org/10.1145/1101779.1101781

Barros, M. de O., Werner, C. M. L., & Travassos, G. H. (2004). Supporting risks in software project management. *Journal of Systems and Software*, *70*(1–2), 21–35. https://doi.org/10.1016/S0164-1212(02)00155-3

Bell, B. S., & Kozlowski, S. W. J. (2002). A Typology of Virtual Teams. *Group & Organization Management*, *27*(1), 14–49. https://doi.org/10.1177/1059601102027001003

Benediktsson, O., Dalcher, D., & Thorbergsson, H. (2006). Comparison of software development life cycles: a multiproject experiment. *IEE Proceedings - Software*, *153*(3), 87. https://doi.org/10.1049/ip-sen:20050061

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64–69. https://doi.org/10.1109/2.976920

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61–72. https://doi.org/10.1109/2.59

Breu, K., & Hemingway, C. J. (2004). Making Organisations Virtual: The Hidden Cost of Distributed Teams. *Journal of Information Technology*, *19*(3), 191–202. https://doi.org/10.1057/palgrave.jit.2000018

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International Digital Library Perspectives*, *27*(1), 18–22. https://doi.org/10.1108/10650751111106528

Chinowsky, P. S., & Rojas, E. M. (2003). Virtual Teams: Guide to Successful Implementation. *Journal of Management in Engineering*, *19*(3), 98–106. https://doi.org/10.1061/(ASCE)0742-597X(2003)19:3(98)

Cho, J. J. (2010). *An exploratory study on issues and challenges of agile software development with scrum.* Utah State University.

Cohen, D., Lindvall, M., & Costa, P. (2003). *Agile software development*.

Connaughton, S. L., & Shuffler, M. (2007). Multinational and Multicultural Distributed Teams. *Small Group Research*, *38*(3), 387–412. https://doi.org/10.1177/1046496407301970

Coram, M., & Bohner, S. (2005). The Impact of Agile Methods on Software Project Management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, 363–370. https://doi.org/10.1109/ECBS.2005.68

Corbin, J., & Strauss, A. (2008). *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, Inc. https://doi.org/10.4135/9781452230153

Creswell, J. W. (1994). *Research design: Qualitative & quantitative approaches.* Sage Publications, Inc.

Creswell, J. W. (2003). *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE.

Creswell, J. W., & Clark, V. L. P. (2011). Best practices for mixed methods research in the health sciences. *National Institutes of Health*.

Crisp, C. B., & Jarvenpaa, S. L. (2013). Swift Trust in Global Virtual Teams. *Journal of Personnel Psychology*, *12*(1), 45–56. https://doi.org/10.1027/1866-5888/a000075

Cunha, M. P. e, & Gomes, J. F. S. (2003). Order and Disorder in Product Innovation Models. *Creativity and Innovation Management*, *12*(3), 174–187. https://doi.org/10.1111/1467-8691.00280

da Silva, F. Q. B., Costa, C., Franca, A. C. C., & Prikladinicki, R. (2010). Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review. *2010 5th IEEE International Conference on Global Software Engineering*, 87–96. https://doi.org/10.1109/ICGSE.2010.18

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical Education*, *40*(4), 314–321. https://doi.org/10.1111/j.1365-2929.2006.02418.x

Dulebohn, J. H., & Hoch, J. E. (2017). Virtual teams in organizations. *Human Resource Management Review*, *27*(4), 569–574. https://doi.org/10.1016/j.hrmr.2016.12.004

Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems*, *24*(1), 135–169. https://doi.org/10.2753/MIS0742-1222240104

Estler, H.-C., Nordio, M., Furia, C. A., Meyer, B., & Schneider, J. (2014). Agile vs. structured distributed software development: A case study. *Empirical Software Engineering*, *19*(5), 1197–1224. https://doi.org/10.1007/s10664-013-9271-y

Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management—agilism versus traditional approaches. *Journal of Computer Information Systems*.

Ford, R. C., Piccolo, R. F., & Ford, L. R. (2017). Strategies for building effective virtual teams: Trust is key. *Business Horizons*, *60*(1), 25–34. https://doi.org/10.1016/j.bushor.2016.08.009

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 28–35.

Gagné, M., & Deci, E. L. (2005). Self-determination theory and work motivation. *Journal of Organizational Behavior*, *26*(4), 331–362. https://doi.org/10.1002/job.322

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Gunduz, M., & Almuajebh, M. (2020). Critical Success Factors for Sustainable Construction Project Management. *Sustainability*, *12*(5), 1990. https://doi.org/10.3390/su12051990

Halcomb, E., & Hickman, L. (2015). Mixed methods research. *Nursing Standard*, *29*(32), 41–47. https://doi.org/10.7748/ns.29.32.41.e8858

Herbsleb, J. D., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, *29*(6), 481–494. https://doi.org/10.1109/TSE.2003.1205177

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, *18*(2), 16–20. https://doi.org/10.1109/52.914732

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2).

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

Hinds, P. J., & Bailey, D. E. (2003). Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams. *Organization Science*, *14*(6), 615–632. https://doi.org/10.1287/orsc.14.6.615.24872

Hinds, P. J., & Mortensen, M. (2005). Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science*, *16*(3), 290–307. https://doi.org/10.1287/orsc.1050.0122

Hoch, J. E., & Kozlowski, S. W. J. (2014). Leading virtual teams: Hierarchical leadership, structural supports, and shared team leadership. *Journal of Applied Psychology*, *99*(3), 390–403. https://doi.org/10.1037/a0030264

Hood, J. C. (2006). Teaching Against the Text: The Case of Qualitative Methods. *Teaching Sociology*, *34*(3), 207–223. https://doi.org/10.1177/0092055X0603400301

Hsu, M.-H., Ju, T. L., Yen, C.-H., & Chang, C.-M. (2007). Knowledge sharing behavior in virtual communities: The relationship between trust, self-efficacy, and outcome expectations. *International Journal of Human-Computer Studies*, *65*(2), 153–169. https://doi.org/10.1016/j.ijhcs.2006.09.003

Jiménez, M., Piattini, M., & Vizcaíno, A. (2009). Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in Software Engineering*, *2009*, 1–14. https://doi.org/10.1155/2009/710971

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a Definition of Mixed Methods Research. *Journal of Mixed Methods Research*, *1*(2), 112–133. https://doi.org/10.1177/1558689806298224

Kothari, C. R. (2004). *Research Methodology: Methods and Techniques* (Second). New Age International.

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489–504. https://doi.org/10.1057/ejis.2008.40

Laporte, C. Y., & April, A. (2017). *Software Quality Assurance*. Wiley.

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, *36*(6), 47–56. https://doi.org/10.1109/MC.2003.1204375

Leedy, P. D., & Ormrod, J. E. (2001). Practical Research: Planning and Design. Seventh Edition. *Education Review*.

Liao, C. (2017). Leadership in virtual teams: A multilevel perspective. *Human Resource Management Review*, *27*(4), 648–659. https://doi.org/10.1016/j.hrmr.2016.12.010

Lilian, S. C. (2014). Virtual Teams: Opportunities and Challenges for e-Leaders. *Procedia - Social and Behavioral Sciences*, *110*, 1251–1261. https://doi.org/10.1016/j.sbspro.2013.12.972

Long, H. (2014). An Empirical Review of Research Methodologies and Methods in Creativity Studies (2003–2012). *Creativity Research Journal*, *26*(4), 427–438. https://doi.org/10.1080/10400419.2014.961781

Lopez-Martinez, J., Juarez-Ramirez, R., Huertas, C., Jimenez, S., & Guerra-Garcia, C. (2016). Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review. *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 141–148. https://doi.org/10.1109/CONISOFT.2016.30

Lycett, M., Macredie, R. D., Patel, C., & Paul, R. J. (2003). Migrating agile methods to standardized development practice. *Computer*, *36*(6), 79–85. https://doi.org/10.1109/MC.2003.1204379

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Marlow, S. L., Lacerenza, C. N., & Salas, E. (2017). Communication in virtual teams: a conceptual framework and research agenda. *Human Resource Management Review*, *27*(4), 575–589. https://doi.org/10.1016/j.hrmr.2016.12.005

Merriam, S. B. (2002). *Merriam, S. B. (2002). Qualitative research in practice : examples for discussion and analysis. In Jossey-Bass eBooks. http://tweb.cjcu.edu.tw/journal/2013_03_05_03_27_38.63.pdf*. Jossey-Bass.

Morrison-Smith, S., & Ruiz, J. (2020). Challenges and barriers in virtual teams: a literature review. *SN Applied Sciences*, *2*(6), 1096. https://doi.org/10.1007/s42452-020-2801-5

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

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, 308–313. https://doi.org/10.1109/ENABL.2003.1231428

Pavlou, Liang, & Xue. (2007). Understanding and Mitigating Uncertainty in Online Exchange Relationships: A Principal-Agent Perspective. *MIS Quarterly*, *31*(1), 105. https://doi.org/10.2307/25148783

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for Design Science Research Evaluation. *European Conference on Information Systems (ECIS)*.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, 1–9.

Saunders, C. S., & Ahuja, M. K. (2006). Are All Distributed Teams the Same? Differentiating Between Temporary and Ongoing Distributed Teams. *Small Group Research*, *37*(6), 662–700. https://doi.org/10.1177/1046496406294323

Schmid, B., & Adams, J. (2008). Motivation in Project Management: The Project Manager's Perspective. *Project Management Journal*, *39*(2), 60–71. https://doi.org/10.1002/pmj.20042

Schwaber, K., & Sutherland, J. (2011). *The Scrum Guide*.

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). SCRUM model for agile methodology. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 864–869. https://doi.org/10.1109/CCAA.2017.8229928

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Journal of Product Innovation Management*, *3*(3), 205–206. https://doi.org/10.1016/0737-6782(86)90053-6

Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*, *181*, 746–756. https://doi.org/10.1016/j.procs.2021.01.227

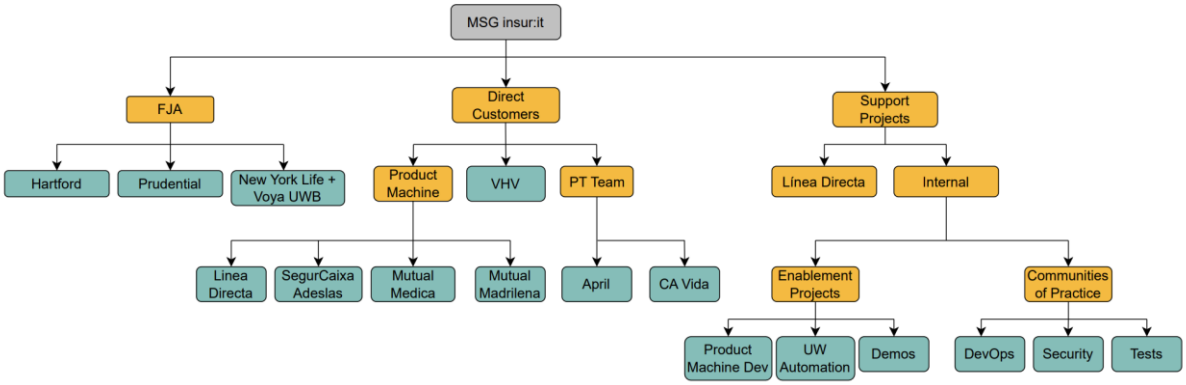
Walkinshaw, N. (2017). *Software Quality Assurance*. Springer International Publishing. https://doi.org/10.1007/978-3-319-64822-4

Watson, R. (2015). Quantitative Research. *Journal of Advanced Nursing*.

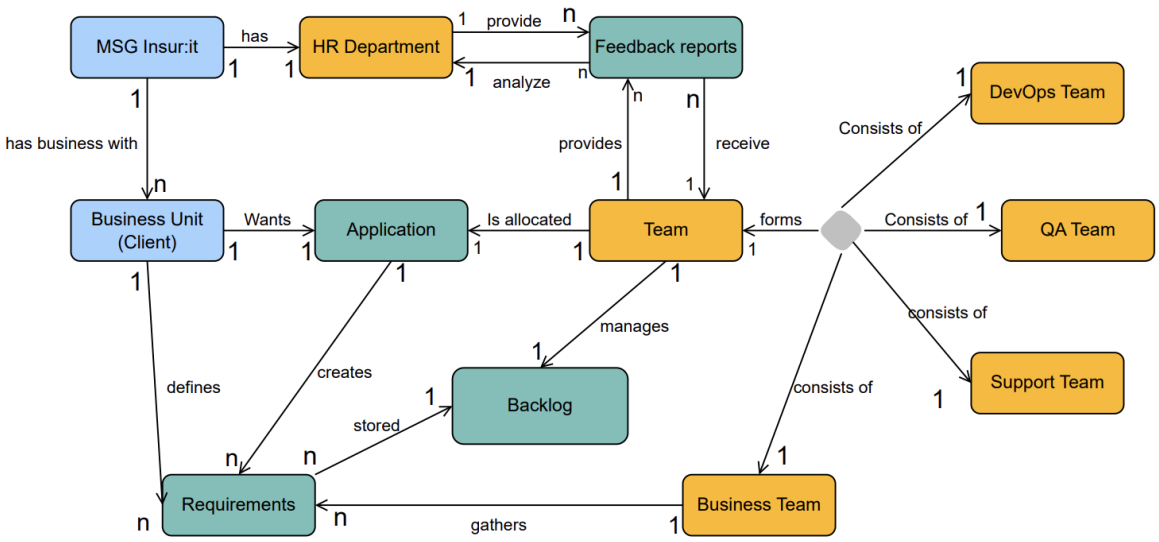
Williams, C. (2007). Research Methods. *Journal of Business & Economic Research*, *5*(3).

Winter, R. (2008). Design science research in Europe. *European Journal of Information Systems*, *17*(5), 470–475. https://doi.org/10.1057/ejis.2008.44

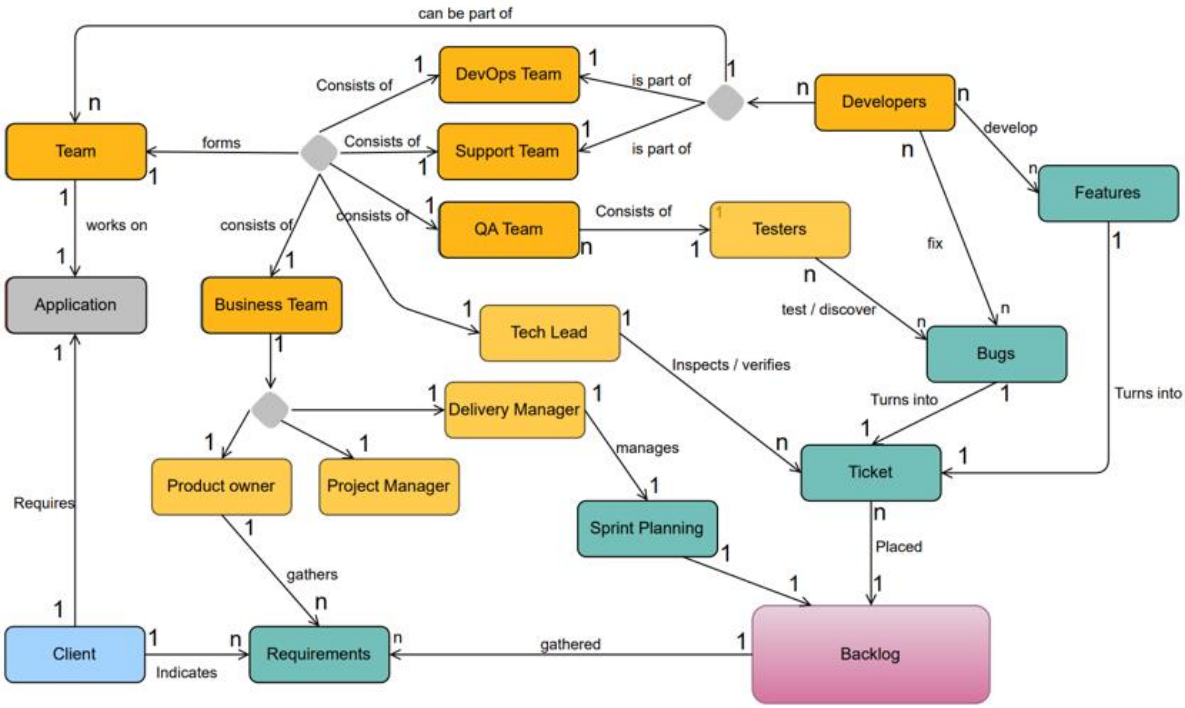
Appendix 1: Company's Teams Organizational Chart



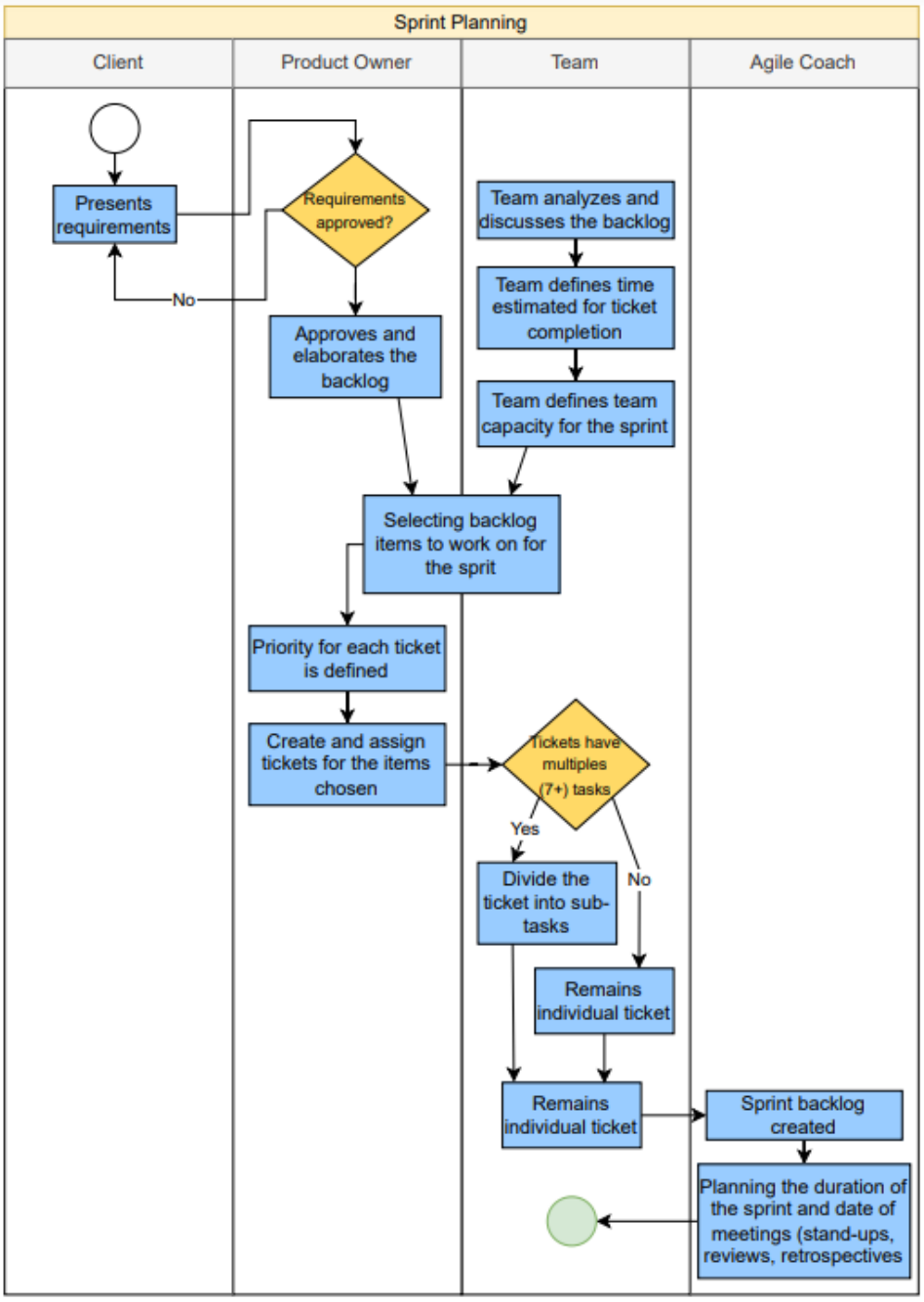
Appendix 2: Model of entities and relationships of the company



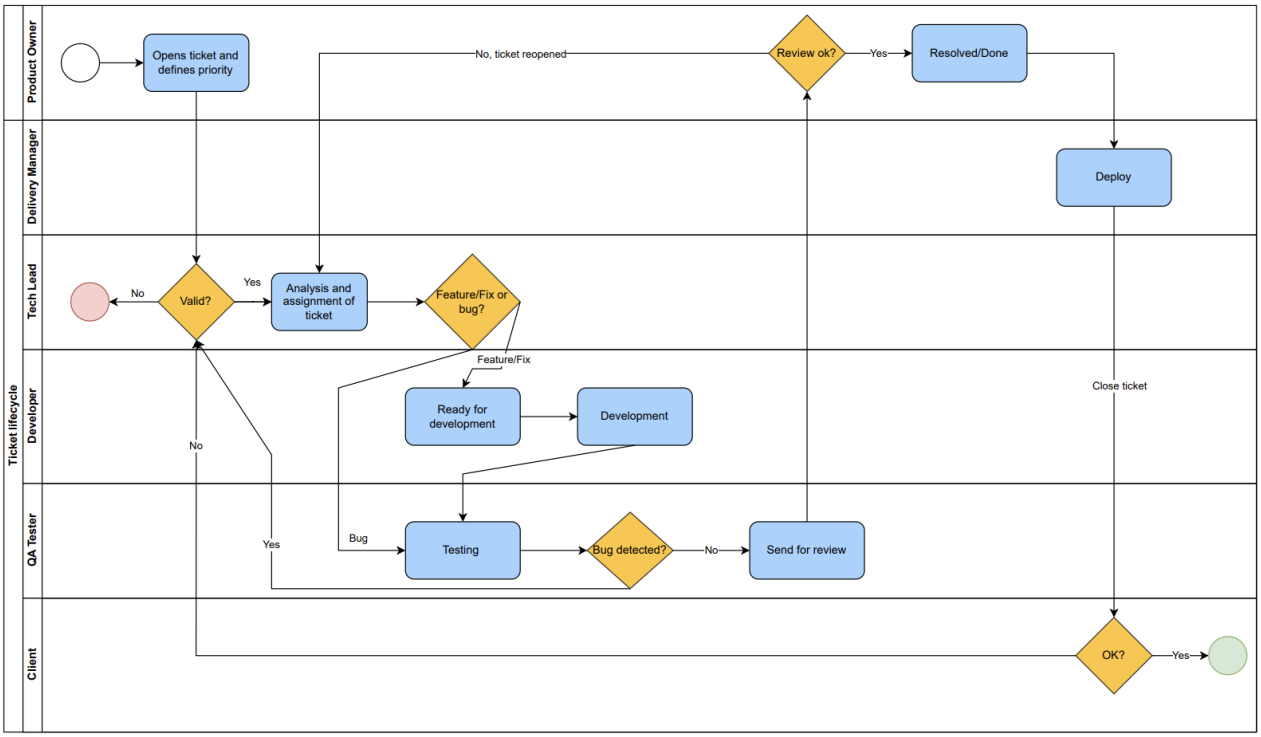
Appendix 3: Model of entities and relationships in backlog management



Appendix 4: Sprint planning swimlane

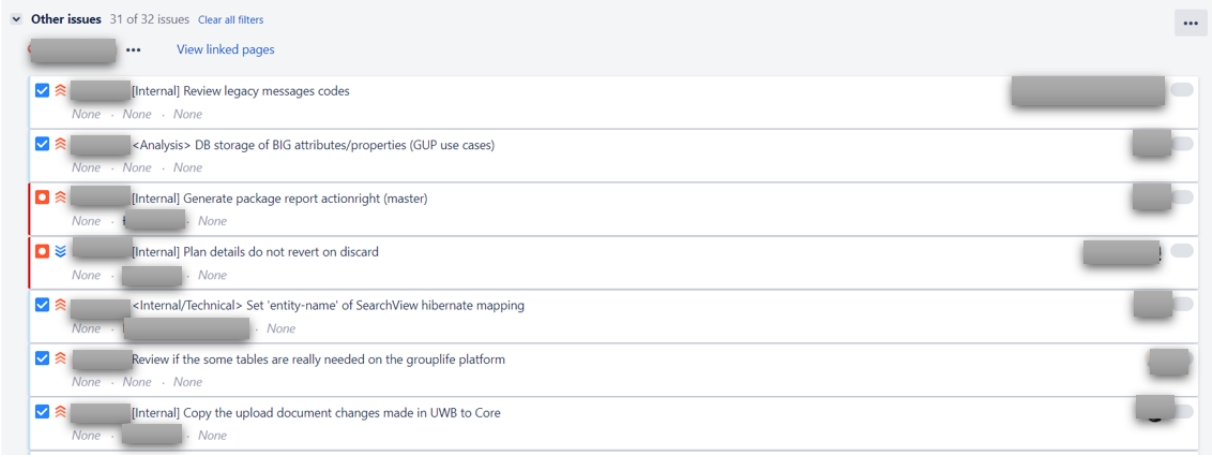


Appendix 5: Ticket creation swimlane

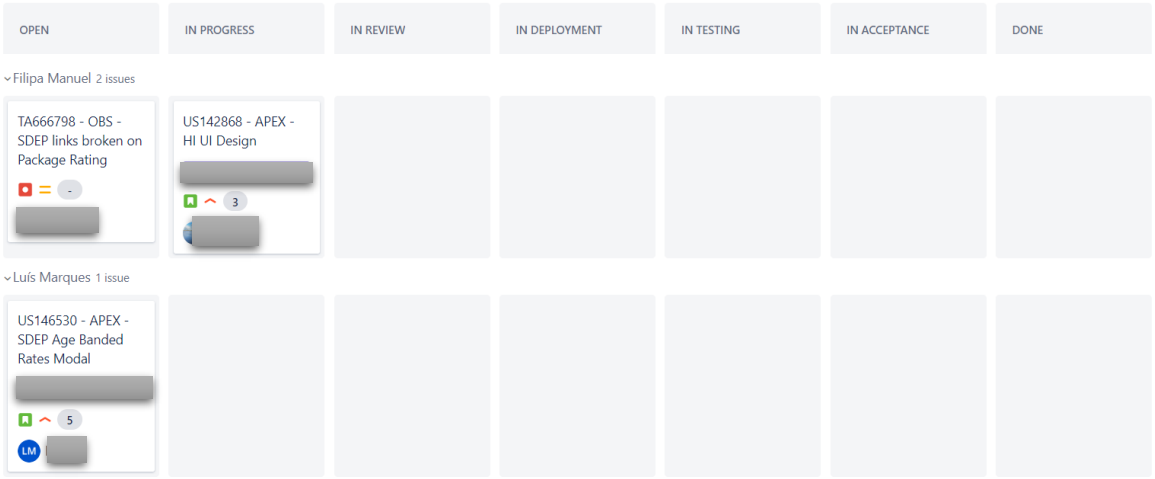


Appendix 6: Benchmarking of FJA

Backlog example:



Kanban Board example:



Appendix 7: DRS Guidelines

Table 1. Design-Science Research Guidelines

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Appendix 8: Interview Agreement

Signed agreement of consent of release of interview data

This shared agreement seeks permission to share and release the data gathered during the interview conducted as part of the master thesis I, Ricardo Nunes, am redacting. The primary objective of this study is to analyze and comprehend agile project management processes and to improve the current situation in the field. This interview will focus majorly on your experience as an employee that utilizes Agile methodologies and will follow a exploratory approach.

I would like to request permissions to potentially share and release the interview data in the following ways:

1. Thesis: The interview data will serve as empirical evidence. The data will be anonymized, ensuring the confidentiality and privacy of the participants.
2. Academic Presentations and Publications: If, by chance, the findings of my thesis are to be presented at academic conferences and/or published in relevant articles in academic journals, the interview data, in anonymous form, may be referenced and cited in these academic presentations and publications.
3. Sharing with Thesis Supervisors and Committee in a form of presentation: The interview data will be shared with my research supervisor and members of the Thesis Committee for the purpose of evaluation, feedback, and assessment of the research study.

All necessary precautions will be taken to maintain the confidentiality and anonymity of the interview participants. The data will be accessed only by authorized individuals involved in the research study and only shared in the contexts identified above.

Therefore, I request your signature as written proof of agreement with the terms above. If there are any conditions that you do not agree with or concern you, I, the interviewer, will be able to supply with you the information. This written consent can be revoked at any time at your request.

Thank you for your consideration of this request. I appreciate your support and cooperation in ensuring the ethical and responsible dissemination of the research findings.

The Researcher, Ricardo Nunes, The Interviewee

_____ _____

Date: __/__/____

Appendix 9: Interview Planning and Conclusions

Stakeholder	Role in the Company	Duration	Key Insights
Miguel Ferreira	Tech Lead	50 minutes	- Early difficult communication with client - Clients sometimes don't fully understand the agile methodology - The onboarding process might take longer than necessary in remote setting.
Ana Saraiva	HR	60 minutes	- Opinions in remote work vary - Introduction of many new employees might change company culture - Remote work might introduce isolation and distraction
Sara Oliveira	HR	55 minutes	- Introduction of many new employees might change company culture - People tend to approve of the flexibility of remote work. - Communication barriers can happen if not checked.
Miguel Castro	Delivery Manager	35 minutes	- Documentation is often outdated or unorganized. - Very favorable on remote work, having already previous experiences with this configuration. - Client sometimes delivers incomplete or inaccurate requirements.
João Carvalho	Delivery Manager	30 minutes	- Doesn't feel like communication is hurt by remote work. - Documentation is often outdated or unorganized. - Onboarding process is harder if the supervisor is remote, and the trainee has no previous work experience.
Márcio Marques	Tech Lead	45 minutes	- Inaccurate requirements are common by the client - The presence of multiple environments demands stronger communication. - Employees, especially newer ones, are more focused on short-term (sprints) than long-term of the project.

Miguel Azevedo	Tech Lead	35 minutes	- Very favorable on remote work - Outdated documentation - Communication takes much longer remotely.
Luís Braga	Scrum Master	40 minutes	- A big influx of new people working remotely, or hybrid, might lead to a change in culture. - Teams must have one senior member to balance the inexperience. - Need of an alignment between the client, project, and project manager/product owner.
Luís Marques	Agile Leader	40 minutes	- Notions of KPIs in the team. - Need to teach and balance different Agile methodologies.
Paulo Sousa	Product Owner and Project Support	45 minutes	- Often inaccurate requirements on part of the client. - Lack of coordination between teams/clients and the intended roadmap/sprint planning. - Dispersed efforts by people working on different teams.
Jorge Miranda	Regional Manager	40 minutes	- Remote work allows for a stronger reach in business, but disconnects the client from the project, leading to inaccurate requirements. - Strong need for project management decision-making structure.
Gonçalo Pereira	Intern (Junior Software Engineer)	20 minutes	- Onboarding process of the company is well-structured. - Lack of connection with remote workers. - Buddy accompaniment is essential.

Script of interviews for problem identification:

- How long have you worked for msg insur:it Iberia?
- Msg works with Agile methodologies. What is your perspective of it?
- Do you find issues in the way the company/your team/your client deals with ticket management? If yes, which ones? Could you exemplify and discusse the causes?
- Do you find issues in the way the company/your team/your client deals with sprint planning? If yes, which ones? Could you exemplify and discuss the causes?
- What are the problems you associate with remote work?
- How do you ensure effective communication in a remote environment?
- How would you deal with conflicts or discord inside the team?
- Regarding the 3 areas references, thinking of possible solutions, which do you think would be best for rapid implementation, in the perspective of a strong cost-benefit relationship?

Appendix 10: Portion of survey presented to the interviewees for the rating of frequency and severity of problems found

Frequency and Severity of Problems Found

This survey is done in collaboration with the company msg insur:it, following the interview you had with Ricardo Nunes. In it, you are meant to evaluate how the problems found throughout the research process (the process discussed with you in the interview) are frequent and how they impact your day-to-day functioning of work. The answers will be evaluated in a Likert Scale from 1-5, with 1 being the least severe/least frequent value and the 5 being the most severe/most frequent value. By responding to this survey, you allow the (anonymous) data gathered to be used in the Ricardo Nunes' dissertation.

Sign in to Google to save your progress. Learn more

* Indicates required question

How often do you feel that the sprint planning is not connected to the long-term roadmap of the project where you're inserted? *

	1	2	3	4	5	
Never	○	○	○	○	○	Always

How severe do you think this problem is? *

	1	2	3	4	5	
Not severe at all	○	○	○	○	○	Very severe

How often do you feel that the daily stand-up meetings have too many members or have an unbalanced focus? *

1	2	3	4	5

Appendix 11: Results of Frequency-Severity Surveys

To reduce visual cluttering, this appendix represents the average results of the surveys answered by the 12 interviewed stakeholders, regarding the perceived frequency and severity of problems:

Problem	Average Frequency of answers	Average Severity of answers
C1.1.	3,7	4,7
C1.2.	4,0	3,0
C2.1.	3,7	4,3
C2.2.	3,7	3,7
C3.1.	3,3	5,0
C3.2.	2,7	3,0
C3.3.	2,5	1,5
C4.1.	2	2
C4.2.	2	2
C4.3.	4,5	3
C5.1.	2,0	2,0
C5.2.	4,0	1,0
C6.1.	2,0	2,0
C6.2.	2,0	3,0
C7.1.	3,6	3,4
C7.2.	3,4	3,2
C7.3.	3,8	3,3
C8.1.	2,4	2,6
C8.2	3,6	2,6
C8.3	4,2	3,5
C9.1.	4,4	2,6
C9.2.	3,1	2,7
C9.3	2,0	2,4

Appendix 12: Portion of survey presented to the interviewees for the rating of effort and impact of implementation of solutions found

Cost-Benefit analysis of the solutions identified

This survey is done in collaboration with the company msg insur:it, following the interview you had with Ricardo Nunes. In it, you are meant to evaluate how the solutions identified throughout the research process (the process discussed with you in the interview) are costly (be it manual effort, time or monetary cost) to implement and how much impact you feel it would have on the causes it intends to solve. The answers will be evaluated in a Likert Scale from 1-5, with 1 being the least impact/least effort and the 5 being the most impact/most effort value. By responding to this survey, you allow the (anonymous) data gathered to be used in the Ricardo Nunes' dissertation.

Sign in to Google to save your progress. Learn more

* Indicates required question

How costly do you believe implementing a tool of alignment between the project roadmap and the sprints planned would be? *

	1	2	3	4	5	
Not costly at all	○	○	○	○	○	Too costly

How much impact do you believe it would have in connecting the overall arc of the project with the team's sprint planning? *

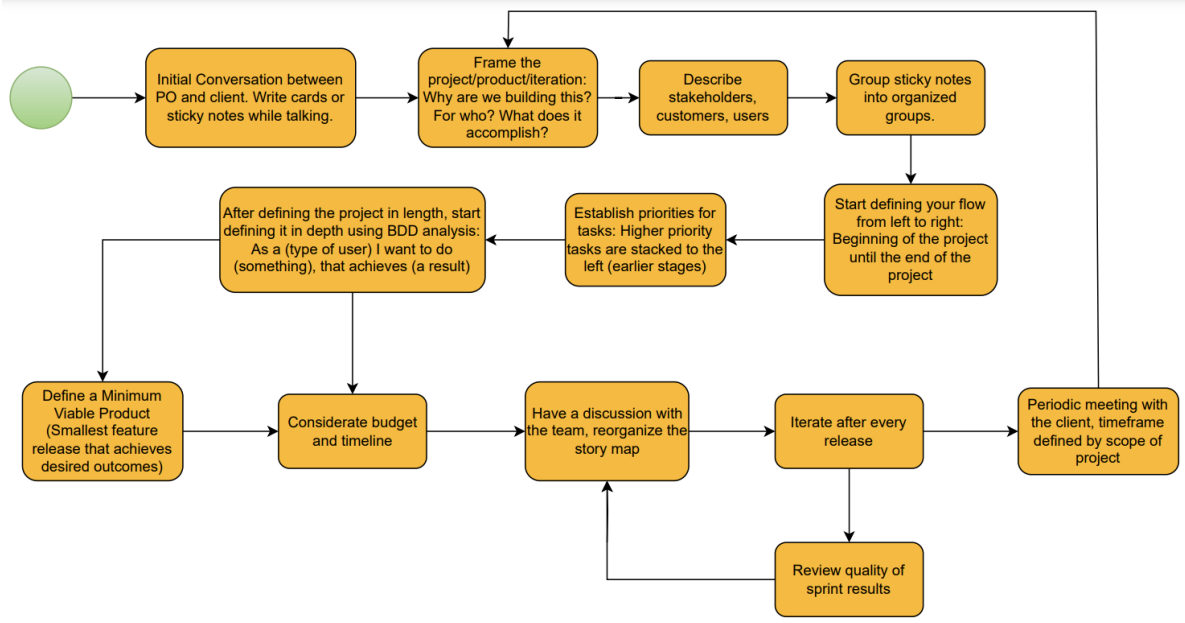
	1	2	3	4	5	
No impact at all	○	○	○	○	○	Severe impact

Appendix 13: Results of Effort-Impact Surveys

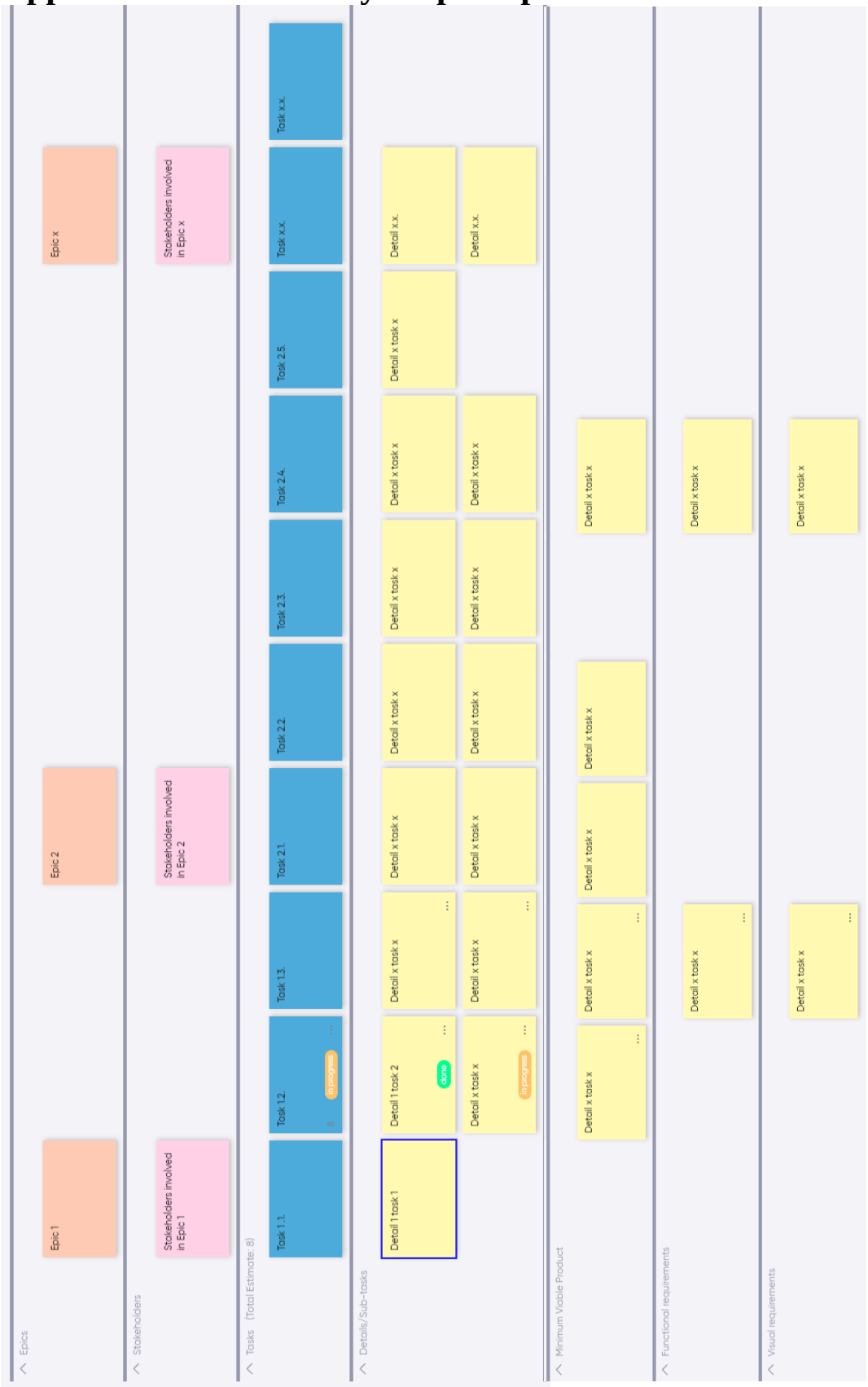
To reduce visual cluttering, this appendix represents the average results of the surveys answered by the 12 interviewed stakeholders, regarding the perceived effort and impact of the proposed solutions:

Solution	Average Impact of Answers	Average Effort of Answers
S1	5	4
S2	4	5
S3	3,5	2
S4	3	1,5
S5	3,5	4,5
S6	3	4,5
S7	4	2,5
S8	2,5	2,5
S9	2,5	2,5
S10	2	1
S11	5	3
S12	3,5	3
S13	3,5	2
S14	2,5	3
S15	3,5	3,5
S16	3,5	4,5
S17	2,5	1
S18	4	4

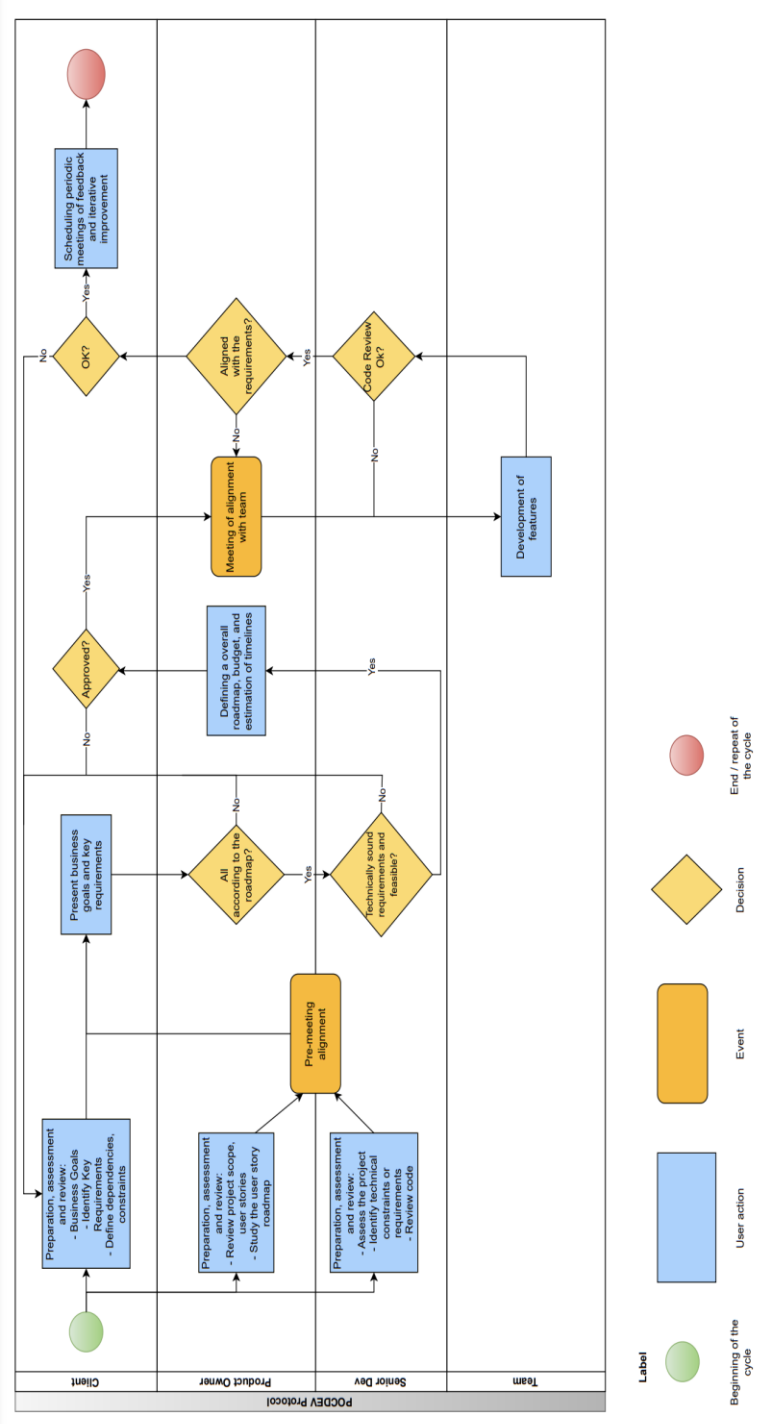
Appendix 14: User story mapping flow



Appendix 15: User Story Map template



Appendix 16: POCDEV Protocol



Appendix 17: Requirement Registry Document

Requirement Registry Document

ID	
Description	
Version	

Type of Issue	
User Story	
Functional Requirements	
Assumptions	
Constraints	
Dependencies	

Suggested Priority	
Acceptance Criteria	

The client:

_____

Date:

The Product Owner:

_____

Appendix 18: Centralized Display of Environments

07 Deployment

Created by Buonanno, Beth, last modified by Peixoto, Renato on Apr 27, 2023

Environments

Servers and Databases

	UW DB	Version	Software version	Wildfly version	Branch	Proxy Configs	Observations
v_s_uw:currentSchema=VHV3;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUW3;	V6.10.1	Sales: 1.11.11.3 Underwriting: 1.4.11.3 PM: 6.9.0	19.1.6	• release/1.11.11.x UW: • release/1.4.11.x	http://le-s-po-lin1:10008/checkConfigs	Dev environment for the next minor version
~~v_s_uw:currentSchema=VHVRELEASE;~~ v_s_uw:currentSchema=VHVHUGE;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUWRELEASE;	V7.3-CFYellow	Sales: 1.12.3.3 Underwriting: 1.5.3.2 PM: 7.4.0	19.1.6	Sales: • no branch used close version UW: • no branch used close version	http://le-s-po-lin1:10038/checkConfigs	This server contains and the 7.3 CFYellow based on a migration from 6.8 to 7.1 and then from 7.1 to 7.3
v_s_uw:currentSchema=VHV2;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUW2;	V7.3-CFRed candidate	Sales: 1.12.3.3-SNAPSHOT Underwriting: 1.5.3.3-SNAPSHOT	19.1.6	Sales: release/1.12.3.x UW: release/1.5.3.x	http://le-s-po-lin1:10018/checkConfigs	Master development branch for V7 (V7)

07 Deployment

Created by Buonanno, Beth, last modified by Peixoto, Renato on Apr 27, 2023

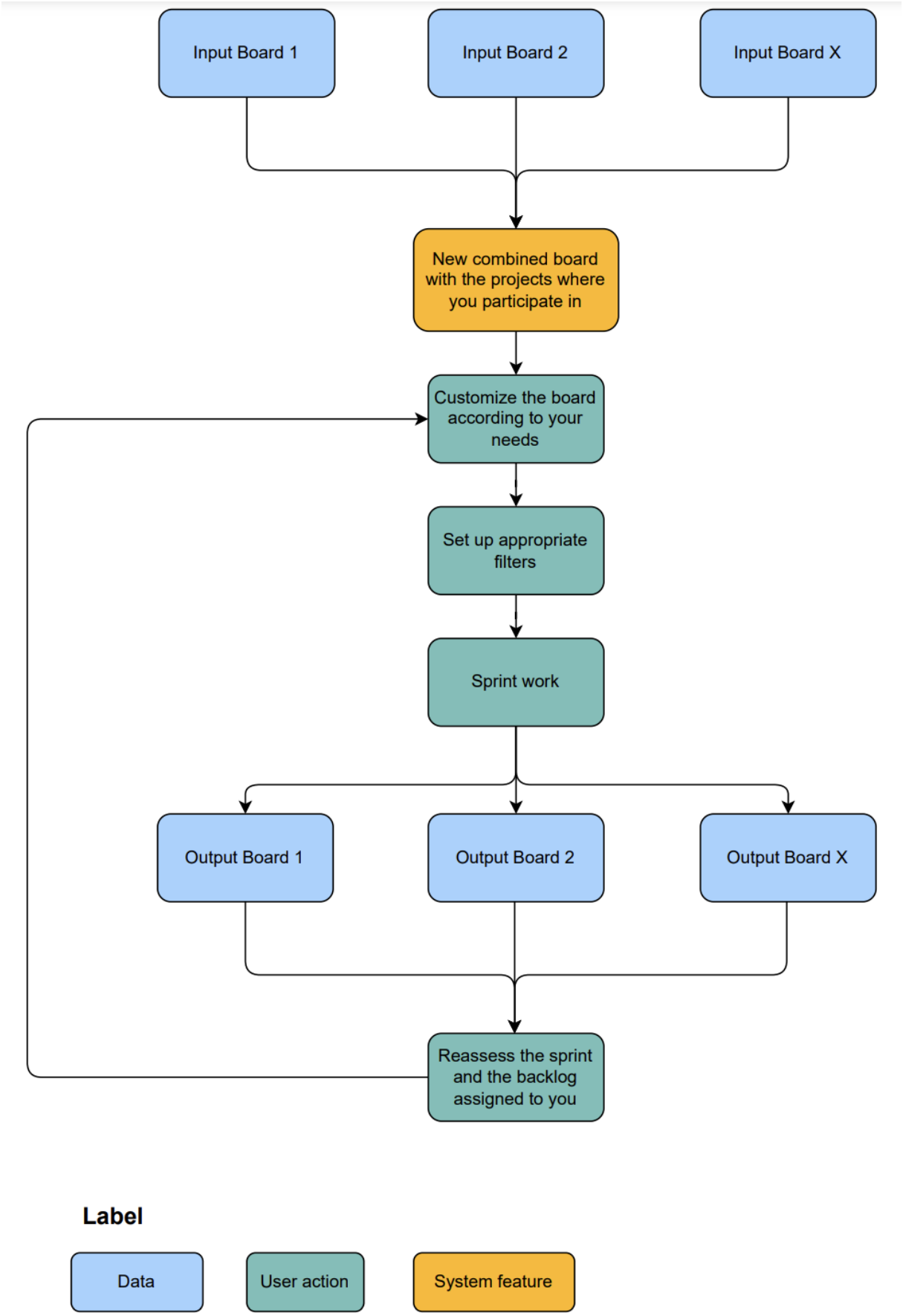
Environments

Servers and Databases

Deployment	CID (proxied)	Sales DB	UW DB	Version	Software version
VHV 3 Running	http://le-s-po-lin1:10008/cid/	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHV3;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUW3;	V6.10.1	Sales: 1.11.11.3 Underwriti 1.4.11.3 PM: 6.9.0
VHV Release (V7.3) Running	http://le-s-po-lin1:10038/cid/	~~jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVRELEASE;~~ jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVHUGE;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUWRELEASE;	V7.3-CFYellow	Sales: 1.12.3.3 Underwriti 1.5.3.2 PM: 7.4.0
VHV 2 Running	http://le-s-po-lin1:10018/cid/	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHV2;	jdbc:db2://le-s-po-lin2.dslocal.com:50000/vhv_s_uw:currentSchema=VHVUW2;	V7.3-CFRed candidate	Sales: 1.12.3.3-SNAPSHOT Underwriti 1.5.3.3-SNAPSHOT

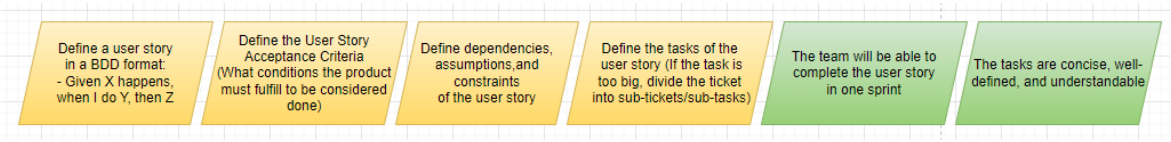
Version	Status	Progress	Start date	Release date	Description
V 7.3 CF Yellow	UNRELEASED			18/Apr/23	Delivery to CE: 24/04/2022
V 7.3 CF Red	UNRELEASED			25/Apr/23	Delivery to CE: 02/05/2023
V 7.5 FC	UNRELEASED			18/May/23	Delivery to CE: 25/05/2023
V 7.4 CF Yellow	UNRELEASED			23/May/23	Delivery to CE: 30/05/2023
V 7.4 CF Red	UNRELEASED			30/May/23	Delivery to CE: 06/06/2023
V 7.5 CF Yellow	UNRELEASED			13/Jun/23	Delivery to CE: 20/06/2023
V 7.5 CF Red	UNRELEASED			20/Jun/23	Delivery to CE: 27/06/2023
V 7.6 CF Red	UNRELEASED			04/Jul/23	Delivery to CE: 11/07/2023

Appendix 19: Workflow of a centralized individual JIRA board

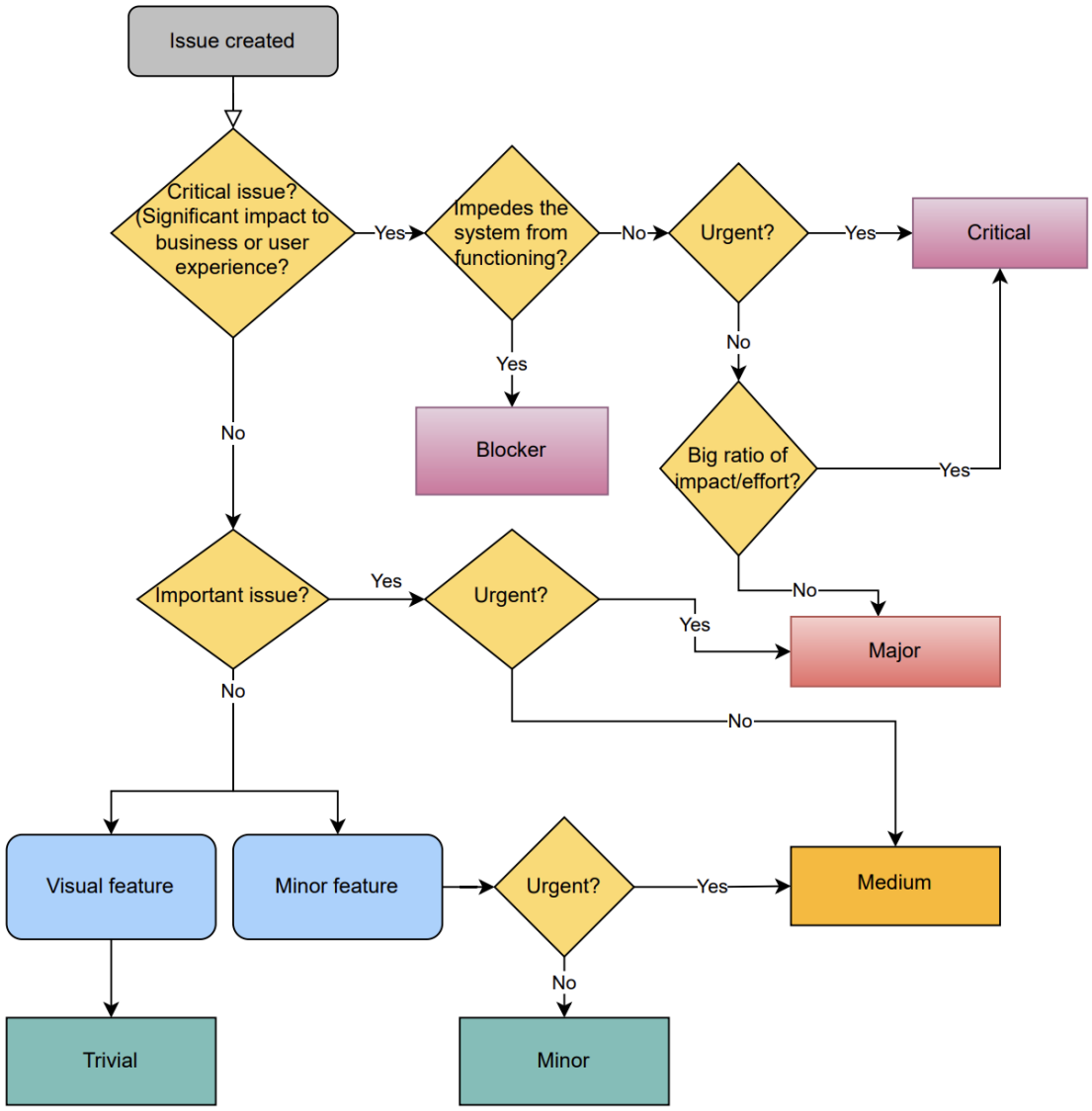


Appendix 20: Definition of Ready and Definition of Done

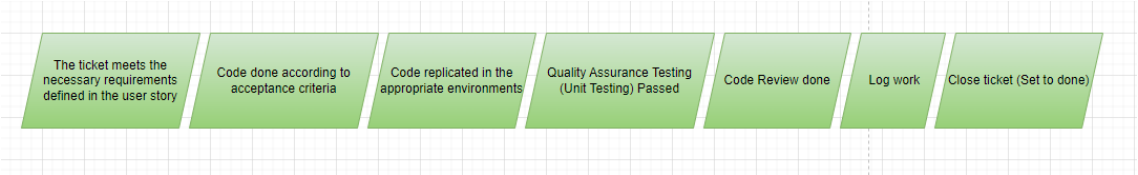
Definition of Ready:



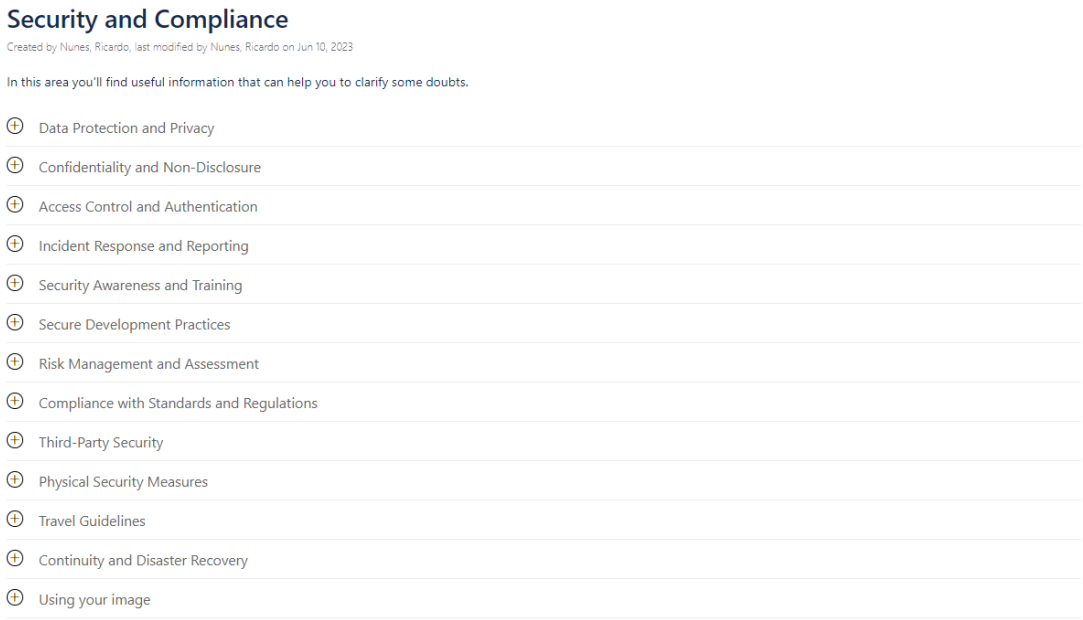
Ticket Priority workflow:



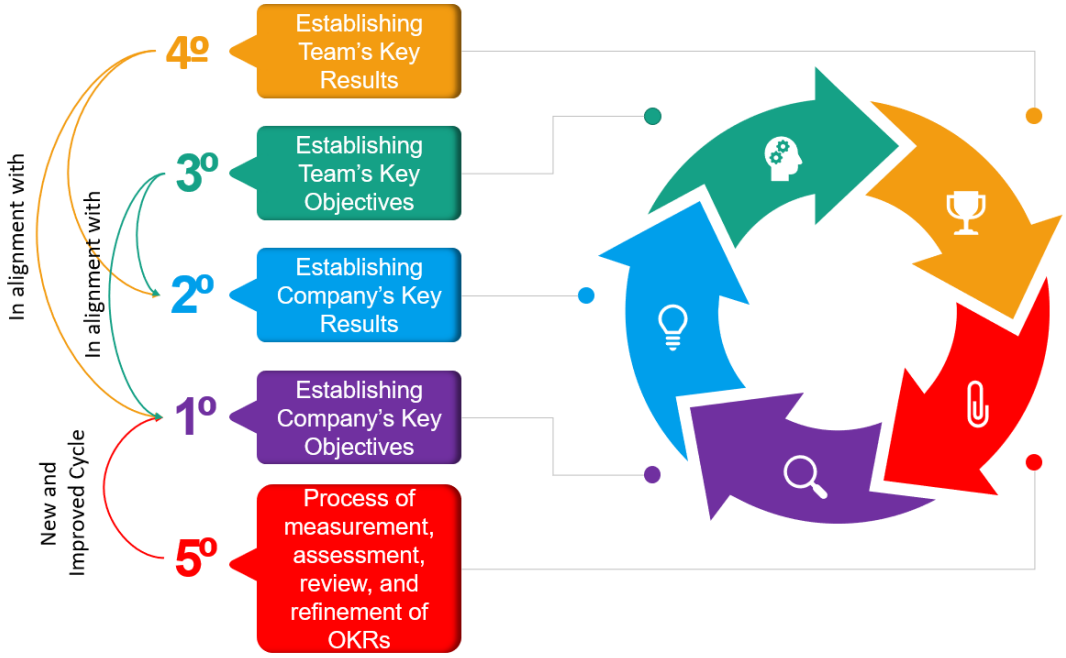
Definition of Done:



Appendix 21: Information and Security Compliance page



Appendix 22: Establishment of OKRs



Appendix 23: Measurement metrics for solutions

Solution	Measurement Metric
S1 – User story Map	Quantitative: - Feedback survey on the workers' part - Analysis of time saved in planning and refinement sessions.
S2 – Protocol between client, product owner, team	Quantitative: - Measuring the time meetings last (with critical analysis) - Tracking the amount of correct requirements gathering in an early stage compared to other projects
S3 - Periodic meetings for alignment between the project roadmap and sprint planning	Quantitative: - Define KPIs for sprint planning and iteratively track them.
S4 - Review of the daily meetings functioning	Quantitative: - Feedback survey on the workers' part - Time spent in meetings now vs before
S5 - Tool of data management	Qualitative: - Critical assessment of outdated information Quantitative: - Ratio of updated information with total size of information
S6 – Homepage Confluence	Qualitative: - Critical assessment of the usefulness of the index Quantitative: - Feedback survey on the workers' part
S7 - Creation of a template document for requirement registry	Qualitative: - Employ control groups in this measure (one team using the solution, another not using it). Compare the level of quality of requirements each team brings.
S8 – Backlog Refinement meeting	Quantitative: - Measure the number of backlog unassigned item and time they stay there - Cycle time: Track the time from when a backlog is identified for development to its conclusion - Analyzing the percentage of unassigned high-priority items compared to the total size of unassigned backlog
S9 - Establish a central display of environments	Quantitative: - Feedback survey on the workers' part

S10 - Documentation of norms for environment management	Quantitative: - Feedback survey on the workers' part
S11 - Development of a tool for automated tests	Quantitative: - Time spent on automated vs manual testing
S12 – Incentive for individual ticket centralization	Quantitative: - Actual ticket resolution time vs ticket estimated resolution time - Auto evaluated and user-reported efficiency
S13 - Establish documentation for ticket management	Quantitative: - Actual ticket resolution time vs ticket estimated resolution time Qualitative: - Critical analysis of the backlog and its quality
S14 - Establishing specific goals, objectives, and key Results	Quantitative: - Comparing the real results with expected results
S15 - Initiative of projects focused on inclusion and team-building activities	Quantitative: - Feedback survey on the workers' part - Rate of participation in the activities - Employee retention
S16 – Betterment of onboarding process	Quantitative: - Time between onboarding and worker being productive - Employee retention - Monitor the completion rate of onboarding training modules and other learning activities - Define KPIs or project milestones for new members. Compare to members who were brought in with a different onboarding process
S17 - Creating a standardized information page for concerns and solutions regarding a distributed company's environment	Quantitative: - Page clicks - Feedback survey on the workers' part
S18 - Creating distributed environment training workshops	Quantitative: - Workshop attendance - Employee retention Qualitative: - *A priori* and *a posteriori* unstructured feedback on the workshops.

Appendix 24: APMDM Diagram Strategy A

ID	Strategy A
Characteristics	This strategy should be employed in a big team, that's working fully remote, with several roles distributed, and in a project with flexible requirements
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information. Since there are plenty of roles in this team, this possibility is even greater. As such, the relevant stakeholders for requirement gathering must be chosen (most likely a Product Owner and Business Analyst) and a protocol of contact between client, chosen members and the overall team should be established. A strong alignment and understanding are necessary in a team of a big dimension.
Responsibilities	A large team with multiple roles should, nonetheless, have only relevant roles. Before beginning the project, one should analyze the necessities and attribute the responsibilities to those with the more knowledge and expertise in the proper areas. Recommended roles are product owner, project manager, delivery manager, business analyst, agile leader, among others.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority.
Backlog management	A big, fully remote team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog, wrong estimation of capacity and effective working hours and the refinement of such. There should be a role of a similar nature to a delivery manager, who is responsible for making sure the tickets are well-defined, worked and delivered on time. The product owner (if one exists) should be making sure the priorities are well-defined and that the client is satisfied with the current progress.
Ticket management	A big, fully remote team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should also be periodic meetings for backlog refinement, be it in the current sprint or in preparation for the next sprints.
Team building	A big, fully remote team might suffer some issues of lack of comradery and belonging. It will be hard, and unlikely, that team members will be able to connect individually between the entire team. However, it's important to improve the engagement and participation of team members in activities that foster positive relations, like, for example, virtual coffee breaks, online games/puzzles, off-work interactions, among others.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 25: APMDM Diagram Strategy B

ID	Strategy B
Characteristics	This strategy should be employed in a big team, that's working fully remote, with several roles distributed, and in a project with rigid requirements.
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information. Since there are plenty of roles in this team, this possibility is even greater. However, in a project with rigid requirements the chances are slimmer, due to constrict and specific requirements. The relevant stakeholders for requirement gathering must be chosen (most likely a Product Owner and Business Analyst) and a protocol of contact between client, chosen members and the overall team should be established, yes, but a strong initial alignment and understanding should be enough, even in a team of a big dimension.
Responsibilities	A large team with multiple roles should, nonetheless, have only relevant roles. Before beginning the project, one should analyze the necessities and attribute the responsibilities to those with the more knowledge and expertise in the proper areas. Recommended roles are product owner, project manager, delivery manager, business analyst, agile leader, among others.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority.
Backlog management	A big, fully remote team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. There should be a role of a similar nature to a delivery manager, who is responsible for making sure the tickets are well-defined, worked and delivered on time. Since the project is more rigid, the priorities and estimations should not be prone to change a lot.
Ticket management	A big, fully remote team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should be periodic meetings to plan the next two to four sprints, as there is not a great expectation of requirements and features changing.
Team building	A big, fully remote team might suffer some issues of lack of comradery and belonging. It will be hard, and unlikely, that team members will be able to connect individually between the entire team. However, it's important to improve the engagement and participation of team members in activities that foster positive relations, like, for example, virtual coffee breaks, online games/puzzles, off-work interactions, among others.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There is more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done, with established meetings, however, with a longer distance between them.

Appendix 26: APMDM Diagram Strategy C

ID	Strategy C
Characteristics	This strategy should be employed in a big team, that's working fully remote, with one to two roles (Developers and most likely PM), and in a project with flexible requirements.
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information. Since the project has flexible requirements, protocol of contact between client, project manager and the overall team should be established. A strong alignment and understanding are necessary in a team of a big dimension. The project manager might find it useful to bring a senior developer with him to the meetings of alignment with client, as they can provide valuable insights and knowledge into technical requirements.
Responsibilities	A large team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team.
Backlog management	A big, fully remote team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. The project manager should be aware of the state of the tickets and of the expected changes in requirements and act accordingly.
Ticket management	A big, fully remote team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should also be periodic meetings for backlog refinement, be it in the current sprint or in preparation for the next sprints.
Team building	A big, fully remote team might suffer some issues of lack of comradery and belonging. It will be hard, and unlikely, that team members will be able to connect individually between the entire team. However, it's important to improve the engagement and participation of team members in activities that foster positive relations, like, for example, virtual coffee breaks, online games/puzzles, off-work interactions, among others.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 27: APMDM Diagram Strategy D

ID	Strategy D
Characteristics	This strategy should be employed in a big team, that's working fully remote, with one to two roles (Developers and most likely PM), and in a project with rigid requirements.
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20 members, all working remote, communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information. Since the project has rigid requirements, the most important thing is a strong initial alignment and understanding of the manager. The project manager might find it useful to bring a senior developer with him to this initial meeting as they can provide valuable insights and knowledge into technical requirements. After this first meeting, update meetings (with less frequency than those in a flexible and iterative project), can have only the presence of the PM and the client.
Responsibilities	A large team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team.
Backlog management	A big, fully remote team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. The project manager should be aware of the state of the tickets and of the expected changes in requirements and act accordingly. This process will be more manageable in a rigid requirements environment.
Ticket management	A big, fully remote team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should be periodic meetings to plan the next two to four sprints, as there is not a great expectation of requirements and features changing.
Team building	A big, fully remote team might suffer some issues of lack of comradery and belonging. It will be hard, and unlikely, that team members will be able to connect individually between the entire team. However, it's important to improve the engagement and participation of team members in activities that foster positive relations, like, for example, virtual coffee breaks, online games/puzzles, off-work interactions, among others.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There is more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done. The established update meetings are what is necessary to make sure that the project is going according to the client's vision and if there is a need for alignment between team and client.

Appendix 28: APMDM Diagram Strategy E

ID	Strategy E
Characteristics	This strategy should be employed in a big team, that's working in a hybrid modality, with several roles distributed, and in a project with flexible requirements
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise. Larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion. The workers who are in-person must be coordinated with the virtual workers, utilizing the established communication channels, as to include, and engage every team member of the team and to make sure information is not lost between physical and virtual settings.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information, especially in a hybrid setting. Since there are plenty of roles in this team, this possibility is even greater. As such, the relevant stakeholders for requirement gathering must be chosen (most likely a Product Owner and Business Analyst) and a protocol of contact between client, chosen members and the overall team should be established. A strong alignment and understanding are necessary in a team of a big dimension.
Responsibilities	A large team with multiple roles should, nonetheless, have only relevant roles. Before beginning the project, one should analyze the necessities and attribute the responsibilities to those with the more knowledge and expertise in the proper areas. Recommended roles are product owner, project manager, delivery manager, business analyst, agile leader, among others.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team.
Backlog management	A big, hybrid team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog, wrong estimation of capacity and effective working hours and the refinement of such. There should be a role of a similar nature to a delivery manager, who is responsible for making sure the tickets are well-defined, worked and delivered on time. The product owner (if one exists) should be making sure the priorities are well-defined and that the client is satisfied with the current progress. It's also preferable to attribute related tickets to people choosing the same modality of work.
Ticket management	A big, hybrid team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should also be periodic meetings for backlog refinement, be it in the current sprint or in preparation for the next sprints.
Team building	A big, hybrid team might suffer some issues of lack of comradery and belonging. This might be further intensified by the choosing of different modalities of work. There is the possibility of tension and/or lack of communication between remote and physical workers. Besides the chosen activities of strategy, A, B, C, D for virtual workers, it's important to establish the same team building between physical workers and between those choosing different configurations of working. Examples could be team-building workshops or one-on-one individual meetings for discussion and sharing of experiences and knowledge.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 29: APMDM Diagram Strategy F

ID	Strategy F
Characteristics	This strategy should be employed in a big team, that's working in a hybrid modality, with several roles distributed, and in a project with rigid requirements
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion. The workers who are in-person must be coordinated with the virtual workers, utilizing the established communication channels, as to include, and engage every team member of the team and to make sure information is not lost between physical and virtual settings.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information, especially in a hybrid setting. Since there are plenty of roles in this team, this possibility is even greater. However, in a project with rigid requirements the chances are slimmer, due to constrict and specific requirements. The relevant stakeholders for requirement gathering must be chosen (most likely a Product Owner and Business Analyst) and a protocol of contact between client, chosen members and the overall team should be established, yes, but a strong initial alignment and understanding should be enough, even in a team of a big dimension.
Responsibilities	A large team with multiple roles should, nonetheless, have only relevant roles. Before beginning the project, one should analyze the necessities and attribute the responsibilities to those with the more knowledge and expertise in the proper areas. Recommended roles are product owner, project manager, delivery manager, business analyst, agile leader, among others.
Decision-making	Since the team is of a large dimension and of a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team.
Backlog management	A big, hybrid team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. There should be a role of a similar nature to a delivery manager, who is responsible for making sure the tickets are well-defined, worked and delivered on time. Since the project is more rigid, the priorities and estimations should not be prone to change a lot.
Ticket management	A big, hybrid team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should be periodic meetings to plan the next two to four sprints, as there is not a great expectation of requirements and features changing.
Team building	A big, hybrid team might suffer some issues of lack of comradery and belonging. This might be further intensified by the choosing of different modalities of work. There is the possibility of tension and/or lack of communication between remote and physical workers. Besides the chosen activities of strategy, A, B, C, D for virtual workers, it's important to establish the same team building between physical workers and between those choosing different configurations of working. Examples could be team-building workshops or one-on-one individual meetings for discussion and sharing of experiences and knowledge.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There should be more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done, with established meetings, with a longer distance between them.

Appendix 30: APMDM Diagram Strategy G

ID	Strategy G
Characteristics	This strategy should be employed in a big team, that's working in a hybrid modality, with one to two roles (Developers and most likely PM), and in a project with flexible requirements.
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, who are all working remote, the communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise, where larger tasks can be attributed, if necessary, in the purpose of achieving the sub-tasks in a faster fashion. The workers who are in-person must be coordinated with the virtual workers, utilizing the established communication channels, as to include, and engage every team member of the team and to make sure information is not lost between physical and virtual settings.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information, especially in a hybrid setting. Since the project has flexible requirements, a protocol of contact between client, project manager and the overall team should be established. A strong alignment and understanding are necessary in a team of a big dimension. The project manager might find it useful to bring a senior developer with him to the meetings of alignment with client, as they can provide valuable insights and knowledge into technical requirements.
Responsibilities	A large team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. If necessary, to simplify the process of opinion and knowledge sharing, the sub-teams as a whole can express their opinion, reducing individuality and cluttering of communication. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Backlog management	A big, hybrid team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. The project manager should be aware of the state of the tickets and of the expected changes in requirements and act accordingly.
Ticket management	A big, hybrid team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should also be periodic meetings for backlog refinement, be it in the current sprint or in preparation for the next sprints.
Team building	A big, hybrid team might suffer some issues of lack of comradery and belonging. This might be further intensified by the choosing of different modalities of work. There is the possibility of tension and/or lack of communication between remote and physical workers. Besides the chosen activities of strategy, A, B, C, D for virtual workers, it's important to establish the same team building between physical workers and between those choosing different configurations of working. Examples could be team-building workshops or one-on-one individual meetings for discussion and sharing of experiences and knowledge.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 31: APMDM Diagram Strategy H

ID	Strategy H
Characteristics	This strategy should be employed in a big team, that's working in a hybrid modality, with one to two roles (Developers and most likely PM), and in a project with rigid requirements.
Internal Communication/ Coordination	A larger team is almost always going to face more communication problems than a smaller team. Since the team is consisted of more than 20+ members, all working remote, communication channels must be well established, with regular team meetings (preferably daily) to keep track of the project. The team should be grouped in "sub-teams" with similar fields of expertise. Larger tasks can be attributed, if necessary, to complete sub-tasks in a faster fashion. The workers who are in-person must be coordinated with the virtual workers, utilizing the established communication channels, as to include, and engage every team member of the team and to make sure information is not lost between physical and virtual settings.
External Communication/ Collaboration	A larger team consisting of several members through which the information is going to circulate is more likely to generate misunderstandings, confusions, and bad information, especially in a hybrid setting. Due to the rigid requirements of the project, the chances are slimmer, due to constrict and specific requirements. Since the project has rigid requirements, the most important thing is a strong initial alignment and understanding of the manager. The project manager might find it useful to bring a senior developer with him to this initial meeting as they can provide valuable insights and knowledge into technical requirements. After this first meeting, update meetings (with less frequency than those in a flexible and iterative project), can have only the presence of the PM and the client.
Responsibilities	A large team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Backlog management	A big, hybrid team, in general, will generate a high volume of tickets. Without proper and solid efforts, a high volume of tickets will generate cluttering of the backlog. The project manager should be aware of the state of the tickets and of the expected changes in requirements and act accordingly. This process will be more manageable and less time-consuming in a rigid requirements environment.
Ticket management	A big, hybrid team, with the high volume of tickets generated, will also reduce the individual visibility of work for each member. Besides the team's board (if working on SCRUM or KANBAN), the individual should also choose to employ a centralized individual ticketing organization system, to be able to maintain and regulate his own load of work. There should be periodic meetings to plan the next two to four sprints, as there is not a great expectation of requirements and features changing.
Team building	A big, hybrid team might suffer some issues of lack of comradery and belonging. This might be further intensified by the choosing of different modalities of work. There is the possibility of tension and/or lack of communication between remote and physical workers. Besides the chosen activities of strategy, A, B, C, D for virtual workers, it's important to establish the same team building between physical workers and between those choosing different configurations of working. Examples could be team-building workshops or one-on-one individual meetings for discussion and sharing of experiences and knowledge.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There is more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done. The established update meetings are what is necessary to make sure that the project is going according to the client's vision and if there is a need for alignment between team and client.

Appendix 32: APMDM Diagram Strategy I

ID	Strategy I
Characteristics	This strategy should be employed in a mid-sized team, with several roles distributed, and in a project with flexible requirements
Internal Communication/ Coordination	The problems found in strategies A-H regarding internal communication are reduced as the members of the team are reduced. A mid-sized team (8 to 20 members) still faces communication problems, but they should be more sporadic. The tightness and standard of communication tools should be maintained, and the regular team meetings can be adjusted to the size and characteristics of the team. Given that this is a project with flexible requirements, it's not recommended to exceed 2 days between periodic meetings. Larger tasks can be associated with different pairings of 2-3 members, to complete the dependent sub-tasks in a connected and swifter action.
External Communication/ Collaboration	A medium-sized team will pose a lesser threat, but still a present one of misunderstanding and ill-defined requirements. To mitigate this, since the project has flexible requirements, a protocol of contact between client, project manager and the overall team should, once again, be established. A strong alignment and understanding are important in a team of this dimension, however, confusions and misalignments are more easily identified and solved. The project manager might find it useful to bring a senior developer with him to the meetings of alignment with client, as they can provide valuable insights and knowledge into technical requirements, but it might not be necessary.
Responsibilities	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Decision-making	Since the team still has some dimension and a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority.
Backlog management	In a medium-sized team, the cluttering of the backlog might occur not because of the amount of tickets present (which might still occur), but because of the lack of correct priority given. Understanding the urgency, cost, and impact that the issue might have is paramount to take care of the backlog in a orderly fashion, with reigning common sense. The workload should be correctly calculated and estimated and there should be regular sessions for refinement of the backlog.
Ticket management	A mid-sized team invites a different kind of problem from the big teams. The lack of correct prioritization might also transfer to the individual members, which makes it important for them to be able to assess them individually and analyze them. The centralized individual ticketing organization system should be used, to be able to organize the tickets by priority and by effort required to complete.
Team building	In a mid-sized team, be it fully remotely or in a hybrid configuration, the spontaneous creation of bonds and belonging start to happen, but they still require incentive. It's recommended for team members to participate in team days, team activities and sessions, and to promote the communication between remote and virtual workers.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 33: APMDM Diagram Strategy J

ID	Strategy J
Characteristics	This strategy should be employed in a mid-sized team, with several roles distributed, and in a project with rigid requirements
Internal Communication/ Coordination	The problems found in strategies A-H regarding internal communication are reduced as the members of the team are reduced. A mid-sized team (8 to 20 members) still faces communication problems, but they should be more sporadic. The tightness and standard of communication tools should be maintained, and the regular team meetings can be adjusted to the size and characteristics of the team. Given that this is a project with flexible requirements, it's not recommended to exceed 2 days between periodic meetings. Larger tasks can be associated with different pairings of 2-3 members, to complete the dependent sub-tasks in a connected and swifter action.
External Communication/ Collaboration	A medium-sized team will pose a lesser threat, but still a present one of misunderstanding and ill-defined requirements. In a project with rigid requirements the chances are even slimmer, due to constrict and specific requirements. Since the project has rigid requirements, the most important thing is a strong initial alignment and understanding of the manager. The project manager might find it useful to bring a senior developer with him to this initial meeting as they can provide valuable insights and knowledge into technical requirements. After this first meeting, update meetings (with less frequency than those in a flexible and iterative project), can have only the presence of the PM and the client.
Responsibilities	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Decision-making	Since the team still has some dimension and a variety of roles, the decision-making process should be of an involvement nature, with team members voicing their opinions, and reaching a consensus. It's also important to manage conflict when team members do not have their preferences or advice chosen in detriment of the majority.
Backlog management	In a medium-sized team, the cluttering of the backlog might occur not because of the amount of tickets present (which might still occur), but because of the lack of correct priority given. Understanding the urgency, cost, and impact that the issue might have is paramount to take care of the backlog in an orderly fashion, with reigning common sense. The requirements should not change too much and if the prioritization is well-handled from the start the refinement of the backlog will not be a necessity frequently.
Ticket management	A mid-sized team invites a different kind of problem from the big teams. The lack of correct prioritization might also transfer to the individual members, which makes it important for them to be able to assess them individually and analyze them. The centralized individual ticketing organization system should be used, to be able to organize the tickets by priority and by effort required to complete.
Team building	In a mid-sized team, be it fully remotely or in a hybrid configuration, the spontaneous creation of bonds and belonging start to happen, but they still require incentive. It's recommended for team members to participate in team days, team activities and sessions, and to promote the communication between remote and virtual workers.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There is more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done. The established update meetings are what is necessary to make sure that the project is going according to the client's vision and if there is a need for alignment between team and client.

Appendix 34: APMDM Diagram Strategy K

ID	Strategy K
Characteristics	This strategy should be employed in a mid-sized team, with 1-2 roles, and in a project with flexible requirements
Internal Communication/ Coordination	The problems found in strategies A-H regarding internal communication are reduced as the members of the team are reduced. A mid-sized team (8 to 20 members) still faces communication problems, but they should be more sporadic. The tightness and standard of communication tools should be maintained, and the regular team meetings can be adjusted to the size and characteristics of the team. Given that this is a project with flexible requirements, it's not recommended to exceed 2 days between periodic meetings. Larger tasks can be associated with different pairings of 2-3 members, to complete the dependent sub-tasks in a connected and swifter action.
External Communication/ Collaboration	A medium-sized team will pose a lesser threat, but still a present one of misunderstanding and ill-defined requirements. To mitigate this, since the project has flexible requirements, a protocol of contact between client, project manager and the overall team should, once again, be established. A strong alignment and understanding are important in a team of this dimension, however, confusions and misalignments are more easily identified and solved. The project manager might find it useful to bring a senior developer with him to the meetings of alignment with client, as they can provide valuable insights and knowledge into technical requirements, but it might not be necessary.
Responsibilities	A mid-sized team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Backlog management	In a medium-sized team, the cluttering of the backlog might occur not because of the amount of tickets present (which might still occur), but because of the lack of correct priority given. Understanding the urgency, cost, and impact that the issue might have is paramount to take care of the backlog in a orderly fashion, with reigning common sense. The workload should be correctly calculated and estimated and there should be regular sessions for refinement of the backlog.
Ticket management	A mid-sized team invites a different kind of problem from the big teams. The lack of correct prioritization might also transfer to the individual members, which makes it important for them to be able to assess them individually and analyze them. The centralized individual ticketing organization system should be used, to be able to organize the tickets by priority and by effort required to complete.
Team building	In a mid-sized team, be it fully remotely or in a hybrid configuration, the spontaneous creation of bonds and belonging start to happen, but they still require incentive. It's recommended for team members to participate in team days, team activities and sessions, and to promote the communication between remote and virtual workers.
Flexibility and changes	An iterative project with flexible requirements is liable to introduce changes in scope, size, budget, timeframe, etc. Such changes might lead to a different roadmap or vision of the project. It's important for the team to stay aligned with the project. For this to be feasible, periodic, and frequent meetings with the customer should be arranged and periodic meetings of alignment of the long-term roadmap with the team should be established, to maintain a sense of understanding and synchrony for the whole team.

Appendix 35: APMDM Diagram Strategy L

ID	Strategy L
Characteristics	This strategy should be employed in a mid-sized team, with 1-2 roles, and in a project with rigid requirements
Internal Communication/ Coordination	The problems found in strategies A-H regarding internal communication are reduced as the members of the team are reduced. A mid-sized team (8 to 20 members) still faces communication problems, but they should be more sporadic. The tightness and standard of communication tools should be maintained, and the regular team meetings can be adjusted to the size and characteristics of the team. Given that this is a project with flexible requirements, it's not recommended to exceed 2 days between periodic meetings. Larger tasks can be associated with different pairings of 2-3 members, to complete the dependent sub-tasks in a connected and swifter action.
External Communication/ Collaboration	A medium-sized team will pose a lesser threat, but still a present one of misunderstanding and ill-defined requirements. In a project with rigid requirements the chances are even slimmer, due to constrict and specific requirements. Since the project has rigid requirements, the most important thing is a strong initial alignment and understanding of the manager. The project manager might find it useful to bring a senior developer with him to this initial meeting as they can provide valuable insights and knowledge into technical requirements. After this first meeting, update meetings (with less frequency than those in a flexible and iterative project), can have only the presence of the PM and the client.
Responsibilities	A mid-sized team with few roles is going to make members share responsibilities of different roles. If a company wants to choose this strategy, the project manager will, most likely, act as product owner, business analyst, tech lead, delivery manager, and more. A person with time management, communication, problem solving, and business sense is essential. This person will be the key to the coordination between the team and the project.
Decision-making	The project manager, who is the most knowledgeable, integrated, and participative of the project among all the stakeholders, should have the final say in team decisions. Nonetheless, he should hear what the team has to say, having the critical ability to understand the needs of the team in combination with the client's requirements. The project manager should be able to explain and justify his decisions, to mitigate the doubts and enhance the sense of trust and responsibility to client and team. It's important for the project manager to also receive feedback in an equal sense from remote and physical workers, not allowing modalities of work to have different weights in the decision.
Backlog management	In a medium-sized team, the cluttering of the backlog might occur not because of the amount of tickets present (which might still occur), but because of the lack of correct priority given. Understanding the urgency, cost, and impact that the issue might have is paramount to take care of the backlog in a orderly fashion, with reigning common sense. The rigid requirements dictate that a regular refinement of the backlog should not be necessary, if the prioritization is done accordingly.
Ticket management	A mid-sized team invites a different kind of problem from the big teams. The lack of correct prioritization might also transfer to the individual members, which makes it important for them to be able to assess them individually and analyze them. The centralized individual ticketing organization system should be used, to be able to organize the tickets by priority and by effort required to complete.
Team building	In a mid-sized team, be it fully remotely or in a hybrid configuration, the spontaneous creation of bonds and belonging start to happen, but they still require incentive. It's recommended for team members to participate in team days, team activities and sessions, and to promote the communication between remote and virtual workers.
Flexibility and changes	A project with rigid requirements will have minimal changes introduced in a longer timeframe. There is more focus on understanding the specifications, as they are less prone to alter, however, the customer can be less involved if the initial alignment of stakeholders is well-done. The established update meetings are what is necessary to make sure that the project is going according to the client's vision and if there is a need for alignment between team and client.

Appendix 36: APMDM Diagram Strategy M

ID	Strategy M
Characteristics	This strategy should be employed in a small team, in a project with flexible requirements
Internal Communication/ Coordination	The internal communication in a small team starts to be more spontaneous and genuine, without much need for initial intervention to implement it. However, other problems might come up that should be noted and tackled. In a smaller team, the familiar nature might induce in relaxation of documentation and communication. It's important to adhere to the standards of communication and documentation established. This means that, although meetings should be conducted on a "when needed" basis, a weekly alignment meeting is advised to keep track of the team's progress regarding the project.
External Communication/ Collaboration	A smaller team is more easily aligned with the project and the overall vision of it. Flaws or misunderstandings are identifiable and worked upon if the project manager is attentive. As such, the protocol between client and team can have a single point of connection (for example, the project manager/product owner), which then can pass the information to the team, in a concise and understandable manner. The line of direct contact between team and product manager allows for doubts to be easily transmitted and quickly answered.
Responsibilities	A smaller team sees members sharing responsibilities. It's important to continue communicating clearly, as the lack of information from just one single member might impede work from advancing. A team member also has to be aware of the responsibility of being an integrating part of a small team, making sure that he only puts work on other's hands if absolutely necessary.
Decision-making	A smaller team is more directly involved with each other. They are aligned with the project and are working closely with the project. They are more aware of the flexible requirements, the changes, and the technical aspects of the application. The process of decision-making should be organic, with the project manager acting only as a moderator, and allowing the team as a whole to reach a consensus.
Backlog management	In a smaller team, the cluttering of the backlog might happen when there are too many issues/tickets for the capacity of the team. It's important to keep the priority of the tickets correct and standardized, as well as communicating to the customer the possibilities and constraints of each sprint. The dependencies created by bigger tickets on smaller tickets are also factors to have in consideration. A periodic meeting to discuss backlog refinement is recommended (timeframe to be decided on a project-to-project basis).
Ticket management	In a smaller team, an individual has more responsibilities in his personal management of time. It's important a team member abides by the standards and templates chosen by the organization for ticket functioning. One should not accept more than what his capacity allows and should work based on the priority defined by the project manager. Regular reviews of individuals should be conducted by the project manager, to avoid the piling of tickets on one single worker.
Team building	In a smaller team, the team building should happen in a more organic way. Besides organizing team days where a team focuses only on team building and extracurricular events, the project manager should promote knowledge and experience sharing between members and trying to foster connections in a natural way.
Flexibility and changes	A project with flexible requirements is always dangerous, in the sense that it can change at any moment. In a small team, this can be a problem if a team is not well-aligned with the vision of the client. In general, for a solid and constant progress, the project manager should conduct a review of the project roadmap as often as he sees fit, stating priorities, key objectives and results for the company, the team, and the client.

Appendix 37: APMDM Diagram Strategy N

ID	Strategy N
Characteristics	This strategy should be employed in a small team, in a project with rigid requirements
Internal Communication/ Coordination	The internal communication in a small team starts to be more spontaneous and genuine, without much need for initial intervention to implement it. However, other problems might come up that should be noted and tackled. In a smaller team, the familiar nature might induce in relaxation of documentation and communication. It's important to adhere to the standards of communication and documentation established. This means that, although meetings should be conducted on a "when needed" basis, a weekly alignment meeting is advised to keep track of the team's progress regarding the project.
External Communication/ Collaboration	A smaller team is more easily aligned with the project and the overall vision of it. Flaws or misunderstandings are identifiable and worked upon if the project manager is attentive. As such, the protocol between client and team can have a single point of connection (for example, the project manager/product owner), which then can pass the information to the team, in a concise and understandable manner. The line of direct contact between team and product manager allows for doubts to be easily transmitted and quickly answered.
Responsibilities	A smaller team sees members sharing responsibilities. It's important to continue communicating clearly, as the lack of information from just one single member might impede work from advancing. A team member also has to be aware of the responsibility of being an integrating part of a small team, making sure that he only puts work on other's hands if absolutely necessary.
Decision-making	A smaller team is more directly involved with each other. They are aligned with the project and are working closely with the project. The rigid requirements should limit the occurring of decision-making moments, however, the process should be organic, with the project manager acting only as a moderator, and allowing the team as a whole to reach a consensus.
Backlog management	In a smaller team, the cluttering of the backlog might happen when there are too many issues/tickets for the capacity of the team. It's important to keep the priority of the tickets correct and standardized, as well as communicating to the customer the possibilities and constraints of each sprint. The dependencies created by bigger tickets on smaller tickets are also factors to have in consideration. In a rigid requirements project, it's the responsibility of the project manager to make sure the team capacity is well fit to take care of the work appointed to each timely planned sprint.
Ticket management	In a smaller team, an individual has more responsibilities in his personal management of time. It's important a team member abides by the standards and templates chosen by the organization for ticket functioning. One should not accept more than what his capacity allows and should work based on the priority defined by the project manager. Regular reviews of individuals should be conducted by the project manager, to avoid the piling of tickets on one single worker.
Team building	In a smaller team, the team building should happen in a more organic way. Besides organizing team days where a team focuses only on team building and extracurricular events, the project manager should promote knowledge and experience sharing between members and trying to foster connections in a natural way.
Flexibility and changes	A project with rigid requirements is, supposedly, more predictable. It's the project manager's role and responsibility to align the team with the long-term vision of the project, that should not change that easily. Meetings of alignment can be had in a longer time-frame (6-12 months depending on the project) and are merely for observations and tracking of key objectives and key results.

Appendix 38: APMDM Metrics

Category	Metrics
Internal Communication/ Coordenation	General: - Feedback (Qualitative and Quantitative) - Task Completion Time (Quantitative; Before vs After) - Task Dependencies (Quantitative; Before vs After) - Sprint efficiency (Quantitative; Tickets Estimated vs Done) Size of the teams: • Big and medium teams - Efficiency in communication (Qualitative) - Knowledge sharing (Qualitative) - Feeling of inclusion and Trust (Qualitative) Configuration of work: • Hybrid - Knowledge sharing and communication between workers in different settings (Qualitative) Roles in the team: • Several roles in the team - Role Clarity (Qualitative: Team's understanding of roles and ability) Flexibility of requirements: • Flexible requirements - Adaptability to changes (Qualitative)
External Communication/ Collaboration	General: - Client Satisfaction (Qualitative and Quantitative) - Protocol Productivity (Qualitative, assess the effectiveness of the POCDEV protocol). Flexibility of requirements: • Flexible requirements - Requirement Accuracy (Quantitative; Percentage of deliverables that match expectation of clients, frequency, and severity of deviations). - Capacity to deal with deviations (Quantitative and Qualitative). - Alignment of long-term vision with short-term vision (Qualitative: Assess how the members evaluate the sprint regarding the changes of requirements and the roadmap of the project).

Responsibilities	<u>General:</u> - Task Ownership (Quantitative: Ability of individual to take ownership, and completion time of individual tickets) <u>Roles in the team:</u> • Several roles in the team - Role adaptability (Qualitative: Ability of team members to change and adapt to roles) • 1-2 Roles in the team - Project Manager performance (Qualitative and Quantitative: Feedback and Surveys) <u>Flexibility of requirements:</u> • Flexible project - Project flexibility (Qualitative: Ability to adapt responsibilities and redistribution of tasks in response to changes)
Decision-Making	<u>General:</u> - Decision-Making Time (Quantitative: Before vs After) - Decision Accuracy (Qualitative: Results of the decisions vs Expectations of decisions) - Decision Review: (Qualitative: Periodic process of refining the process of decision-making) - Decision documentation: (Qualitative: Assess easiness of access to the clarity, details, and results of decisions) <u>Size of the team:</u> • Big to medium teams: - Decision alignment (Quantitative: Percentage of key decisions with unanimity or strong percentage of agreement) • Small teams: - Participation rate (Quantitative: Members of the team involved in the decision-making process) <u>Configuration of work:</u> • Hybrid teams: - Conflict Resolution (Qualitative: Analyzing the reactions and discrepancies between members in different configurations of work) <u>Roles in the team:</u> • Several roles in the team: - Conflict Resolution (Qualitative: Analyzing the reaction of members with different roles in disagreements) <u>Flexibility of Requirements:</u> • Flexible requirements: - Adaptability of decisions to change (Qualitative: Ability to adapt or convert decisions in the face of change)

Backlog management	General: - Backlog completion rate (Quantitative: Percentage of backlog items completed within a given time) - Backlog size (Quantitative: Size of the backlog given time) - Backlog item dating (Quantitative: Evaluate if items in the backlog are being worked on effectively and solved timely) - Lead time (Quantitative: Time between inserting an item in the backlog and completing it) - Sprint Analysis (Quantitative: Average work done within a sprint) Size of the team: • Big teams: - Backlog Transparency (Qualitative: Understanding of all team members of the backlog items) Roles in the team: • Several roles in the team - Backlog Agreement (Qualitative: Shared understanding between people of different roles in the team) Flexibility of requirements: • Flexible Requirements: - Backlog Accuracy (Qualitative)
Ticket management	General: - Sprint Efficiency (Quantitative: Tickets/Capacity estimated vs work done) - Ticket Resolution Time (Quantitative: Average time taken to resolve or close tickets) - Ticket Response Time (Quantitative: Time taken for new tickets to be responded) - Ticket Return Rate (Quantitative: Percentage of tickets resolved successfully without need for rework or additional follow-up) - Customer Satisfaction (Qualitative) Size of the team: • Big and medium teams: - Efficiency in ticket collaboration (Qualitative: Efficiency between collaborating members in tickets) Roles in the team: • Several roles in the team - Swiftness of sprint (Qualitative: Evaluate how efficient the sprint moves from the different phases between the different team roles) Flexibility of requirements: • Flexible requirements: - Ticket Escalation Rate (Quantitative: Percentage of tickets that have requirements, dependencies, or priority changes)

Team building	<u>General:</u> - Team engagement (Quantitative: Participation and satisfaction of team members in team-building activities). - Team satisfaction (Qualitative) - Employee Retention (Quantitative: Employee retention rates in teams) - Team dynamics, diversity, inclusion (Qualitative) - Knowledge Sharing (Qualitative) <u>Size of the team:</u> • Big and medium teams: - Assessment of trust and relationships (Quantitative: Feedback surveys) - Efficiency of the onboarding process (Qualitative and Quantitative, with feedback surveys and junior workers retention rate) <u>Configuration of work:</u> • Hybrid teams: - Connection between different configuration workers (Qualitative) • Fully remote teams: - Engagement and connection with team members (Qualitative) <u>Roles in the team:</u> • Several roles in the team: - Levels of connection between senior and junior workers (Qualitative)
Flexibility and changes	<u>General:</u> - Time-to-Market (Quantitative: Speed at which team delivers releases or new features) - Customer satisfaction (Qualitative) - Burndown chart (Quantitative: Comparing ideal expected progress versus work completed) - Defects due to changes (Quantitative: Defect rates and customer complaints and Qualitative: General assessment of quality of work affected by flexible requirements). - Client engagement (Qualitative: Evaluate the level of client engagement in the project regarding their needs and expectations). - Business Delivery (Qualitative: Assess the ability to keep sending out viable products or features valuable or relevant to the business, even in the face of changes).