Universidade do Minho
Escola de Engenharia

Danna Beatriz Rodrigues Dias

Implementing an agile project management methodology on a Minimum Viable Product development

November, 2022

Universidade do Minho
Escola de Engenharia

Danna Beatriz Rodrigues Dias

Implementing an agile project management methodology on a Minimum Viable Product development

Master Thesis
Master's in Engineering Project Management

Work developed under the guidance of:
Prof.ª Doutora Anabela Pereira Tereso
Prof.ª Ingrid Teixeira de Souza

November, 2022

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

ACKNOWLEDGMENTS

I have always dreamed of studying abroad, getting to know other cultures, new people, and new places while developing professionally. When I was a child, my dream looked distant and almost impossible. Growing old, I did not stop believing it. For that, I would like to thank my family, especially my mother and father, who always motivated me to dream big and supported all my decisions, even when they made us live in an ocean of distance. The "Saudade" is enormous.

I found in life a partner who dreams together with me. And without all his love and support, nothing of the last two years would be possible. Gustavo Moreno, thank you for your daily support, making this path easier, and for always holding my hand and telling me the words I need, to continue.

When I realized I would need to write a thesis project, I wanted more than anything to do it while applying it in a real-world environment, such as a company or a small project. One month before defining the project, I started working at MetaProvide, and even being just a new member, the organization relied on me and allowed me to implement an entirely new process within the organization. For that, I thank Björn Magalhaes for trusting in my work and giving me the autonomy to lead this project.

At MetaProvide, I found an environment I have never imagined working in. A fully remote team, working from several different countries with the same goal: to help people. An organization with no office but full of good hearts. I want to thank Marcela Muñoz for seeing my potential to be part of this team. I would also like to thank João Raposo for being such a good leader and helping me find my way. Thank you, Tiago Lamelas, Filipa Pinheiro, Igor Oliveira, Magnus Walbeck, and Henry Bergström, for supporting me during this project implementation on the development of Adminly and for helping me make it possible.

Finally, I would like to thank Professor Anabela Tereso for supporting my decision to implement an agile method at MetaProvide and for her time in orienting me through this project. And thank you, Ingrid Souza, for all the Sunday mornings you spent trying to help me with the decisions I could not make alone and for sharing your Kanban experience with me.

STATEMENT OF INTEGRITY

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

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

ABSTRACT

Implementing an agile project management methodology on a Minimum Viable Product development

In the past few years, the world has faced a rapid change in the digitalization of processes over several different economic sectors. But there are services where human contact is a part of their job, such as massagists, mindfulness guidance, yoga, therapy, etc. In those areas, digitalization and process automation are less important than human well-being.

This thesis project was developed in a not-for-profit Swedish startup that creates solutions, including services and products, to help well-being providers increase efficiency in their service. The startup has identified a lack of automation in the well-being services providers' administrative tasks, consequently wasting time they could use to help more people, with activities that do not add value to their business or society.

This master thesis presents the project of developing a Minimum Viable Product of a solution to help well-being providers with their administrative tasks. This thesis project focus on selecting and implementing an agile methodology to support the software development of the MVP, with a restricted time and budget.

The research aims to answer the question: "Which agile method or framework best suits the studied startup environment, considering the restricted time and budget to develop an MVP and why?". To achieve its objectives, the research method used to support the selection, implementation, improvement, and adaptation of the agile methodology was action research, considering the organization's needs, to support the delivery of a functional product that added value to the customer.

The chosen methodology was Kanban, and through its improvement cycles it was possible, with the team's help, to define a structure that would help them achieve their goals. It was noted that the methodology promoted excellent communication between the teams and aligned the customer's needs and the software development process. The project saved 14.5% of the budget and ended on time, considering the six months of planned development.

KEYWORDS

Agile, Project Management, Software Development, Minimum Viable Product

RESUMO

Implementação de metodologias ágeis de gestão de projetos no desenvolvimento de um Produto Viável Mínimo

Nos últimos anos, o mundo enfrentou uma rápida mudança na digitalização de processos em diversos setores económicos. Mas há serviços em que o contato humano faz parte de seu trabalho, como massagistas, orientação de meditação, ioga, terapia, etc. Nessas áreas, a digitalização e a automação de processos são menos importantes que o bem-estar humano.

Este projeto de dissertação foi desenvolvido numa *startup* Sueca sem fins lucrativos que cria soluções, incluindo serviços e produtos, para ajudar os profissionais da área do bem-estar a aumentar a eficiência no seu serviço. Essa *startup* identificou uma falta de automação nas tarefas administrativas dos prestadores de serviços de bem-estar, que consequentemente os faz perder tempo que poderiam usar para ajudar mais pessoas, em atividades que não agregam valor ao seu negócio ou à sociedade.

Esta dissertação apresenta o projeto de desenvolvimento de um Produto Mínimo Viável de uma solução para auxiliar os prestadores de bem-estar nas suas tarefas administrativas. Este projeto de mestrado foca na seleção e implementação de uma metodologia ágil para apoiar o desenvolvimento de software do MVP, num contexto em que o tempo e orçamento são restritos. A investigação procurou responder à pergunta: "Qual método ou *framework* ágil que se adequa melhor ao ambiente de *startup* apresentado, considerando o tempo e orçamento restritos para desenvolver um MVP e porquê?". Para atingir os seus objetivos, o método de investigação utilizado para apoiar a seleção, implementação, melhoria e adaptação da metodologia ágil foi investigação-ação, considerando as necessidades da organização, para apoiar a entrega de um produto funcional que agregasse valor ao cliente.

A metodologia escolhida foi o *Kanban*, e através de seus ciclos de melhoria foi possível, com a ajuda da equipa, definir uma estrutura que os ajudasse a atingir os seus objetivos. Notou-se que a metodologia promoveu uma excelente comunicação entre as equipas e alinhou as necessidades do cliente com o processo de desenvolvimento de software. O projeto economizou 14,5% do orçamento e terminou no prazo, considerando os seis meses de desenvolvimento planeado.

PALAVRAS-CHAVE

Ágil, Gestão de Projetos, Desenvolvimento de Software, Produto Viável Mínimo.

INDEX

FIGURE INDEX

TABLE INDEX

LIST OF ABBREVIATIONS AND ACRONYMS

AGM – Annual General Meeting

AR – Action Research

DevOps – Software Development and IT Operations

HR – Human Resources

IPMA – International Project Management Association

MVP – Minimum Viable Product

PD – Product Development team

PHP – Hypertext Preprocessor (programming language)

PMBOK – Project Management Body of Knowledge

PMI – Project Management Institute

PMP – Project Management Professional

PR – Public Relations

RACI Matrix – Responsibility Assignment Matrix (Responsible, Accountable, Consulted, Informed)

R&R – Roles and Responsibilities

SDLC – System Development Life Cycle

STATIK - Systems Thinking Approach To Introducing Kanban

TPS – Toyota Production System TPS

UI – User Interface

UX – User Experience

VA – Virtual Assistant

WIP – Work in Progress

XP – Extreme Programming

1. INTRODUCTION

This chapter presents context and motivation, research question and objectives, research methodology, and thesis structure.

1.1 Context and Motivation

The world has been facing a pandemic in the last two years, making everyone rethink how we work. Nowadays, it is even more necessary to be online and connected than two years ago (Fabris, 2022). According to McKinsey Global Survey (2020), many companies have accommodated working remotely, and the responses to the pandemic have speeded up the adoption of digital technologies for several years. With that, the market could supply the demand of being more online and reduce the risk of an infection from a deadly disease. It was easier to change in some areas, such as information technology, because remote work was already a reality. But there are services where human contact is a part of their job, such as massages, mindfulness guidance, yoga, therapy, etc. In those areas, digitalization and process automation are less important than human well-being.

MetaProvide, the company where this thesis project was developed, is a not-for-profit Swedish startup, developing solutions, including services and products, to help well-being providers increase efficiency in their service. MetaProvide has identified a lack of automation in the well-being services providers' administrative tasks, consequently wasting time they could use to help more people with activities that do not add value to their business or society.

The author of this thesis works at MetaProvide as an Agile Project Manager and plays a key role in ensuring that projects are delivered on time, on budget, and on scope, and that all stakeholders have easy access to project information.

In 2019, MetaProvide started working to understand better the providers' pain points and the problems they have been experiencing in their business, compromising their time and task management, which could be invested in quality time with their relatives or increasing the number of clients they have. The organization's mission is "help others to help more" and it "seeks to connect anyone searching for, or providing, wellbeing and insight" (metaprovide.org, Jan 09, 2022).

The goals at MetaProvide are divided into phases, namely:

- Phase 0 – The organization identified an opportunity of helping well-being providers with their administrative tasks;

- Phase 1 – Adminly Minimum Viable Product (MVP) Concept development. MetaProvide defined the problem they would solve, the personas, the concept of MVP, and the solution prototypes to be built on the next phase;

- Phase 2 – MVP or beta version development;

- Phase 3 – Adminly market launch and platform improvement.

The organization finished its first phase in January 2022, when the Product Team defined the Minimum Viable Product (MVP) concept. For the subsequent two phases, their solution's beta version (or MVP) would be developed, called Adminly, a platform that will support the administrative tasks of well-being providers, such as yoga and meditation teachers.

The minimum viable product concept comes from the startup environment and the product development literature. "The foundation for the development of an MVP is to collect feedback on a product to be put on a market from various stakeholders" (Lee and Geum, 2021, p. 2). Lee and Geum (2021) also mentioned that customer feedback on an MVP is crucial to avoid spending extra time and resources building less wanted product attributes. Conforto et al. (2014, p. 24) present the importance of "obtaining continuous feedback from the customer to respond to constant changes in requirements, needs, risks, new opportunities, and so forth".

The startup environment is complex and uncertain, being tough to define the whole scope before the project starts, as it is traditionally done in project management (PMI, 2017). The traditional approach "assumes that events are predictable and that all tools and techniques are well understood" (Bergmann and Karwowski, 2019, p. 406). While the agile approach, according to Zasa et al. (2021), aims to release different subparts of the output as soon as possible, at the end of each iteration, to collect feedback from the main project stakeholders.

The project management challenge presented by MetaProvide is the development of an MVP in a startup context to be delivered in six months with the budget constraint of 400,000 euros, preserving the organization's open-source values and respecting data privacy. Considering all the mentioned aspects, the project requires an agile response to the technology development and constant validation with potential clients (Reis, 2011; Bergmann and Karwowski, 2019; Dingsøyr et al., 2012).

Another point is that the organization has implemented, in the last phase, a Kanban board to manage the tasks visually and some Scrum events, such as Sprint and Retrospective meetings, to create an agile space of decision-making within the team and to promote flexibility in the management of the deliverables. The Scrum events and Kanban Board do not represent a full implementation of either the Kanban Method or Scrum Framework. Still, until now, it has been working correctly for its purposes. The organization is open to agile methodologies, but the current work management is not enough for software development.

There are some agile methodologies and frameworks that could be useful to help manage this project. Kanban, Extreme Programming, Scrum, and Cristal are some of them (Schwaber and Sutherland, 2020; Anderson, 2010; Beck, 2000; Boehm, 2007). Scrum is the most famous agile methodology in software development, followed by Kanban (Shastri et al., 2021; Digital.ai, 2021). Every agile method, like any other, has its pros and cons, so it is crucial to understand the environment, the project team, and project characteristics, to implement the method that is more likely to be accepted by the people involved and to fit into the organizational culture.

1.2 Objectives and Research Question

The main research objective was to implement an agile methodology to guarantee that the Minimum Viable Product of Adminly could be delivered on time and within budget for phase two, starting in February and ending in August 2022. To do so, the specific objectives are:

- Study the most efficient agile methodologies used for software development and their results;
- Define the methodology that most accommodates the team and project particularities;
- Implement the methodology;
- Define data to be collected and studied during the project;
- Adapt the methodology to attend the project best;
- Analyze the impact of changes;
- Define the next steps for the organization's project management.

As a result, it aimed to understand the company characteristics and their impact on the methodology implementation. Another expected outcome is measuring the implementation's success, based on the project management metrics and results. How will the team respond to the chosen methodology events, communication protocols, productivity metrics, and so on? Considering that, the following research question was defined: "Which agile method or framework best suits the studied startup environment, considering the restricted time and budget to develop an MVP and why?".

1.3 Research Methodology

After defining the research question and objectives, the next step was selecting a research methodology that guided the study. Saunders et al. (2016) is a relevant reference in terms of research methods for business students, and his approach will be the one followed throughout this thesis.

Considering the Research Onion of Saunders et al. (2016), presented in Figure 1, the research comprises a philosophy, an approach, a strategy, choices, time zones, techniques, and procedures. For this master's project, the philosophical approach is Pragmatism. The reason is that the investigation begins with a problem and aims to contribute to practical solutions that can be considered in future practices (Saunders et al., 2016).

The deduction was used as the approach for theory development, as the project was supported by a literature review that guided the selection of the methodology to be implemented. According to Saunders et al. (2016), the deductive approach uses data to prove or contest theories and hypotheses from literature reviews or other conditions that support the theory.

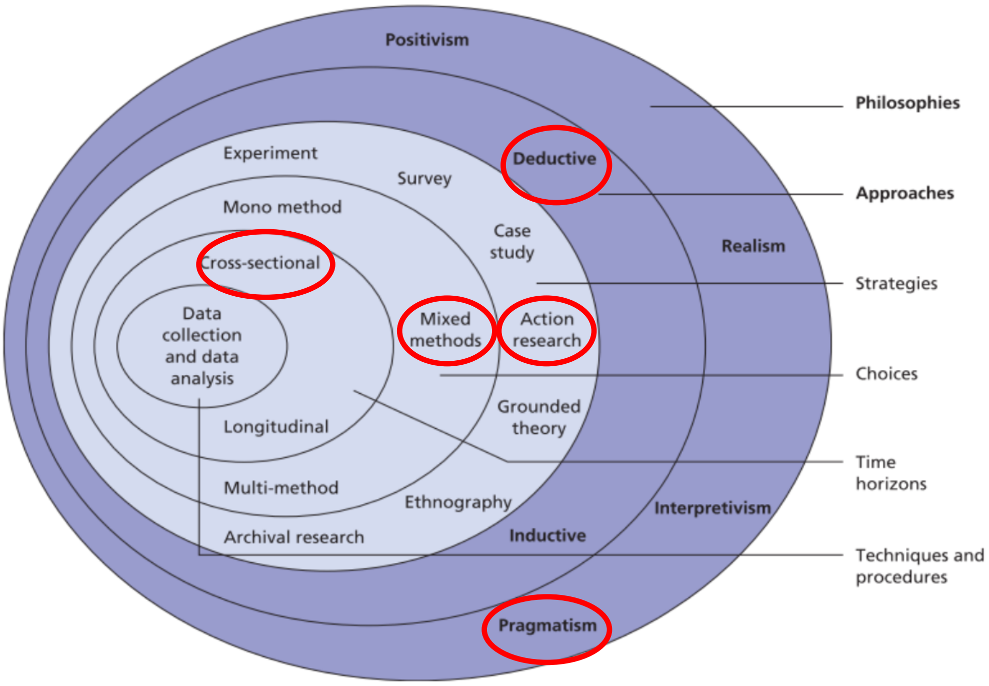


Figure 1 - Research Onion
(Saunders et al., 2016)

The author of this thesis was part of the team responsible for selecting and implementing the methodology for managing the MVP project. For this project, the research strategy chosen was Action Research (AR). The main reason for this selection was that, according to Coughlan and Coghlan (2002), action research is participative. A group of people allocated in the studied context is part of the research and not only the object of it.

Coughlan and Coghlan (2002, p. 225) also defend that action research is about change. "AR applies to the understanding, planning, and implementing change in business firms and other organizations". This statement matches precisely with the proposal of this research, to plan and implement a method that somehow changed the way of working of a team. This thesis aims to have, as an outcome, the result of these implementations and the impact of it regarding the goals previously defined.

According to Saunders et al. (2016) and as presented in Figure 2, the action research suggests a cycle dynamic. The research starts with a context and a purpose, followed by a research question. After that, the cycles are composed of diagnosing, planning, acting, and evaluating action. With the last cycle evaluation results, a new process, made of the same steps, starts.

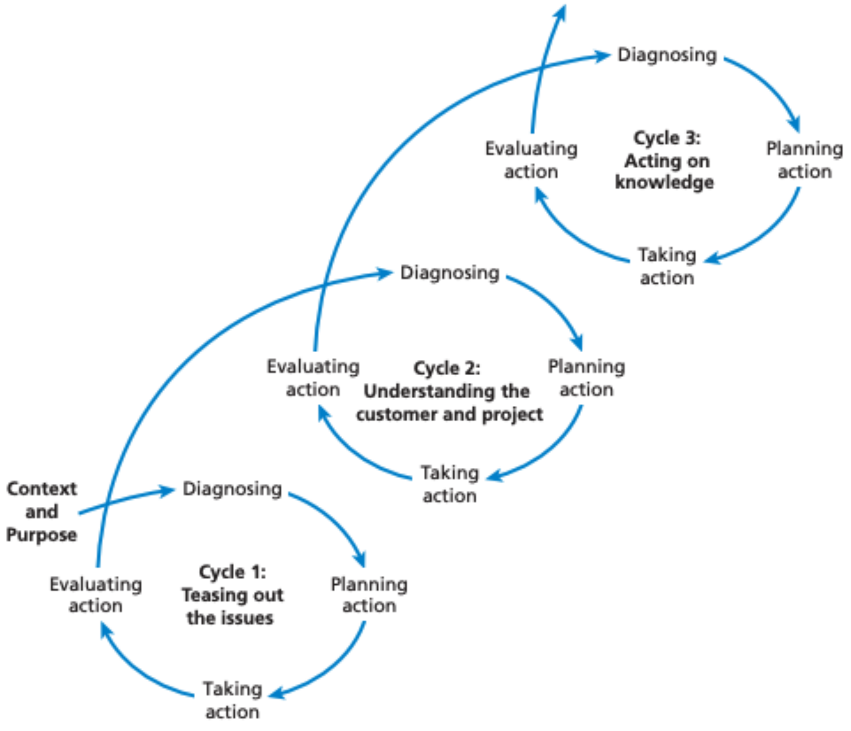


Figure 2 - Action Research Cycles
(Saunders et al., 2016)

The results of each action research cycle were validated by the researcher and the team working on developing the project to decide the study's next steps. It was crucial to the project that the studied team understands the importance of the applied project management methodology and collaborates for its success. It was also relevant to consider what can be implemented in the studied context and what is not valid to implement.

The study is cross-sectional because there was a time limit of six months to conduct the research (Saunders et al., 2016).

Throughout the action research cycles, qualitative and quantitative data were collected using the mixed method. The qualitative data considered in this research was the literature review, the information discussed during the meetings with the group, and the feedback the team gave, which supported the evaluation and improvements on the proposed method to be implemented. The literature review was the input for the action plan for the first cycle of the action research, supporting the diagnosis and defining the purpose of the study. It also helped determine what methodology better fit the organization context, given what it was possible to find in the academic literature.

The second cycle of action research and the following were supported by cycles of validation and feedback within the studied team, to understand the impact of the action on the project's success. The result of each cycle was the input for the next one, adapting things, when necessary, to guarantee the success of the agile project management methodology implementation for the MVP creation.

For the quantitative data, the study analyzed the budget and the established date of delivery to determine if the chosen project management methodology helped the team deliver the project on time and within budget. The study also analyzed some metrics regarding the work produced over time.

1.4 Thesis Structure

This master's thesis is structured into six chapters to present the project of implementing an agile methodology for developing an MVP.

The first chapter is the Introduction, composed of the motivation for the research, the objectives, the chosen research methodology, and how it will be applied to this thesis.

The second chapter presents the literature review that supports this thesis project. The literature review is fundamental to the project's development as it helped the decision of which agile method would be implemented.

The third chapter is about the context of the organization that this project takes place. It presents the reality of the organization, the teams, the work structure, the Adminly development context, and the project prototypes.

The fourth chapter presents the implementation of the chosen methodology. It explains how the methodology was defined, the planning process, and the execution of the implementation itself.

The fifth chapter regards improvements and the impact of the method implementation on the team. It will also present if the objectives of the research were achieved or not and its results.

The sixth and last chapter, it presents the conclusions, the lessons learned, and limitations, as well as the proposals for future work.

2. LITERATURE REVIEW

This chapter presents the literature review that supports this thesis. It will start with the concepts of Project Management and how it began, the evolution of Project Management for Software Development. It will also present the basic concepts of Lean and Agile. Ultimately, it will advance with the main Agile Methodologies, such as Scrum, Extreme Programming, Crystal Methodology, and Kanban.

2.1 Project Management

Projects have been part of the world since the beginning of what we know about society. The construction of houses, bridges, cities, monuments, or even wars to conquer new cities and civilizations that happened thousands of years ago can be considered projects (Garel, 2013; Shenhar and Dvir, 2007). Considering that according to PMI (2017), a project is an effort undertaken by a group of people in a defined space of time to create a unique result, which can be a service or a product.

According to Garel (2013), project management started to be studied and became a field of research only in the second half of the 20th century. In 1965, International Project Management Association (IPMA) was born, followed by the PMI (Project Management Institute) in 1969. Both organizations are responsible for creating standards and certifications related to the best practices for project managers (Bredillet et al., 2015).

Nowadays, the PMBOK (Project Management Body of Knowledge) published by the PMI, along with their PMP certification, are the most widespread and consolidated references for the ones that want to become project managers (Sánchez-Arias and Solarte-Pazos, 2010).

The PMBOK (2017) presents the lifecycle of a predictive project with four phases: starting the process, organizing and preparing, carrying out the work, and ending the project, as presented in Figure 3 below. Five process groups support the phases of a project: initiating processes, planning processes, executing processes, monitoring, and controlling processes, and closing processes. Besides that, the PMBOK (2017) divides the project into ten areas of knowledge, all covered by the five processes, as shown in Figure 3 below.

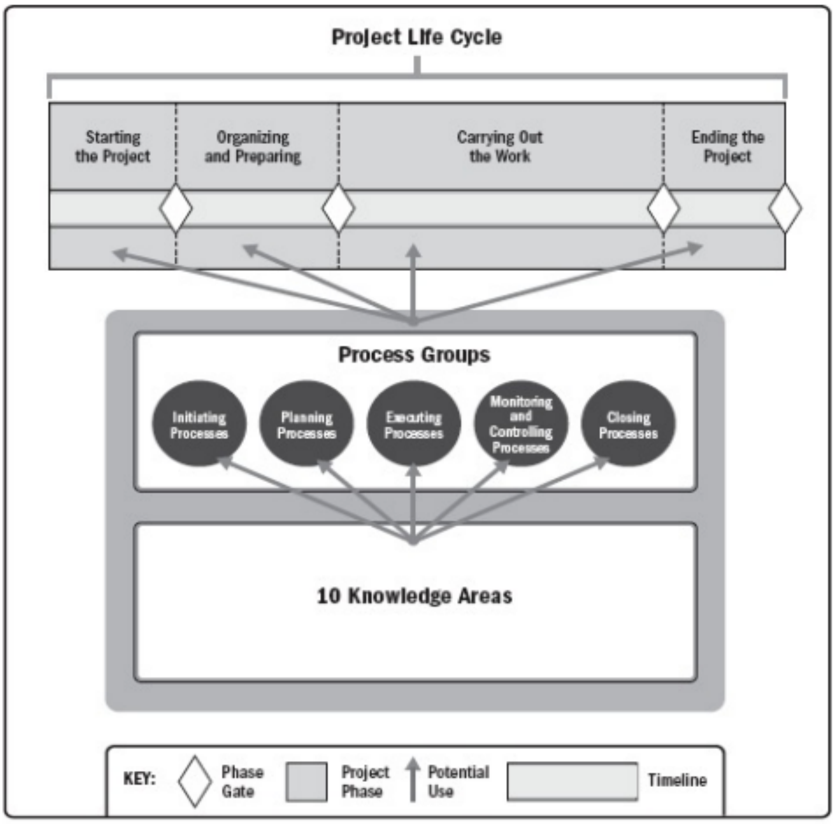


Figure 3 - Interrelationship of PMBOK Guide Key Components in Projects (PMBOK, 2017)

The ten knowledge areas are (PMBOK, 2017):

- Project Integration Management
  - The processes and activities needed to integrate the various processes and project management activities within the Project Management Processes Group.
- Project Scope Management
  - Define the project deliverables and ensure that the project includes all work required to be completed successfully.
- Project Schedule Management
  - The management of the time for the completion of the project.
- Project Cost Management
  - Budget management in a way to guarantees that the project will be delivered as planned with success. It involves planning, financing, budgeting, managing, and controlling costs.

- Project Quality Management
  - Meet the stakeholder's expectations and includes the process of incorporating the organization's quality policies regarding planning, managing, and controlling project and product quality requirements.
- Project Resources Management
  - Responsible for guiding the identification, acquisition, and management process of the resources needed to complete the project.
- Project Communications Management
  - Includes the processes required to ensure timely and appropriate communication involving planning, collection, distribution, management storage, control, monitoring, and ultimate disposition of the project information.
- Project Risk Management
  - Includes the processes of risk management planning, identification, analysis, response planning, and implementation, as well as the monitoring of risk on a project.
- Project Procurement Management
  - Includes the purchase and acquisition processes of services, results, or products from outside the project team.
- Project Stakeholder Management
  - Includes the processes of identifying everyone (people, organizations, or groups) that could impact or be impacted by the project and managing their expectations and engagement in the decisions and executions of the project.

A project comprises deliverables and succeeds when the stakeholders are satisfied with its results. Three significant factors impact a project's success: quality, cost, and time, as presented in Figure 4 below (Yuliadi and Nugroho, 2019; Todorović et al., 2015). If one of them was not planned correctly or is not well balanced during the execution of the project, it will impact the other two. For example, if a project is running overtime, at some point, it will affect the project's cost and/or quality, and so on.

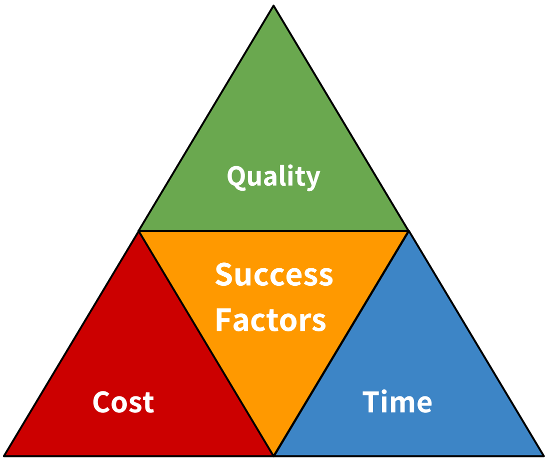


Figure 4 - Triangle of Project Management Success
(Yuliadi and Nugroho, 2019)

As time and costs are limited resources in organizations, those are the critical factors for most projects. But Yuliadi and Nugroho (2019) presented data showing that other factors impact a project's success, such as sharing information across the teams and visibility into what the team and others are working on, both issues related to communication. Reich, Gemino, and Sauer (2010) mentioned the importance of creating value for the business as a primary goal for a project manager, which can also be considered a success factor.

The standard presented by PMBOK is considered the traditional project management model (Bergmann and Karwawski, 2019), and it has been changed from the 6th to the 7th edition (PMI, 2021). Figure 5 below presents the migration from the 6th to the 7th edition of PMBOK (PMI, 2021), the 6th from 2017, and the 7th from 2021. In 2022, both editions are considered references to the PMI standards, as it is in the migration process.

The new version of PMBOK has changed the ten areas of Knowledge Areas to eight project performance domains (PMI, 2021). This evolution presented in PMBOK follows the world's change regarding project management methodologies.

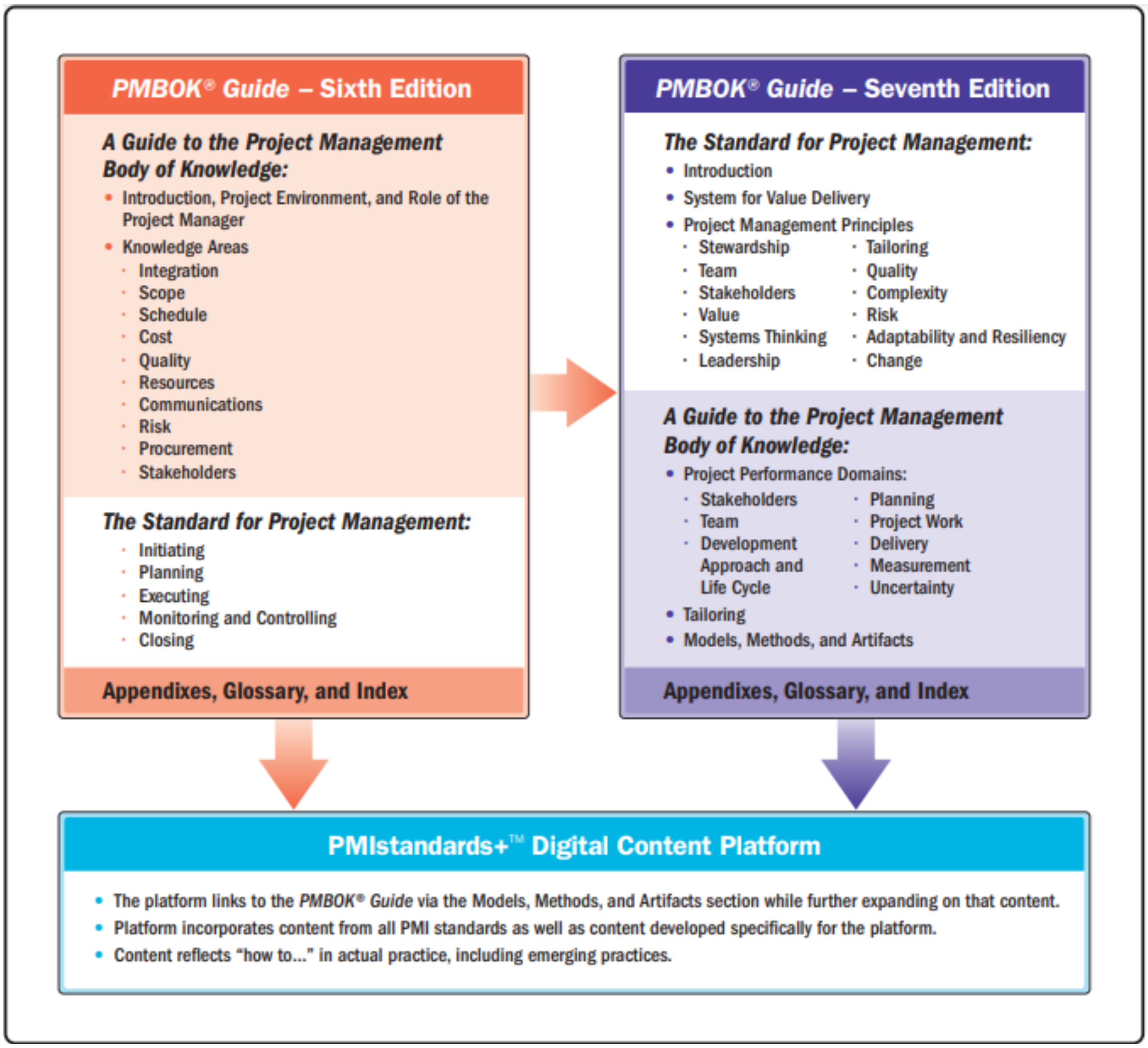

PMBOK® Guide – Sixth Edition
A Guide to the Project Management Body of Knowledge:
Introduction, Project Environment, and Role of the Project Manager
Knowledge Areas
Integration
Scope
Schedule
Cost
Quality
Resources
Communications
Risk
Procurement
Stakeholders
The Standard for Project Management:
Initiating
Planning
Executing
Monitoring and Controlling
Closing
Appendixes, Glossary, and Index
PMBOK® Guide – Seventh Edition
The Standard for Project Management:
Introduction
System for Value Delivery
Project Management Principles
Stewardship
Tailoring
Team
Quality
Stakeholders
Complexity
Value
Risk
Systems Thinking
Adaptability and Resiliency
Leadership
Change
A Guide to the Project Management Body of Knowledge:
Project Performance Domains:
Stakeholders
Planning
Team
Project Work
Development Approach and Life Cycle
Delivery
Measurement
Uncertainty
Tailoring
Models, Methods, and Artifacts
Appendixes, Glossary, and Index
PMIstandards+™ Digital Content Platform
The platform links to the PMBOK® Guide via the Models, Methods, and Artifacts section while further expanding on that content.
Platform incorporates content from all PMI standards as well as content developed specifically for the platform.
Content reflects "how to..." in actual practice, including emerging practices.

Figure 5 - PMBOK 7th *edition* - migration from the 6th
(PMI, 2021)

According to Bergmann and Karwawski (2019), traditional project management, such as the model presented in the PMBOK 6th edition (PMI, 2017), is a well-structured process and focuses its importance on the requirements, which assumes that the events of the project are predictable and present limitations. It is difficult for the clients to define the beginning of the project, and all the requirements and deliverables for the project, especially if it is in a dynamic and complex environment, such as a startup and software development industry (Sánchez-Arias and Solarte-Pazos, 2010; Eisenmann, 2021; PMI, 2021).

2.2 Project Management for Software Development

In 1970, Royce developed a traditional model for managing software development called the Waterfall model (Kurman and Bhatia, 2014). The Waterfall is the oldest and most well-known approach to a System Development Life Cycle (SDLC), a sequence of steps that must be followed to develop software or a system (Alshamrani and Bahattab, 2015). The stages of an SDLC include requirement, design, coding, testing, deployment, and maintenance (Hlaing and Ochimizu, 2018).

The waterfall model requires that the team knows most of the project requirements, solutions, and goals, not supporting significant changes in its scope (Lei, et al. 2015). It can be advantageous for some projects, as it is well-documented and well-structured, but a disadvantage because it is rigid and does not support changes after the contract between the parties is signed.

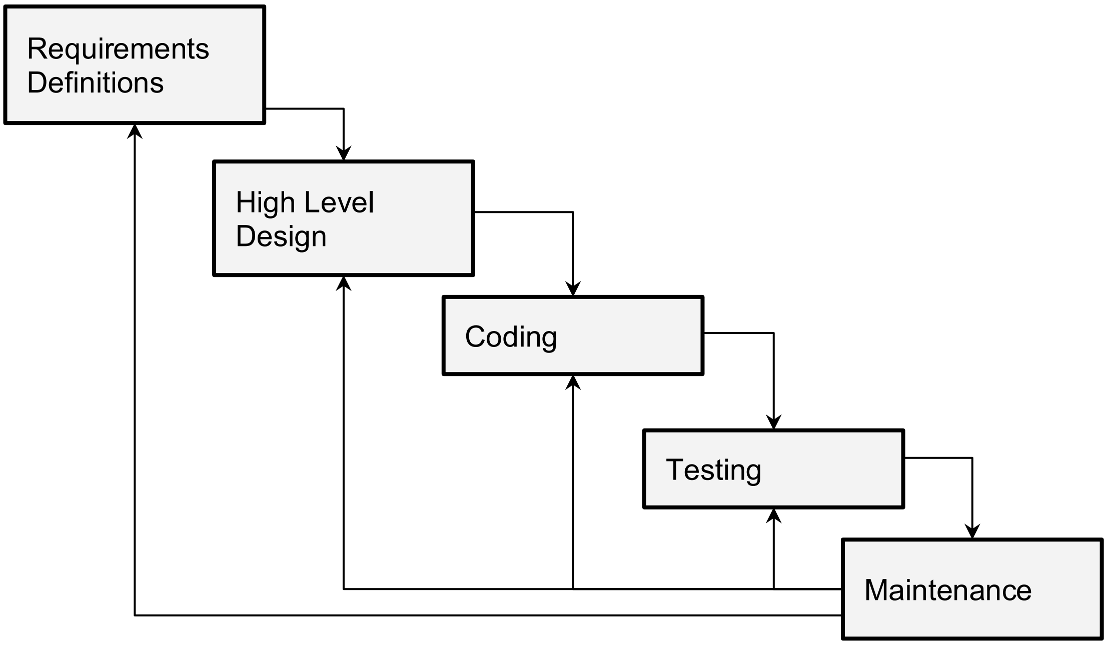


Figure 6 - Waterfall Model
(Alshamrani and Bahattab, 2015)

Alshamrani and Bahattab (2015), as shown in Figure 6, present the phases of the Waterfall as:

- Requirements Definitions: A description of the behavior of the system to be developed. It is a moment to collect client information and agree on the software specifications and requirements;

- High-Level Design: At this moment, the information from the last phase is evaluated, and it is the process of planning and designing the problem-solving for a software solution, including architecture, database, and so on;
- Coding: The requirements will be converted to the production environment during this phase;
- Testing: In this phase, testing is executed to check if the software solution works correctly and if it meets the original requirements;
- Maintenance: This phase happens after the software is released, and at this point, it might need some improvements, error corrections, and modifications. This phase may impact from design to the release.

Waterfall has this name because it looks like a cascade, as presented in Figure 6 above, and one activity can only start when the previous has been finished in a sequence. In real projects, it is rare to have a sequential order. Kumar and Bhatia (2014) highlight that this model has a low presence of the customer in the process. It has no accommodation for changes, does not scale up to large projects, and is difficult to manage risks.

Besides the traditional waterfall, Hijazi et al. (2012) present other models of SDLC for software development, such as spiral, which is risk-driven, and evolutionary, which is code-driven, and classify them, such as conventional (waterfall) and agile (scrum). He also defends that no methodology fits every project of software development, so the software development context must be considered.

In the last three decades, the world has been facing a boom in the development of new technologies and the advanced software development industry, presenting a drastic change in its business processes and operations (Chadli. et al., 2016). Saynisch (2010) explains that traditional project management is a limited model to manage projects in a society that does not develop predictably, considering its people, markets, and organizations, but unpredictably. He also argues that the traditional project management in the PMBOK works "mainly on a mechanical, monocausal, nondynamic, linear structure and a discrete view of human nature and societies and their perception, knowledge, and actions".

This change in society and the growth of complexity and dynamicity in the software development area led the market to the emergence of new software development methodologies, called agile methodologies. The agile method is closer to the client, allowing constant evaluation and adjusting if the user needs it. Also, it is more adaptive, enabling it to work in rapid interactive planning and development cycles (Bergmann & Karwowski, 2019; Dingsøyr et al., 2012).

2.3 Lean

Kim (2014) presents the concept of lean as a systemic method of eliminating waste in a manufacturing process. The principles and values of lean started in Japan, in the 1940s, after the war, which was a critical time for Japan, with limited access to resources. At this moment, Toyota developed a revolutionary system to optimize the use of resources and reduce costs, the Toyota Production System (TPS). The TPS defended that only small fractions of a process are responsible for delivering value to the client, but only in the 1990's the term lean started to be used in books and articles related to the Toyota Production System (Melton, 2005; Sugimori et al. 1977; Womack et al., 1990).

According to Sugimori et al. (1977), the TPS initially focused on two basic concepts: reducing cost through waste reduction and creating a system that allows workers to display their full capabilities by themselves. To achieve the two goals, Toyota has developed the kanban system, signaling the product with cards as it moves through the process. It is a way of visualizing the waste in the flow, such as overtime and line-stop and connecting all the internal and external processes related to the managed production system (Sugimori et al. 1977).

Ohno (2019) suggests the complete elimination of wastes in the productive process and classifies them based on their origin, which are seven types:

1. Overproduction: When the industry produces more than necessary;
2. Waiting: Line of products waiting for the next step of the process;
3. Inventory: Unprocessed inventory;
4. Motion: Unnecessary movement of people in the production line;
5. Transportation: Excess of product movements from one place to another in the production line;
6. Over-processing: steps in the process that do not add value to the product.

Besides the importance of reducing waste in the production line, Womack (1990) also presents the possibility of applying lean thinking in any organization. He also explains the benefits of lean thinking, such as doing more with less, once it focuses on reducing waste and increasing customer satisfaction. Five concepts guide Lean thinking (Womack, 1990):

- Value: Identity precisely what is valuable for the end customer;

- Value stream: A process map that identifies and categorizes the steps of the process based on the value it adds, considering three perspectives: the problem solution, the information management, and the raw material transformation. The tasks that do not add value to the flow must be eliminated;
- Flow: The goal is to have a continuous flow in the production chain, which means that it does not present interruptions;
- Pull: The end customer pulls the production. It only starts after the order to avoid unnecessary work;
- Perfection: Continuous improvement in the process, product, and people to identify and remove waste, striving for perfection.

Lean thinking and principles have also been applied to the software development process, known as Lean Software Development (Poppendieck and Poppendieck, 2003). Besides, lean approaches have also been involved in Agile Software Development (Wang et al., 2012). Wang et al. (2012) present that some lean principles have facilitated agile implementation, such as kanban boards, eliminating waste, value stream mapping, root-cause analysis, etc.

2.4 Agile Methodologies

The Agile Methodologies for managing projects, mainly related to software development, were born and influenced by the changes implemented in the manufacturing industry at the end of the last century, led by the Lean Production of Toyota (Poppendieck and Poppendieck, 2003; Wang et al., 2012; Browaeys and Fisser, 2012). This topic presents the agile manifesto, the methodologies: Scrum, Extreme Programming, Crystal and Kanban.

1. Agile Manifesto

A compilation of agile can be found in the Agile Manifesto. The Agile Manifesto was born in 2001, when a group of seventeen software development leaders gathered forces and formalized the agile movement, stating four values and twelve principles that would be the basis for the movement (Beck et al., 2001). The values honored by the group are:

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

According to Fowler and Highsmith (2001), in each bullet point, the first part of the sentence indicates a preference, while the second part is also important but has less priority.
The twelve principles defended by the Agile Manifesto are (Beck et al., 2001):

- Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software;
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage;
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale;
- The business team and developers must work together daily throughout the project;
- Build projects around motivated individuals, give them the environment, and support they need, and trust them to get the job done;
- A face-to-face conversation is the most efficient and effective method of conveying information to and within a development team;
- Working software is the primary measure of progress;
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely;
- Continuous attention to technical excellence and good design enhances agility;
- Simplicity --the art of maximizing the amount of work not done-- is essential;
- The best architecture, requirements, and designs emerge from self-organizing teams;
- The team regularly reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Fowler and Highsmith (2001) state that one of the big motivators for creating the Agile Manifesto was to make a "safe" environment for software developers. He also mentions that the manifesto promises a better way of developing software by following the original values and principles.

2. Scrum

Scrum is an agile framework for project management. It can be used in any project, as we can see the application in other industries (Bin et al., 2021; Parsons et al., 2018; Garcia et al., 2022), but it has its origin related to software development. Scrum is the most famous and used agile framework in software development projects, and it is used by around 66% of the organizations that implemented agile methodology (Digital.ai, 2021). The values of Scrum are commitment, focus, openness, respect, and courage (Schwaber and Sutherland, 2020).

According to the Scrum Guide, the framework has a flat organization, which means that all the roles related to the framework have the same relevance, and there is no hierarchy between them and no subteams. Schwaber and Sutherland (2020) define it as "a cohesive unit of professionals focused on one objective at a time, the Product Goal”. The framework suggests a small team of 10 to fewer people (Schwaber and Sutherland, 2020). As an agile methodology and part of the motivation for the Agile Manifesto creation, Scrum prioritizes people and interactions over tools and documentation (Agile Manifesto, 2001).

The roles in the Scrum are (Garcia et al., 2022; Schwaber and Sutherland, 2020; Holtzhausen and Klerk, 2018; Morandini et al., 2021):

- Scrum Master: responsible for guaranteeing the proper use of the methodology, working as a coach for the team regarding the skills and scrum events. This role is also responsible for understanding the project needs and facilitating communication between the company and the team. Another important activity is helping the development team to remove impediments from their tasks;
- Product Owner: responsible for understanding the customer needs and communicating them with the development team, transforming them into requirements for the product/software. The product owner has the mission of defining and communicating the product goal to the team;
- Development Team: responsible for executing the project development, transforming the requirements developed by the Product Owner into an increment of value for the customer at the end of the sprint. The team has the autonomy to decide how they organize their work.

Scrum also has well-defined time-boxed events to organize the team's interactions and the tasks' execution. Scrum work is divided into cycles, in which part of the product is developed and delivered to the customer. The main point is to provide an increment value to the customer after every cycle. This cycle is called a sprint, and each sprint has a specific goal. The timebox for a sprint is from two to four weeks. In the sprint, there are the events, those are (Schwaber and Sutherland, 2020; Hron and Obwegeser, 2022; Morandini et al., 2021):

- Sprint Planning: This event happens at the beginning of each sprint to plan the deliverables for the sprint. During the sprint planning, the team needs to define a sprint goal and the sprint backlog. All the members are involved. It has a timebox of a maximum of eight hours for a one-month Sprint;

- Daily Meeting: The development team meets every day to understand what happened the day before, what is coming next, and if there is any impede their activities. It usually takes place at the same time and local every day to reduce complexity. The timebox for this meeting is a maximum of 15 minutes;
- Sprint Review: The sprint review is a meeting that happens at the end of the sprint to analyze the outcome of the sprint, and it is a moment that also involves the participation of key stakeholders. In this meeting, they can understand if the sprint goals were met and discuss the future of the product. The timebox for this event is a maximum of four hours;
- Sprint Retrospective: The sprint retrospective is the last event of the sprint. It is a moment to analyze how the last Sprint went regarding people, processes, tools, and interactions. It is also a moment where the team reflects on how they acted when they encountered problems during the spring. The scrum team looks for ways to improve its effectiveness and implement impactful changes as soon as possible. The timebox for this event is a maximum of three hours for a one-month Sprint.

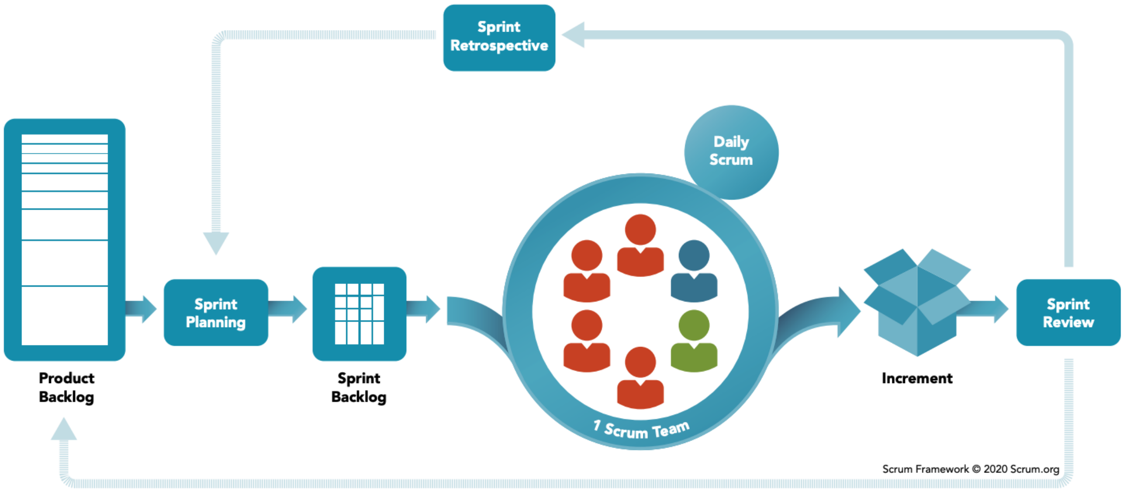


Figure 7 - Scrum Framework
(scrum.org, 2020)

Figure 7 above shows an example of the scrum framework, from the Product Backlog on the left side to the sprint review on the right side. This picture represents the Sprints in cycles. The Daily Scrum happens daily within a Sprint, and the Sprint occurs every one to four weeks, as defined by the team. It also shows that the team should deliver an Increment for the client at the end of each Sprint. The feedback and continuous improvement at the end of each sprint happen during the Sprint Review and Sprint Retrospective.

3. Extreme Programming (XP)

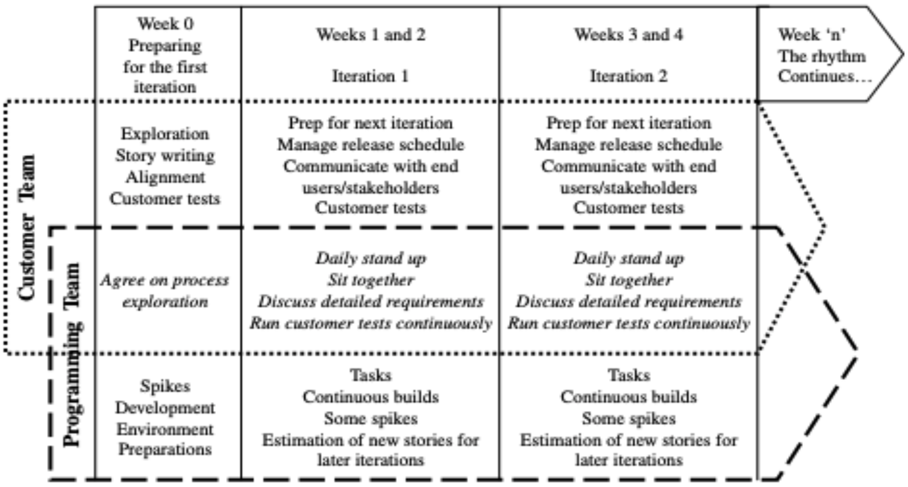
Extreme Programming is an agile software development method that does not point out a specific tool but presents fundamental values and best practices for the software development process (Angioni et al., 2006). Angioni et al. (2006) explain that XP is extreme because it uses some of the best practices of traditional development and carries it to an extreme level. For example, regarding communication, XP presents extreme communication to avoid misunderstanding during daily meetings and other moments to share the vision, project requirements, and status.

According to Beck (2000), XP has communication, feedback, simplicity, courage, and respect as its values. He also presents XP as "a path of improvement to excellence for people coming together to develop software”. The XP practices are (Beck, 1999):

- Planning game: The first step of XP is the planning game, in which the customer decides the scope and timing of releases;
- Small releases: In only a few months, the system is put into production, and after that, minor releases are made on a daily to monthly basis;
- Metaphor: Programmers and customers define a metaphor or a set of metaphors for the shape of the system;
- Simple Design: The design should be as simple as possible, running all the tests, communicating everything that it is supposed to, with no code duplication, and having the fewest possible classes and methods;
- Tests: The system is tested by programmers and customers. Programmers run unit tests, and they must all run correctly. The customer writes functional tests for the story;
- Refactoring: Transformation of an existing design to promote the system's evolution;
- Pair programming: The code in production is written by two people simultaneously, sharing screens, keyboard, mouse, etc;
- Continuous integration: The system is constantly updated with new code. To integrate a new code, the team builds the system from scratch. All tests must run, and the changes must be discarded;
- Collective ownership: Every programmer has the ownership of changing the code throughout the system if they find an opportunity for improvement;
- On-site customer: The customer is part of the team;

- 40-hour week: A programmer working overtime is considered a severe problem. No one can work overtime a second week consecutively;
- Open Space: The team needs a large room and a space for pair programming in the center of the area;
- Just rules: When teams decide to use Extreme Programming, they agree to follow the rules. But they are just rules. If at some point, the team wants to change anything, they are allowed to do so if they assess the effects of the change;

According to Lindstrom and Jeffries (2003), an XP Project lasts two weeks, in which the whole team works to deliver small releases that offer value to the client. In Figure 8 below, they present the initial interactions of a project and the main activities of customers and programmers' teams and this initial phase. In this image, it is possible to see the constant alignment between programmers and customers to guarantee the success of the project, one of the agile values (Beck et al., 2001).



	Week 0 Preparing for the first iteration	Weeks 1 and 2 Iteration 1	Weeks 3 and 4 Iteration 2	Week 'n' The rhythm Continues…
Customer Team	Exploration Story writing Alignment Customer tests	Prep for next iteration Manage release schedule Communicate with end users/stakeholders Customer tests	Prep for next iteration Manage release schedule Communicate with end users/stakeholders Customer tests	
Customer Team / Programming Team	*Agree on process exploration*	*Daily stand up* *Sit together* *Discuss detailed requirements* *Run customer tests continuously*	*Daily stand up* *Sit together* *Discuss detailed requirements* *Run customer tests continuously*	
Programming Team	Spikes Development Environment Preparations	Tasks Continuous builds Some spikes Estimation of new stories for later iterations	Tasks Continuous builds Some spikes Estimation of new stories for later iterations	

Figure 8 - The Rhythm of an XP Project
(Lindstrom and Jeffries, 2003)

4. Crystal

According to Anwer et al. (2017), Crystal is a group of agile software management methodologies created by Cockburn in the 1990s. It can be used in different projects with distinct levels of complexity. The levels of crystal methods are defined by a color, from crystal clear for small projects to red for large projects, as shown in Figure 9 below, depending on the number of participants and the importance of the project (Zhong et al., 2011).

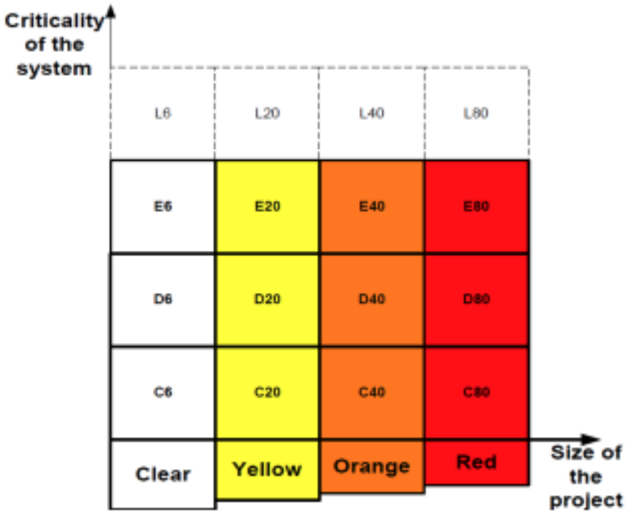


Figure 9 - The Family of Crystal Methods
(Anwer et al., 2017)

The Crystal family is more flexible than the other agile methodologies because it focuses more on the people than on the processes and tools (Highsmith, 2002). Boehm (2007) explains that Crystal has only two rules. First, the incremental cycles cannot be longer than four months; second, to self-adapt the methodology, it is crucial to have reflection workshops after every delivery. According to Boehm (2007), only Crystal Clear, represented by the white area in Figure 9, and Crystal Orange, represented by the orange area in Figure 9, have been well defined. The author of this thesis created a Table based on Boehm (2007) to summarize the two Crystal models.

Table 1 - Crystal Family Models
(Boehm, 2007)

Model	Crystal Clear	Crystal Orange
Number of People	• Small teams, 1 to 6 people	• Medium-size teams - 20 to 40 people
Documents and artifacts	• Release Plan; • Schedule of Reviews; • Informal/ Low-Ceremony; • Use Cases; • Design Sketches; • Running Code; • Common Object Model; • Test Cases; • and User Manual.	• Requirements Document; • Release Plan; • Schedule; • Status Reports; • UI Design Document; • Inter-Team Specs; • Running Code; • Common Object; • Model; • Test Cases; • Migration Code;

Model	Crystal Clear	Crystal Orange
		• and User Manual.
Roles	• Project Sponsor/customer; • Senior Designer Programmer; • Designer Programmer; • and User (part-time at least).	• Project Sponsor; • Business Expert; • Usage Expert; • Technical Facilitator; • Business Analyst; • Project Manager; • Architect; • Design Mentor; • Lead Designer Programmer; • UI Designer; • Reuse Point; • Writer and Tester.
Processes	• Incremental delivery; • Releases less than two to three months; • Some automated testing; • Direct user involvement; • Two user reviews per release and methodology tuning retrospectives; • Progress is tracked by software delivered or major decisions reached, not by documents completed.	• Incremental delivery; • Releases less than three to four months; • Some automated testing; • Direct user involvement; • Two user reviews per release and methodology tuning retrospectives.

5. Kanban

Anderson (2010, p.31) describes the Kanban as “Kanban (capital K) an evolutionary change method that utilizes a kanban (small k) pull system, visualization, and other tools to catalyze the introduction of Lean ideas. The process is evolutionary and incremental”. Anderson (2010) adopted the capital K to differentiate the kanban, the pull system created by Toyota, from the Kanban method.

According to Anderson (2010), Kanban is a complex adaptive system for lean and not an approach to managing software development lifecycle projects, which differentiates it from Scrum (Hijazi et al., 2012). The reason for that, explains Anderson (2010), is that to implement Kanban, it is required to have some process in place, so that it can be incrementally changed by the Kanban method. He also affirms that teams should not implement a defined method or a process template, but they "might be empowered to evolve their unique process solution that obviates the need for such services and requires a new set of tools" (Anderson, 2010, p. 44). He also states that the Kanban method has five properties:

1. Visualize workflow;
2. Limit work-in-progress (WIP);
3. Measure and manage the flow;
4. Make process policies explicit;
5. Use models to recognize improvement opportunities.

The benefits of the visual management of the Kanban board are also used by some teams using Scrum as the selected agile framework (Romano and Silva, 2015). Ahmad et al. (2018) studied the literature related to Kanban in software engineering and had, as a result, benefits associated with processes, people, and organizations, such as an increase in visibility, improvements in the workflow, faster time to market, and improved team building and cohesion.

Anderson and Carmichael (2016) explain how to introduce Kanban with the System Thinking Approach To Introducing Kanban (STATIK) in steps that are not necessarily sequential but interactive. The steps are, according to them:

Step 0: Identify services;

For each service

Step 1: Understand what makes the service fit for the purpose for the customer;

Step 2: Understand sources of dissatisfaction with the current system;

Step 3: Analyze the demand;

Step 4: Analyze the capability;

Step 5: Model workflow;

Step 6: Discover classes of services;

Step 7: Design the Kanban system;

Step 8: Socialize the system and board design and negotiate implementation.

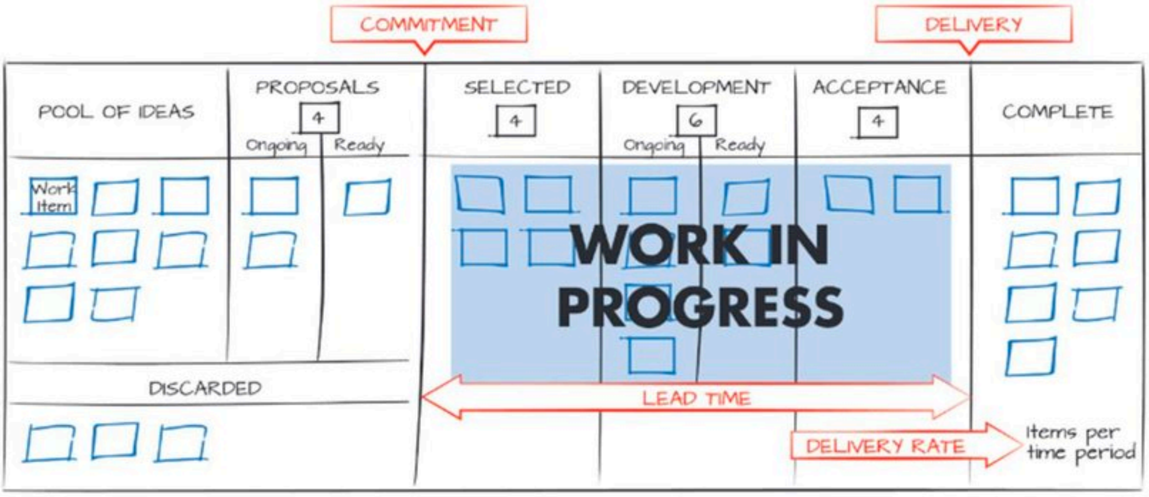


Figure 10 - Kanban Board
(Anderson and Carmichael, 2016)

Figure 10 above is an example of a Kanban board, which presents some important concepts defended by the Kanban method. The top of the picture, in red, shows the commitment and delivery points. The commitment point, according to Anderson and Carmichael (2016), is the point that represents an agreement between the customer and the service. From this point on, the customer makes an order, and the service will produce and deliver it to the customer. Each column represents one activity within the service. In the center of the picture, it is possible to see the work in progress.

At the bottom of Figure 10, in red, they presented two critical metrics, the Lead Time and Delivery Rate. According to Anderson and Carmichael (2016), the lead time is the time an item takes from the commitment point to the delivery point. Nakazawa and Tanaka (2016) point out the importance of limiting the WIP to shorten the lead time so that the service can be delivered faster to the customer. The Delivery Rate, also known as Throughput, is the rate of delivered items, and it is calculated by "dividing the number of deliveries by the length of the time period" (Anderson and Carmichael, 2016).

For implementing the Kanban, Anderson and Carmichael (2016) defend that the organization should start with what they do now, and it is not necessary to create new roles or job titles. However, two roles have emerged from the practice of Kanban, but they reinforce that the purpose of the role is essential, rather than assigning someone specifically with a job title. The two functions are (Anderson and Carmichael, 2016):

- Service Request Manager: communicates directly with the customers, understanding their needs and expectations. This person is also responsible for selecting and prioritizing the work items at the Replenishment Meeting;
- Service Delivery Manager: facilitates the Kanban Meetings and Delivery Meetings and is responsible for the workflow in delivering items to the customer.

In the structure of meetings, Anderson and Carmichael (2016) consider the Replenishment Meeting and the Kanban Meeting a baseline for the Kanban implementation. Other than that, additional meetings are used to keep the feedback loop working in the Kanban method, such as (Anderson and Carmichael, 2016):

- Service Delivery Review: the purpose of this meeting is to examine and review the effectiveness of a service;
- Replenishment Meeting: a session to move items over the commitment point;
- Kanban Meeting: daily coordination for the people working to deliver the service. It is a moment of alignment, which uses the "stand-up" format to be short and energetic;
- Delivery Planning Meeting: monitor and plan deliveries to the customers.

3. CONTEXT AND PURPOSE

This chapter presents the organization's context, MetaProvide, history, goals, team, and work structure. Furthermore, it also shows the context of Adminly development, its idea, roadmap, mockups, and crew, to provide an overview of the project.

3.1 Organization context

MetaProvide was created in 2019 with a base of volunteers that worked to understand the well-being providers' problems and try to shape a potential solution. The organization's vision is to "reduce individual suffering through human-centered IT services that foster inner growth in every person we meet". MetaProvide has as a goal for 2022 and 2023 the development of two products, one focused on helping well-being providers' administrative tasks, named Adminly, and another one is a platform free of charge that aims to help explorers, people searching for growth or guidance, to find a vetted provider to help them in their exploring process.

In 2021, MetaProvide received a donation, and with that, it was possible to give workers remuneration and increase the number of people involved in the project. During Phase 1, which happened in the second semester of 2020, MetaProvide started a deep research process to understand the well-being market and develop the MVP concept to help them in their administrative tasks.

At the beginning of 2022, at the end of the first phase, MetaProvide received around 400,000 €, the budget for the six months of Adminly, the platform that will help well-being providers help more people. This donation made it possible to increase the team, and during Phase 2, MetaProvide aims to develop the Beta Version of Adminly, and for Phase 3, the goal is to have it public, as presented in Figure 11.

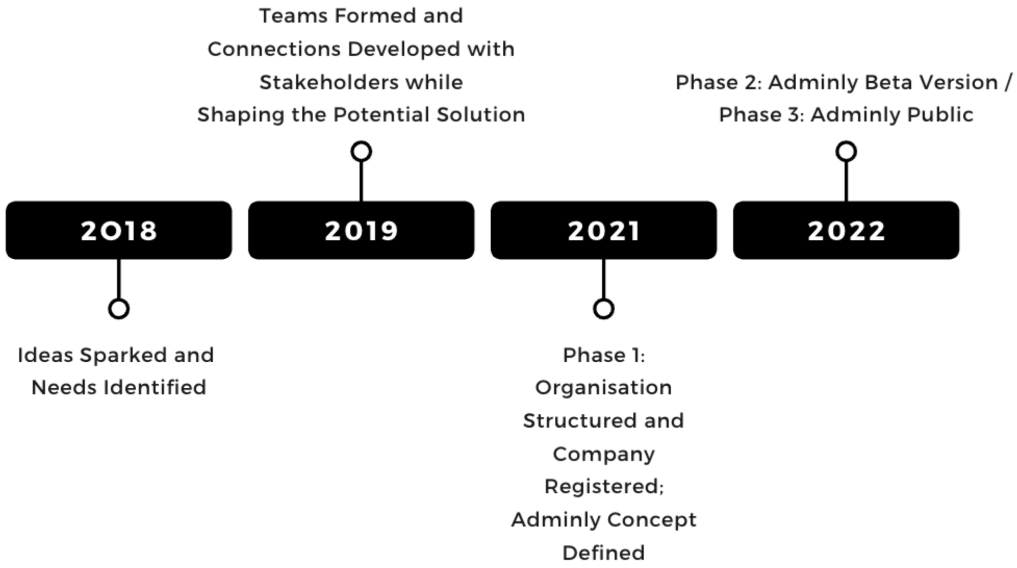
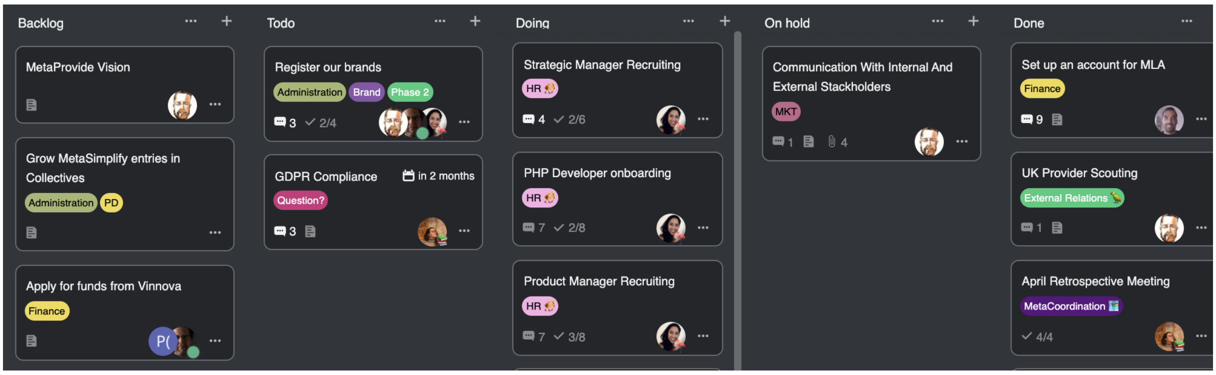

Teams Formed and Connections Developed with Stakeholders while Shaping the Potential Solution
Phase 2: Adminly Beta Version / Phase 3: Adminly Public
2018
2019
2021
2022
Ideas Sparked and Needs Identified
Phase 1: Organisation Structured and Company Registered; Adminly Concept Defined

Figure 11 - MetaProvide timeline - History and Goals

3.2 Communication and Work Management at MetaProvide

One of the core values of MetaProvide is data privacy protection. Based on that, MetaProvide uses NextCloud as a platform for communication, file storage, e-mail, work organization, and calendar. NextCloud is a self-hosted and open-source platform that has been improved and adapted to MetaProvide's needs.

Since its beginning, MetaProvide has chosen to use some agile principles as its way of working, either to manage the projects or the operations themselves. To support that, NextCloud offers a feature of visual management of the work called Deck, which is very similar to other products in the market, such as Trello, for creating kanban boards, making the visual management of the work possible. In the Deck, the work is divided into “Backlog”, “To do”, “Doing”, “On Hold”, and “Done”, as represented in Figure 12, as the status of the task.



Backlog

- MetaProvide Vision
- Grow MetaSimplify entries in Collectives — Administration, PD
- Apply for funds from Vinnova — Finance

Todo

- Register our brands — Administration, Brand, Phase 2 — 3 ✓ 2/4
- GDPR Compliance — in 2 months — Question? — 3

Doing

- Strategic Manager Recruiting — HR — 4 ✓ 2/6
- PHP Developer onboarding — HR — 7 ✓ 2/8
- Product Manager Recruiting — HR — 7 ✓ 3/8

On hold

- Communication With Internal And External Stackholders — MKT — 1 4

Done

- Set up an account for MLA — Finance — 9
- UK Provider Scouting — External Relations — 1
- April Retrospective Meeting — MetaCoordination — ✓ 4/4

Figure 12 - Kanban board on Deck – NextCloud

Another agile practice adopted by MetaProvide, since the beginning, was to adopt some agile events based on Scrum, such as sprint planning and retrospectives. Those events are part of the work activities, involving the whole organization and not only the people involved in product development.

A sprint meeting happens once a week, and it's a moment for the teams to share the status of their tasks toward the phase's goals. The retrospectives are a moment to self-reflect as a team, regarding a period on what the team is doing well and must continue and what the team is not doing so well and should improve. Action plans are created for the next month or so, in that meeting. Retrospectives happen every 90 work days.

One of the channels on Talk, the message feature of NextCloud is called Daily Standup, where the members of the organization share daily what they did the day before, what they plan for the day, and if there are any blockers on their tasks. The agile project manager is responsible for helping them unblock their impediments.

The events and the communication channel on Talk are not strictly implemented as Scrum suggests, and the work is not divided into sprints. The work management method is based on the best practices of Scrum and Kanban in a way accepted by the organization. These practices were implemented before the start of this project and have been continuously improved.

3.3 Team Context

MetaProvide is a flat organization, and the board makes the decisions. The board meets every three weeks, composed of three members and four deputies. Annually, there is a meeting called AGM (Annual General Meeting), in which the organization members vote on the operational plan for the year. This plan guides the organization's decisions and goals; every member has one equal vote on the topics discussed during AGM. Weekly, the organization members meet to discuss possible decisions the organization might take. At that moment, every individual that works for the organization can be heard, regarding the proposed changes. The topics that can impact the organization's structure, way of working, or budget needs a decision from the board. Three teams are responsible for the organization's function: the Product Team, the Operational Team, and the Tech Team. The Operational Team is responsible for supporting product development needs and guaranteeing the sustainability of the projects. As shown in Figure 13 below, the Operational Team comprises HR, Fundraising and Funds, Financial, Legal, Institutional, Branding, Business and Marketing, PR and External Relations.

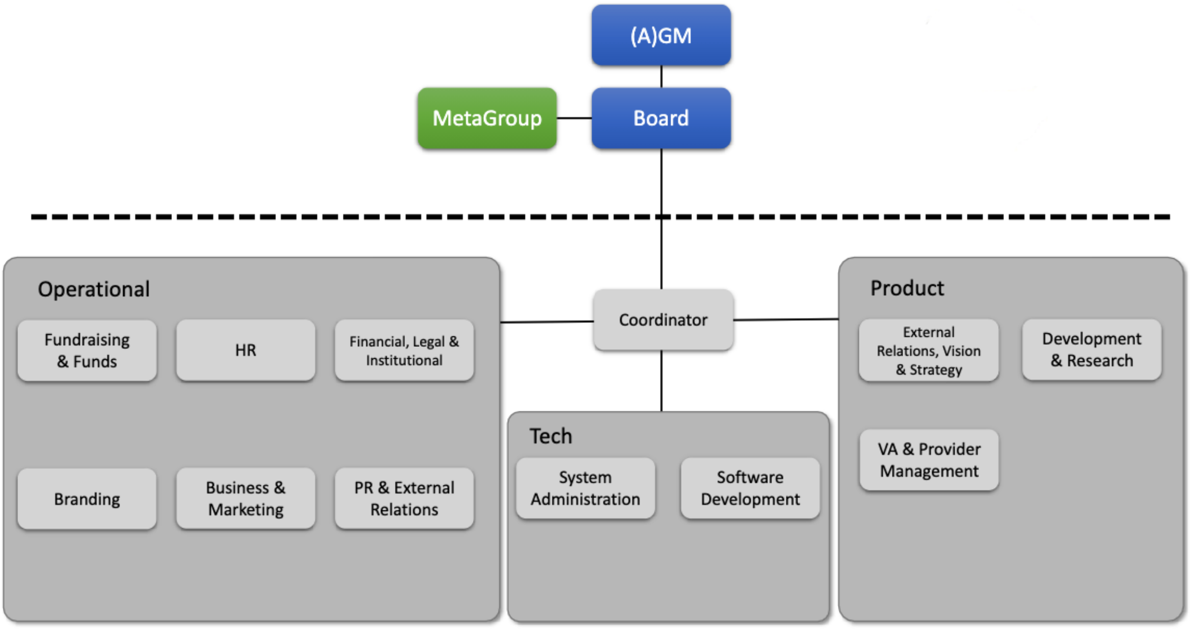


Figure 13 - MetaProvide's Structure

The Product Team is responsible for researching and developing the products for MetaProvide, which includes constant alignment with the organization's vision and strategy. The Product Team comprises External Relations, Vision and Strategy, Development and Research, and VA and Provider Management. The VA and Provider Management areas will be better explained in the Adminly Development Context.

The Tech Team is composed of System Administration and Software Development areas. The System Administration is responsible for improving and implementing the NextCloud for the organization, and the Software Development is responsible for developing the Adminly platform.

For the development of Adminly, the need to hire more people for the Tech and Product teams was identified, such as specialists in Software Development, User Interface, and User Research.

A bigger and better-structured squad, with a big goal of developing an innovative platform, raised the need to implement an agile project management methodology to support that.

3.4 Adminly Development Context

This topic presents the Adminly idea and the principals it was based on to be developed. Additionally, it also presents the project roadmap and the Adminly prototypes, the mockups showing how the platform will look like when finished, as well as the team that is responsible for the product development.

3.5 Idea and Principles of Adminly

MetaProvide started researching a way of helping providers of well-being services in 2019 when a group of volunteers identified the need for a solution to help well-being providers with their administrative tasks.

After interviewing a group of professionals, mainly focused on alternative therapies for body, mind, and soul, such as yoga teachers and meditation/mindfulness specialists, MetaProvide identified a lack of expertise in essential areas for entrepreneurs, such as accountability, management, and finances. Besides, most interviews showed that those professionals had no digital tools to integrate and manage payments, client information, calendar, etc.

It was considering that MetaProvide defined requirements related to the organization's values and principles and their potential customer's lifestyles. The first one is the human touch, as the organization is developing a solution to people that help and treat other people's pain. MetaProvide defined that the answer could not be only technology focused, but also it should involve people that could understand and relate to the provider's pain and reality. And with that, research around offering human assistance started, and Adminly will be a platform that will provide a human Virtual Assistant (VA) to help the providers manage their activities.

The next aspect was already mentioned: Data Privacy and Open Source. Since the Adminly development is focused on the NextCloud platform, the functionalities of booking and scheduling, client information, e-mail, assistance, and payments will be improved and adapted to the well-being providers' needs based on user research and interviews.

3.6 Adminly Roadmap

The main goal of Adminly is to have a functional MVP (Minimum Viable Product) by the end of Phase 2, which goes from February 21st, 2022, to August 16th, 2022, as presented in Figure

14. By June 16th, it is programmed to have the first release and start testing with the providers who support the research process of Adminly development.

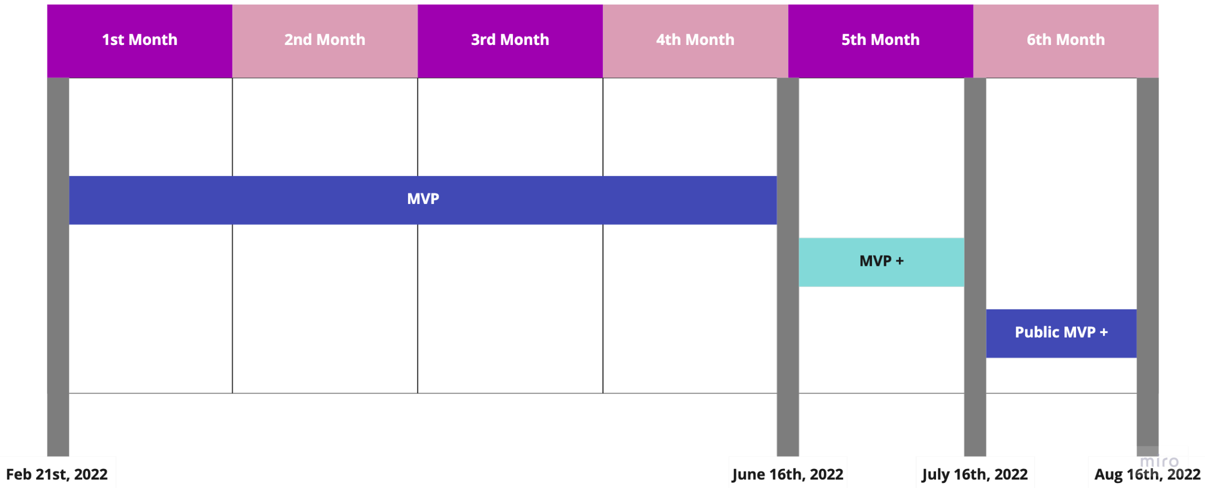


Figure 14 - Adminly Roadmap

After the initial six months of research, the Product Development Team identified that the primary need was a platform to centralize information and to help the providers control and manage their booking sessions. Additionally, the system aims to provide client information, payment information, and a contact portal between them and their virtual assistants.

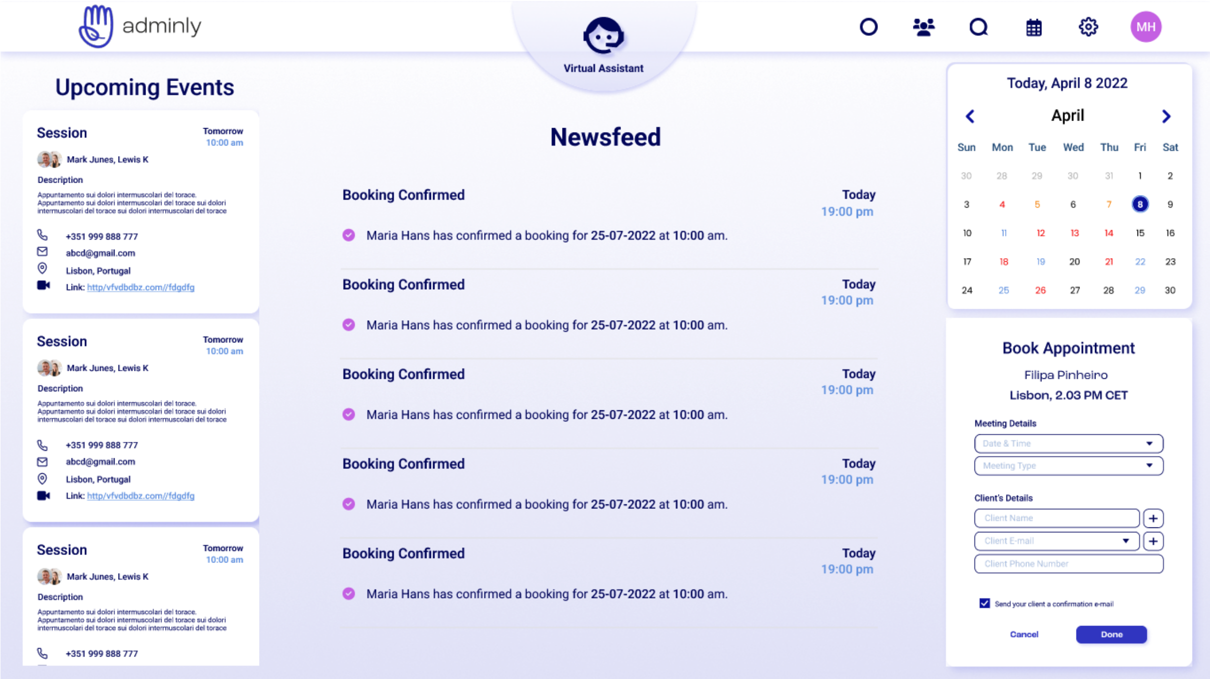
The Adminly features are the following:

- Bookings:
  - Calendar detailed view: calendar view for the month, week, or day with the sessions and events information;
  - Sessions' detailed information: by clicking on the sessions, it is possible to see the link for the meeting and client and payment information;
  - Schedule generator: a feature to define providers' availability for sessions;
  - Public link to book a session: a link where the client can access the provider's availability and schedule the meeting directly on their calendar.
- Assistant
  - A channel of communication between provider and assistant.
- Clients
  - Client's information: detailed information about clients, contacts, last sessions, future sessions, contracts, and payments;
  - List of clients: list of clients where it is possible to filter the kind of information shown.

- Payments
  - Payments information;
  - Payments status per client;
  - Charts with payments history and reports.
- Dashboard
  - Offer a view of the primary information for the provider, such as upcoming events, newsfeed, calendar, and so on.

3.7 Adminly Prototype

The following figures present the design of the Adminly platform made by the UI/UX (User Interface/User Experience) Designer of the project using the software Figma. The platform comprises a Dashboard (Figure 15), where the provider can have an overview of the upcoming events, generate an appointment, and access the newsfeed with the latest updates.



Upcoming Events

Session — Tomorrow 10:00 am
Mark Junes, Lewis K
Description
Appuntamento sui dolori intermuscolari del torace. Appuntamento sui dolori intermuscolari del torace sui dolori intermuscolari del torace sui dolori intermuscolari del torace
☎ +351 999 888 777
abcd@gmail.com
Lisbon, Portugal
Link: http/vfvdbdbz.com//fdgdfg

Session — Tomorrow 10:00 am
Mark Junes, Lewis K
Description
Appuntamento sui dolori intermuscolari del torace. Appuntamento sui dolori intermuscolari del torace sui dolori intermuscolari del torace sui dolori intermuscolari del torace
☎ +351 999 888 777
abcd@gmail.com
Lisbon, Portugal
Link: http/vfvdbdbz.com//fdgdfg

Session — Tomorrow 10:00 am
Mark Junes, Lewis K
Description
Appuntamento sui dolori intermuscolari del torace. Appuntamento sui dolori intermuscolari del torace sui dolori intermuscolari del torace sui dolori intermuscolari del torace
☎ +351 999 888 777

Newsfeed

Booking Confirmed — Today 19:00 pm
Maria Hans has confirmed a booking for 25-07-2022 at 10:00 am.

Booking Confirmed — Today 19:00 pm
Maria Hans has confirmed a booking for 25-07-2022 at 10:00 am.

Booking Confirmed — Today 19:00 pm
Maria Hans has confirmed a booking for 25-07-2022 at 10:00 am.

Booking Confirmed — Today 19:00 pm
Maria Hans has confirmed a booking for 25-07-2022 at 10:00 am.

Booking Confirmed — Today 19:00 pm
Maria Hans has confirmed a booking for 25-07-2022 at 10:00 am.

Today, April 8 2022

‹ April ›

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Book Appointment
Filipa Pinheiro
Lisbon, 2.03 PM CET

Meeting Details
Date & Time ▾
Meeting Type ▾

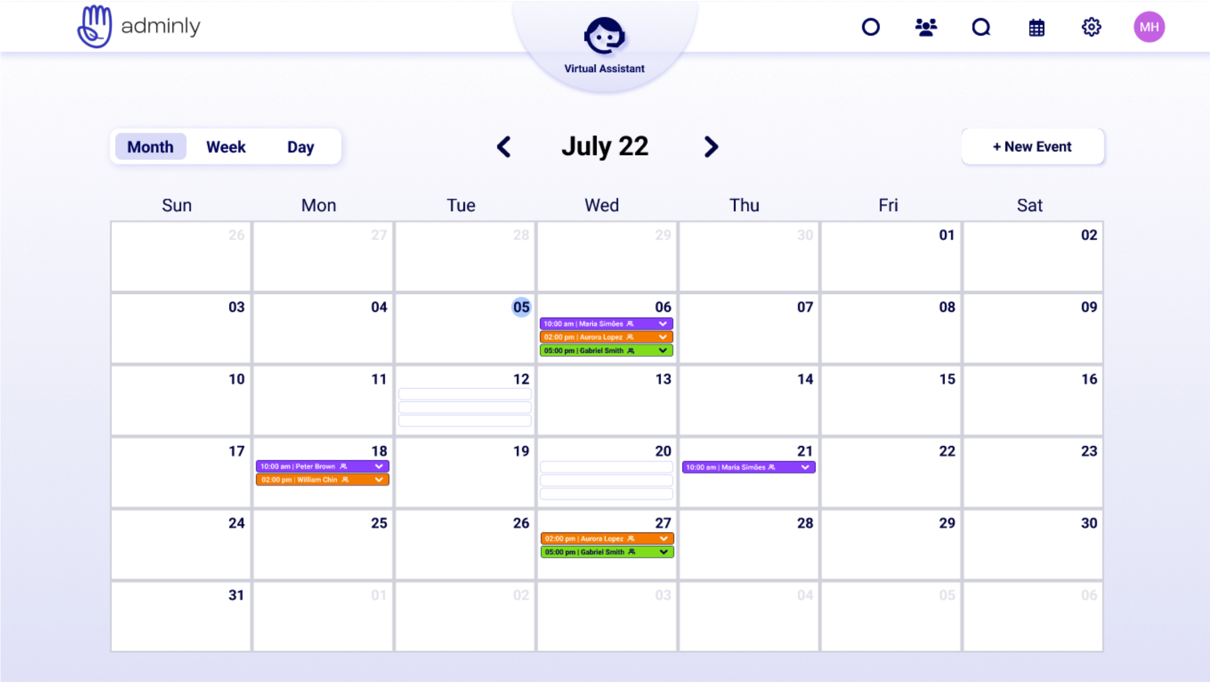
Client's Details
Client Name +
Client E-mail ▾ +
Client Phone Number

☑ Send your client a confirmation e-mail

Cancel | Done

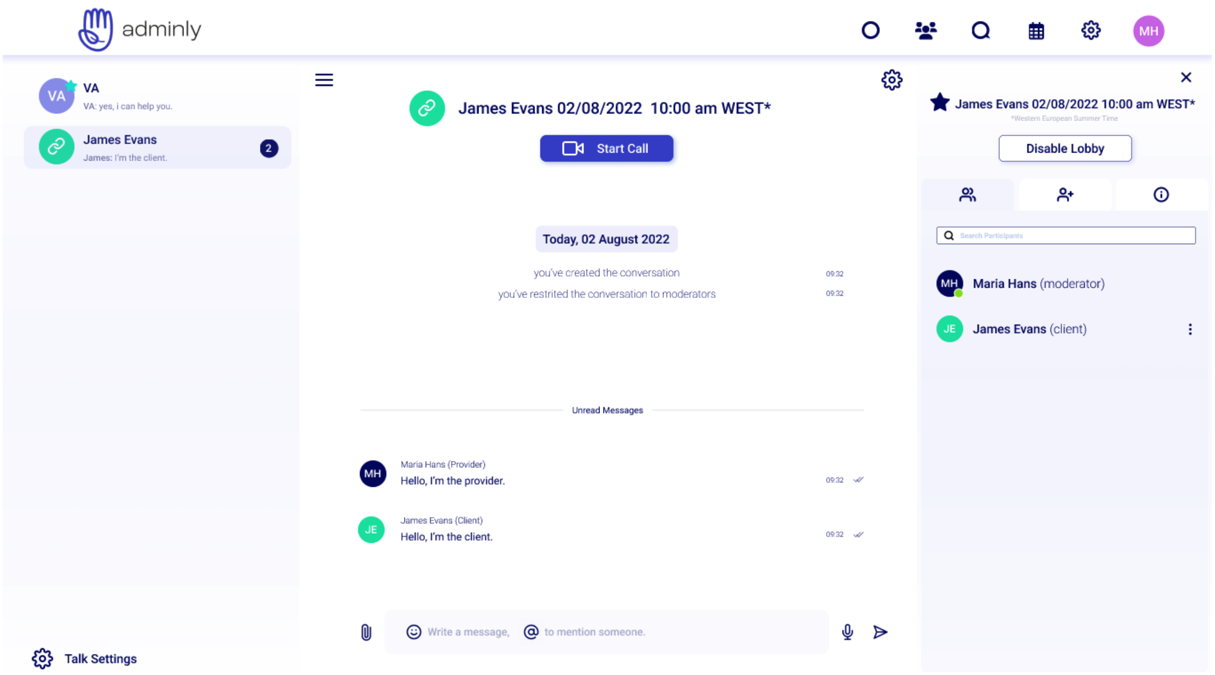
Figure 15 - Adminly Dashboard Prototype

A Calendar (Figure 16) presents a monthly, weekly, or daily view of the provider's schedule. The platform will also have a Chat (Figure 17), where the provider can contact their clients or their VA (Virtual Assistant), an assistant who works for them a few hours per month. The pack of work of a VA is defined based on the providers' subscription, and they help the provider with appointments, social media, and other administrative tasks.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	01	02
03	04	05	06 10:00 am | Maria Simões 02:00 pm | Aurora Lopez 05:00 pm | Gabriel Smith	07	08	09
10	11	12	13	14	15	16
17	18 10:00 am | Peter Brown 02:00 pm | William Chin	19	20	21 10:00 am | Maria Simões	22	23
24	25	26	27 02:00 pm | Aurora Lopez 05:00 pm | Gabriel Smith	28	29	30
31	01	02	03	04	05	06

Figure 16 - Adminly Calendar Prototype



VA
VA: yes, i can help you.

James Evans
James: I'm the client.

James Evans 02/08/2022 10:00 am WEST*

Today, 02 August 2022

you've created the conversation 09:32
you've restrited the conversation to moderators 09:32

Maria Hans (Provider)
Hello, I'm the provider. 09:32

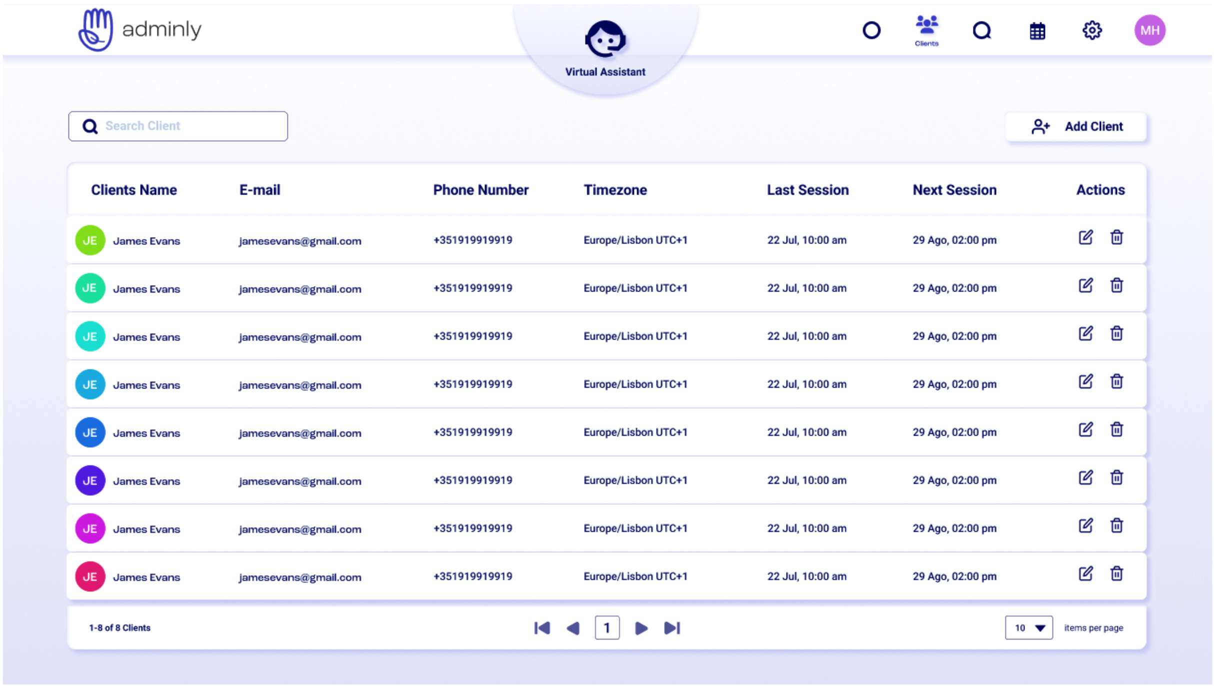
James Evans (Client)
Hello, I'm the client. 09:32

James Evans 02/08/2022 10:00 am WEST*
*Western European Summer Time

Maria Hans (moderator)
James Evans (client)

Figure 17 - Adminly Chat Prototype

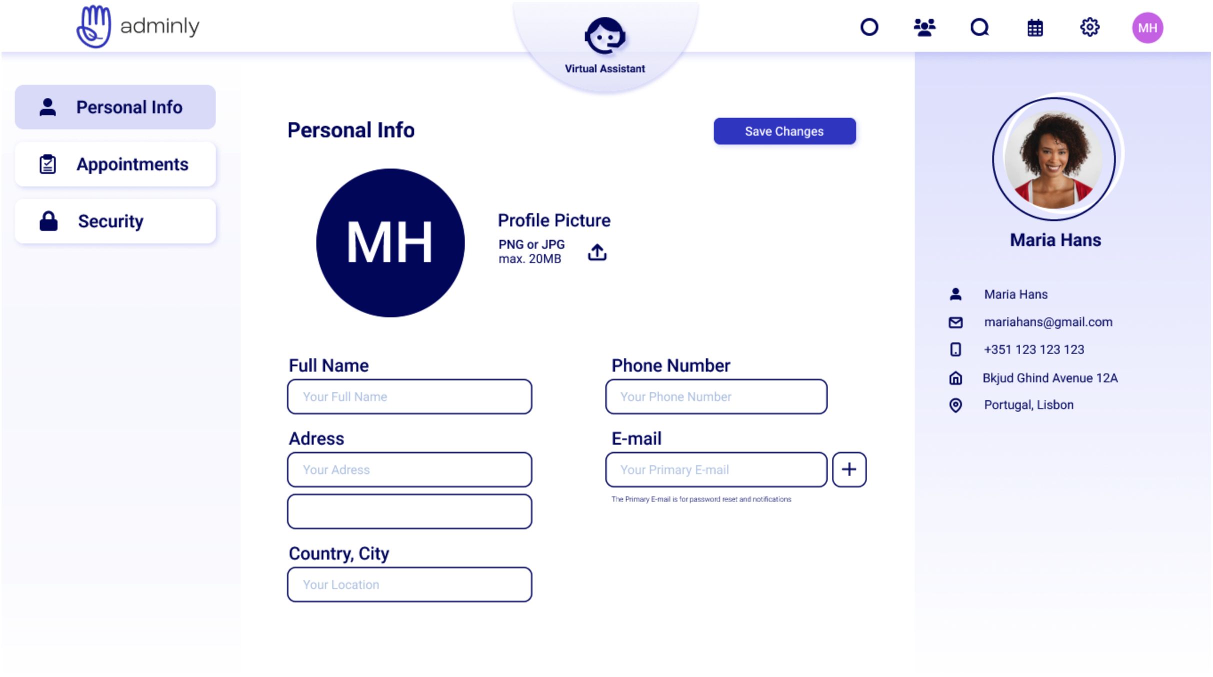
In the Clients' session (Figure 18), the provider can access the contact information of all their clients. And finally, the platform will have a Settings Page (Figure 19), allowing the provider to configure their profile.



Clients Name	E-mail	Phone Number	Timezone	Last Session	Next Session	Actions
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	
JE James Evans	jamesevans@gmail.com	+351919919919	Europe/Lisbon UTC+1	22 Jul, 10:00 am	29 Ago, 02:00 pm	

1-8 of 8 Clients   1   10 items per page

Figure 18 - Adminly Clients Prototype



Maria Hans
mariahans@gmail.com
+351 123 123 123
Bkjud Ghind Avenue 12A
Portugal, Lisbon

Figure 19 - Adminly Settings Prototype

3.8 Adminly Development Team

Two teams work developing Adminly, Product Development, and Tech. The process started in 2021 in the Product Development team, the one responsible for the discovery process, which allowed the organization to understand better the competitor, the market analysis, and the providers' needs and to get to a potential solution to solve the problem. In February 2022, the Tech team joined the development process responsible for developing the Adminly Platform.

Adminly is not only a technological platform but also a service. Along with the use of the platform, MetaProvide will sell Virtual Assistant (VA) hours, which means that the Adminly plans will go from the use of the software only, to the combination of VA hours with the platform, in different plans, depending on the providers' needs and the number of clients.

To develop that, the product team counts on the following:

- Product Leader, responsible for managing all the demands related to the project and communicating directly with the Tech team, regarding the platform requirements and design;
- Product Manager, responsible for managing the research project, testing the prototypes, and managing the VA's work;

- Vision and External Relations Manager, which is the founder of the organization, and owner of the idea of the project, and for contacting the external relations and bringing insights and people to help on the product development;
- VA and Providers Relationship is responsible for selecting and integrating providers in the research process, checking in with the VAs on how the process is going, and being accountable for interviewing providers;
- VA Manager: Responsible for the VA operations, coordination, creating, and supporting the VA Process;
- Product Marketing Manager: Responsible for product positioning and messaging, planning and executing go-to-market strategies, achieving product-market fit, and exploring, scoping, and building on content that helps create product positioning;
- User Interface and User Experience (UI/UX) Designer: Own and drive the end-to-end process for solving UI/UX design problems: ideate, conceptualize, design, test, and refine.

On the other hand, before the beginning of phase 2, the only responsibility of the Tech Team was the system administration, which was to support and improve the NextCloud platform for internal usage. At the time, the team was composed of two people with part-time commitments. After planning the project, it was identified the necessity of hiring two people to integrate the team, supporting both activities, system administration and Adminly software development, one of the two positions was already filled.

Regarding Adminly Development, the Tech team has two people focused on software development and one supporting the development and testing to ensure the technical quality of the platform, the Tech Leader. One position of PHP developer is open to keep the software development process.

4. Implementation - First Cycle

This chapter presents the process of planning and acting of the Action Research. As well as which one was the chosen agile methodology implemented within the project context and why. Furthermore, it also presents the implementation process of the method.

4.1 Methodology Selection

The research and literature review presented in Chapter 2 is part of the data collection and input for selecting the methodology to be implemented in the project. This project involves a Minimum Viable Product, it is related to software development, and it takes place in a startup environment, where the funds and time are very limited for the development, so it was defined that the traditional methodology is not an option. As Saynisch (2010) explained, the traditional project management method is a limited model for managing unpredictable projects.

The criteria for selecting the methodology were a combination of what was found during the research process as the most used and successful agile methodologies applied in software development projects and the analysis of the context in which the project takes place. The Agile Project Manager, also the author of this thesis project, presented two proposals to the Development team, including the Product and Tech teams. For discussion, both teams were heard, and their ideas and concerns were considered in the decision process.

As presented before, Scrum is the most famous agile methodology in software development, followed by Kanban (Shastri et al., 2021; Digital.ai, 2021). Based on that, the two chosen methods to be presented to the Teams and evaluated to decide which one would be better accepted, based on each’s characteristics, were Scrum and Kanban.

As Scrum is the most disseminated, it was the first to be considered. As presented before, the framework is well structured, with all the roles, time-boxed events, and a predetermined duration, going from two to four weeks (Schwaber and Sutherland, 2020). On the other hand, the Kanban Method is not a predefined process or a template, as stated by Anderson (2010). Still, he defends that the team "should be empowered to evolve their unique process solution that obviates the need for such services and require a new set of tools”. Anderson and Carmichael (2016) suggest a meeting structure to implement and determine a feedback loop in the system, such as Kanban Meeting and Replenishment. Still, their time and frequency are determined by the team.

Despite the Tech Team is tiny, being one full-time and two part-time developers, the time for developing the MVP was only six months. The project is based on an open-source tool, which reduces its complexity in terms of development and time. The chosen methodology was Kanban. The decision was taken in agreement with the teams.

Before this project, the Tech Team only worked in the support and system administration, so the development process will also be implemented along with the beginning of the MVP building. Considering that, it will not be possible to measure the difference between the implementation of Kanban and the previous work, as it did not exist. But the Kanban application comes in the way of answering the question: "Which agile method or framework best suits the studied startup environment, considering the restricted time and budget to develop an MVP and why?", being the delivery of a functional MVP on time and budget the success measure.

4.2 Planning the Kanban Implementation

The planning of the Kanban implementation was based on the literature review presented on the 5, and it starts with an analysis and modeling of the process. Then it was defined how the process would be transformed into a Kanban board and the system (roles, events, and metrics) that would support the methodology,

4.3 Analyze and Model the Process

The Kanban implementation was based on the STATIK (System Thinking Approach to Introducing Kanban) defined by Anderson and Carmichael (2016). The author of this thesis was responsible for analyzing the STATIK steps and creating a proposal for the Kanban System, which was presented to the Product and Tech Teams for validation and approval.

Considering the steps of the STATIK, as presented in section 2.4.5:

In Step 0, identify service, the software development process of Adminly was the service identified.

In Step 1, understand what makes the service fit for the purpose of the customer. After the first phase of work, MetaProvide identified a lack of digitalization of the tasks that do not add value to their potential client’s business and which is not related to their primary expertise, making them waste time that they could be helping more people. The service fit to is to deliver a solution for the providers of well-being services that help them improve their administrative tasks management and free up their time.

For Step 2, understand sources of dissatisfaction with the current system, it was not possible to find these sources once the process of software development would be implemented in the organization at the same time as the Kanban method, once it is the first product that the organization is launching.

For Step 3, analyze the demand, it is possible to find the analysis in Chapter 3, where it is described the context of the organization and the context of the Adminly development, as well as the technology that will be used. The mockups of the system are presented.

In Step 4, Analyze the capability, before the beginning of the project, the Tech Team and the Product Team met to align the deliverables of the next phase. They also analyzed if the team was enough to deliver what the Product Team had planned for the phase. In this meeting, the necessity of hiring two more part-time developers or one full-time developer was identified, to make it possible to deliver the projects within the estimated time. One full-time developer was hired to be part of the team.

In Step 5, model the workflow, the software development process was defined as presented in Figure 20 below. The process starts after the discovery process in the Product Team. In their discovery process, they prototype, test, and design the mockups, so that the Tech Team can develop the feature. The Product Team is responsible for defining the requirements for the cards and for replenishing the Backlog session. After that, the Product Team defines the priorities for delivering each feature.

The Tech Team is responsible for developing and testing the feature. After done and deployed, the product team tries the feature in the testing environment. When the feature is approved, it is delivered to the client.

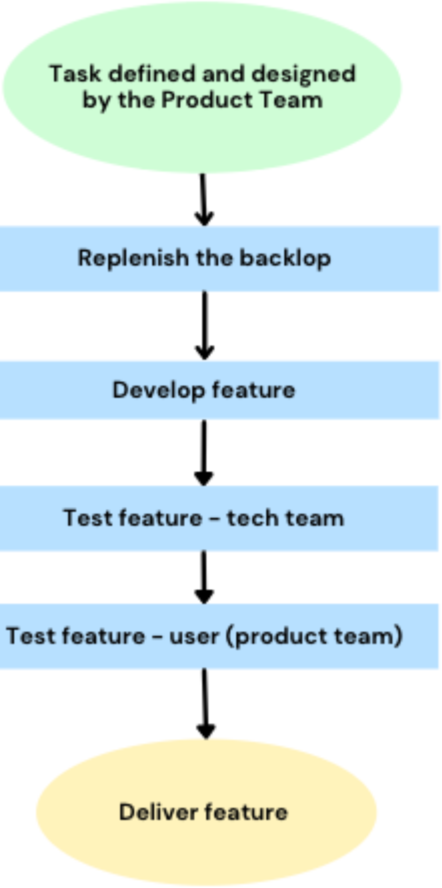

Task defined and designed by the Product Team
Replenish the backlop
Develop feature
Test feature - tech team
Test feature - user (product team)
Deliver feature

Figure 20 - Software Development Process for Adminly

In Step 6, discover the classes of services, it was decided that it won't have, at this moment, a differentiation in the classes of services. All the cards/tasks would be considered a standard work routine for developing the software. And that happens because of the way the project has been managed. It had six months of discovery before the software development phase began. And it will take six months, the length of this project, to deliver to the final customer. After that phase, this rule will probably be revisited and changed.

4.4 Proposal for the Kanban Board

Figure 21 on the next page illustrates the proposed Kanban Board to be implemented in the studied context. In continuity with the STATIK for planning the implementation of the Kanban Method, as suggested by Anderson and Carmichael (2016), in Step 7, they indicate the design of the Kanban Board. In the Kanban Board, the cards flow from the left to the right, so the card enters this board in the "Backlog" column, replenished by the Product Leader. The cards at the top of the “Backlog” column are the priority for the next two weeks of work. When the card is ready to be developed, which means it is a priority in alignment between Tech and Product Teams and has the design tested and approved, and the card has enough information to be worked on by the Tech Team, it moves to the “To Do”. The “To Do” column is the “Point of Commitment” of the Kanban Board. After moving to this point, the task must be delivered.

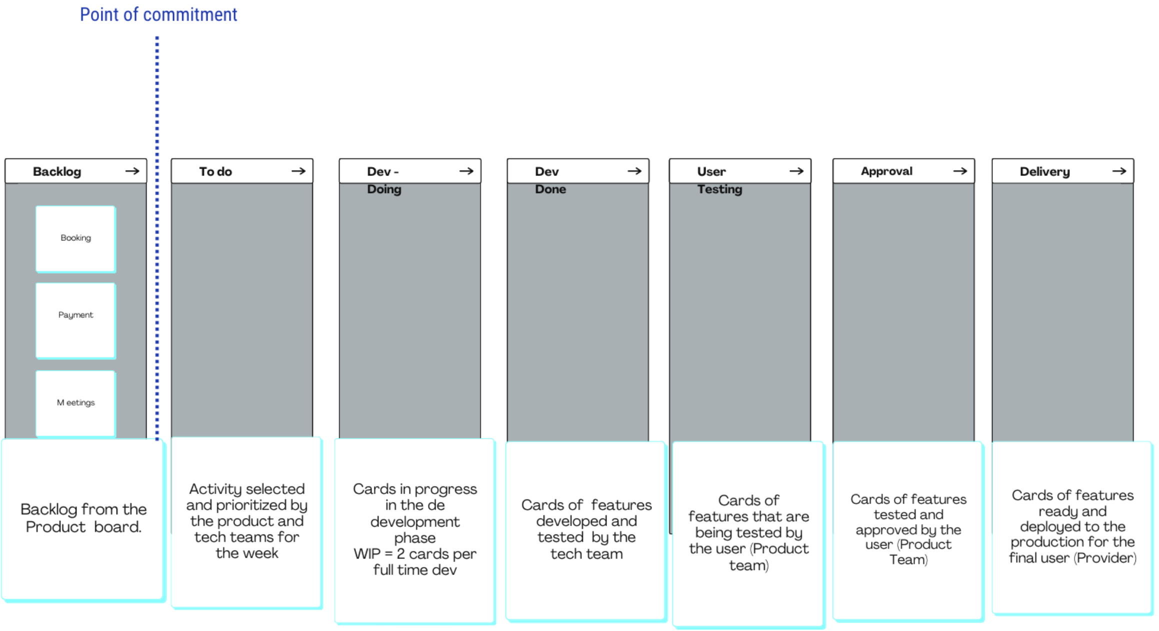

Point of commitment
Backlog
Booking
Payment
M eetings
Backlog from the Product board.
To do
Activity selected and prioritized by the product and tech teams for the week
Dev - Doing
Cards in progress in the de development phase WIP = 2 cards per full time dev
Dev Done
Cards of features developed and tested by the tech team
User Testing
Cards of features that are being tested by the user (Product team)
Approval
Cards of features tested and approved by the user (Product Team)
Delivery
Cards of features ready and deployed to the production for the final user (Provider)

Figure 21 - Kanban Board for the Adminly Software Development Process

When a developer is free, which means he has finished a task, he takes the following card available to start and moves it to the “Dev – Doing” column. This column has a Work in Progress (WIP) limit of two cards per full-time developer. If the column has achieved its limit and a developer is free, he needs to help another developer to finish his task before taking a new card. After the task is completed, it goes to the “Dev – Done” column, where another Developer tests it, and a tag of “Ready to Test” is added when the task is tested and deployed in the staging environment. After that, the Product Leader moves the card to the “User Testing” column, where the Product Leader tests the task.

If the card/feature is approved, the card moves to the “Approval” column. After that, the task is ready to be deployed in the production environment to be tested by the final customer. To facilitate the process, the finished features are only going to be deployed into production when all the features planned for the milestone are ready.

This proposal for the Kanban Board presents three out of the five Kanban properties defended by Anderson (2010). It helps visualize the workflow, limit the work-in-progress, and make the process policies explicit. In the next topic, the other two properties will be addressed: measuring and managing the flow and using models to recognize improvement opportunities.

4.5 The Kanban System

The Kanban System will contain the roles, events, metrics, and the Kanban Board. The Kanban Board was already presented in the previous topic. This topic will be divided into Kanban Roles, Kanban Metrics, and Kanban Events. These following points will address how the Kanban System was designed to be implemented in the MetaProvide context for the Adminly development.

Everything presented in this topic, and the last one, is based on the good practices presented in the literature review and was tailor-made to be applied in this specific context. These ideas were presented and aligned with the Product and Tech Teams. Adjustments in this proposed System will be constantly implemented once the concept of the System is to be improved over time, especially to respond to changes.

1. The Kanban Roles

Anderson and Carmichael (2016) define that two roles are needed for implementing and operating Kanban. They also defend that the tasks and responsibilities related to the roles are more important than the job title. The duties and people assigned to the position are the following:

- Service Request Manager: The Product Leader assumed the responsibility of being the Service Request Manager, which is:
    - Understand the customer needs;
    - Transform customer needs into requirements to support software development;
    - Create the backlog;
    - Order the work items from the backlog and facilitate prioritization of what comes next;
    - Test and approve the features according to the customers' needs.
- Service Delivery Manager: The Agile Project Manager is responsible for managing the delivery and is also the author of this thesis. The responsibilities are:
    - Guarantee that the cards flow through the Kanban Board;
    - Facilitating meetings and guaranteeing the Kanban Method is understood and followed by the team;

- Facilitate change and continuous improvement activities;
- Collect data about the work items on the Kanban Board and discuss it with the team;
- Unblock items;
- Measure metrics.

Other than the Agile Project Manager and the Product Leader, the Developers are responsible for the platform's development and support the selection and prioritization of the work to be done. In the Product Team, a group of collaborators supports the research and discovery process, conducting interviews with the customers and helping the Product Leader understand and satisfy the customer's needs. Another crucial role in this process is the UI/UX Designer, responsible for creating the mockups of the platform in a way that presents a good experience for the client.

2. The Kanban Events

Anderson and Carmichael (2016) point out that the Replenishment Meeting and the Kanban Meeting are the baselines for implementing the Kanban Method. Other than that, also based on Anderson and Carmichael (2016), the events implemented to support the Kanban implementation in the MetaProvide context are the following:

- Kanban Meeting: the objective of this meeting is to visualize the Kanban Board, unblock tasks, and guarantee that the work is flowing through the process. The meeting will happen twice a week and last 15 minutes. The meeting is facilitated by the Agile Project Manager and has the participation of the Tech Team;
- Delivery Planning Meeting + Replenishment: Before this meeting, the Product Leader is responsible for creating the cards in the "Backlog" column based on the roadmap and priorities defined for the project. During the meeting, the cards for the week will be moved from "Backlog" to "To Do", so the development can start. This meeting will happen biweekly, on a Monday, and is planned to be one-hour long. The Product Leader, the UI/UX designer, the Agile Project Manager, and the Tech Team attend this meeting;
- Service Delivery Review + Retrospective: In this meeting, the Product Leader would Review, along with the Tech Team, if the work planned for the two weeks of work is working correctly and explain the changes or possible malfunctions in the software to decide what is approved and what is not. After this moment, a Retrospective Meeting is held, regarding the last two weeks of work. The Retrospective is done on a Board, with

the columns "Well", "Not so Well", and "New Ideas". The Agile Project Manager is responsible for facilitating both meetings and ensuring the Ideas from the Retrospective will be implemented and tested, what works will become part and a change in the Kanban System or the Kanban Board, and what does not work will be discarded. This meeting will happen biweekly, on a Friday, and is planned for 90 minutes.

In addition to meetings, the Tech and Product Team have a channel on Talk, the NextCloud message app, to communicate asynchronously, especially on the days they do not have meetings. The creation of this channel of communication aims to facilitate communication among the people involved in Adminly Development. The team can ask questions regarding the cards, the meetings, the design, and any other topic related to the development of the platform. With this channel, the team does not need to wait for the next meeting to solve an issue or to unblock one of the tasks they might be having trouble finishing.

3. The Kanban Metrics

There are three main metrics to follow up on the performance of this project, Lead Time, Throughput, and WIP, or work in progress. All of them were defined according to Anderson and Carmichael (2016).

The WIP was limited based on the time the development team could dedicate to this project. As the team was composed of two part-time developers, with a commitment of 20 hours per week each, and one full-time developer, with a commitment of 40 hours per week, it was defined as a WIP for the "Dev – Doing" column of 4 tasks. For every 20 hours of weekly commitment, one task could be added to the WIP. The developers must respect the limit of WIP, and the follow-up of this metric would happen during the Kanban Meeting, twice a week, by the Agile Project Manager.

The Throughput, or Delivery Rate, is the average of tasks delivered in a determined period (Anderson and Carmichael, 2016). To calculate this metric, at the end of the project, the number of cards delivered will be divided by the number of weeks, so it will be possible to measure the weekly delivery rate.

According to Anderson and Carmichael (2016), the Lead Time is the time from the committed point to the delivery point. The commitment point in the board happens when the card is moved from the "Backlog" to the "To Do" column. The delivery point, in this situation, is going to be considered the "Approved" column because in this first phase of the project, the card is only going to be moved to the "Delivery" at the end of a milestone, as the final customer will not be using the platform yet.

The Lead Time will be calculated at the end of the project when the team delivers the functional MVP. The main idea here is to understand how long the team takes to work on one demand or one card, so it will be possible to estimate it based on the history of deliverables of this phase. This data will be gathered from NextCloud, as every card registers its movements through the columns of the Kanban Board.

Another interesting metric that will be applied to this project is the Work Item Age, the age of a card on the board. This metric will guide the team to prioritize a task that is taking longer than usual, so during the Kanban Meetings, the team can evaluate actions to work on this task and manage the board better.

4. Implementing the Kanban Method

The implementation started on February 21st, along with the beginning of the Adminly development. In this first meeting, the proposed structure for the Kanban Method was presented to the Development Team, including the Product Leader and the Tech Team. For that, a presentation material was created that is available to check in Appendix 1 – Kanban Presentation.

The team agreed to test the proposed framework created for this process. During the presentation, it was clear to everyone that it was a proposal. While implementing it, the team would participate in suggesting changes and adaptations to fit the team better to help improve the visualization of the work and the management of the workflow. Changes could be made to, for instance, the responsibilities of the people involved in adding or removing elements in the Kanban Board structure, as well as meeting frequency and duration.

The Retrospective Meeting is the main cadence of the feedback loop of the proposed Kanban for the development of Adminly. It is a moment for the team to discuss what is going well and what is not going well and present new ideas to be implemented during the next work cycle. In the first two months, the Retrospective happened every two weeks to adjust the Kanban Method, and after defining a framework that works well for the team, it will start happening every month. The team is also encouraged to discuss improvements and give feedback during the Planning, Review, and Kanban Meetings.

As the Product Team had worked for the past six months on the discovery process for Adminly, there was already a backlog created for the subsequent development cycles. The main task is prioritizing this backlog, considering what delivers more value to the customer.

5. Improvement Cycles and Results

This chapter presents the improvements implemented in the five feedback cycles until achieving a Kanban System that fits the team's dynamics and context. It also shows the project's results, if it was delivered on time and within budget, and the metrics of the Kanban.

5.1 Improvements

As explained in the last chapter, the Kanban Method promotes the Kanban cadences as cycles of constant improvement in the system (Anderson and Carmichael, 2016). In this case, the most important event for feedback and improvement is the Retrospective Meeting. Figure 22 presents how the Kanban was implemented based on the Action Research structure. The project started defining the context and purpose and selecting an agile methodology to be implemented. After that, there was one cycle of Kanban Implementation, presented in chapter 4. Then, the research was composed by five cycles of improvements, that will be presented in this chapter. The main moment of feedback, as already cited, is the Retrospective Meetings, that happened at the end of each cycle.

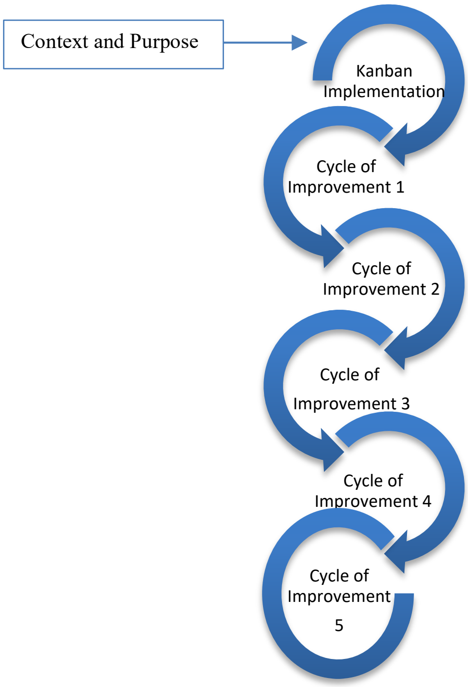


Figure 22 - Kanban cycles of improvement

Each Kanban cycle starts with Kanban Planning meeting, a meeting involving Product and Tech teams to define the priorities for the next two weeks of work. During this time, two Kanban Meetings per week are done to unblock impediments to the tasks and visualize the Kanban

Board, to guarantee it is following its policies and respecting its limit of WIP. After two weeks, there is a meeting to review the work delivered along with a retrospective.

From the beginning, the idea of the Retrospective was to be a simple moment of reflection on understanding what is going well and thinking of new ways of working as a team to deliver better or to improve the Board and the communication within the team and between teams.

A board supports the retrospective on the NextCloud platform, which is used to support the Kanban Board. The columns, as shown in Figure 23 below, are the following: "Well", "Not so Well", and "New Ideas". In the column "Well," the team points out what is going well in the last cycle of work. In the column "Not so Well", the team mentions what needs to be improved. And finally, in the column "New Ideas", the team thinks about actions to improve the work, the ideas can or cannot be related to the point of improvement cited in the previous column.

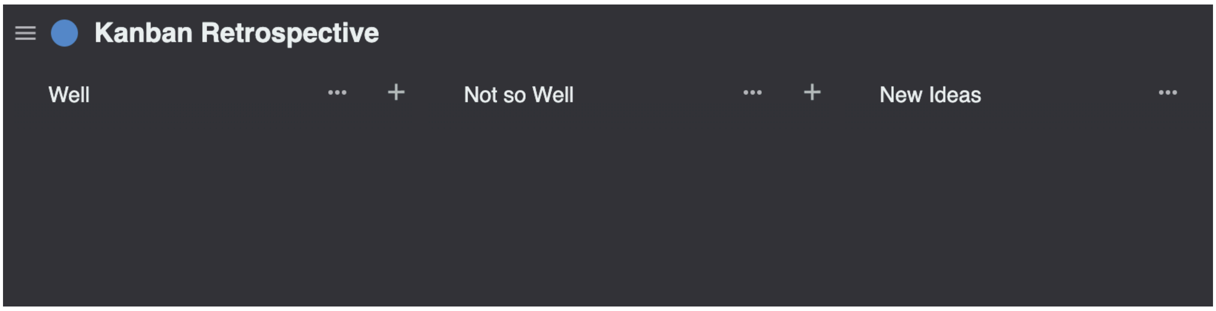


Figure 23 - Kanban retrospective board

Even being the Retrospective Meeting the official moment to discuss improvements in the Kanban System, the team is encouraged to give feedback and suggestions that can make the tasks flow in a better way, to improve the process, communication, or the way the meetings are conducted.

1. 1st Cycle of improvement

After the presentation, it took two weeks to finally start the Kanban implementation and test the proposed structure for the Kanban System, the one that was detailed in session 4.4. The first Kanban Meeting happened on March 8th, 2022. The Retrospective Meeting for this cycle occurred on the 25th of the same month.

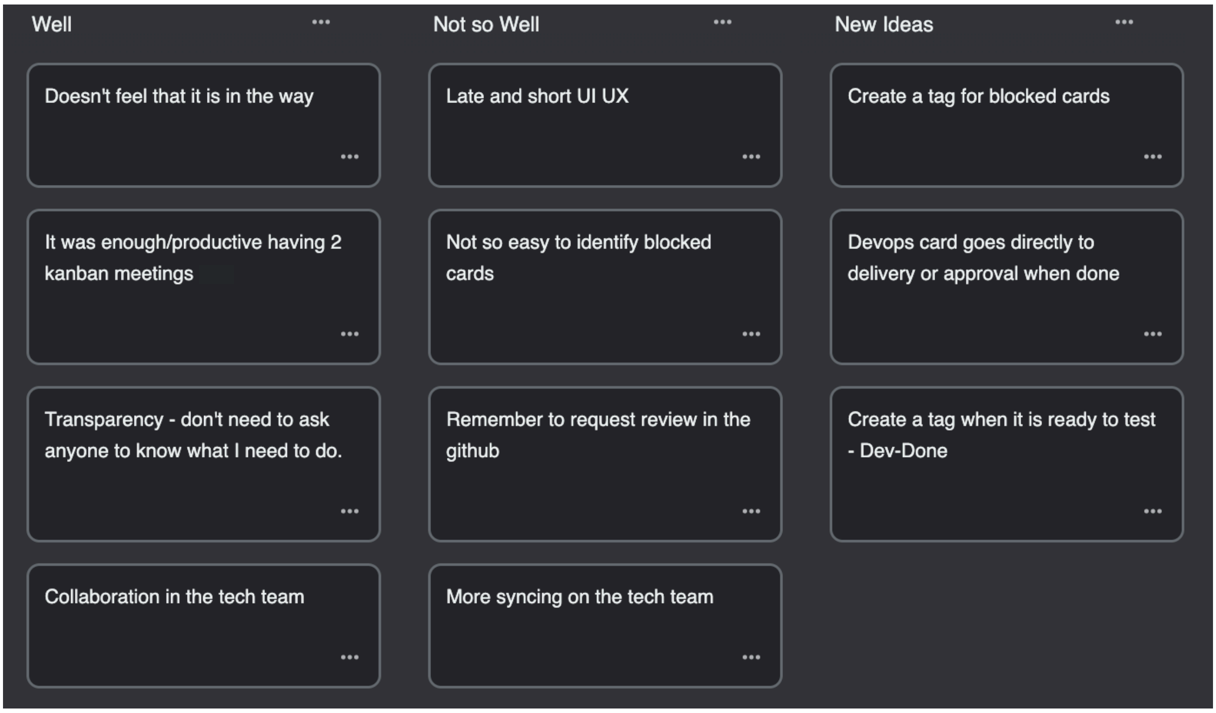
In the first week of Kanban usage, before the Retrospective Meeting, a few suggestions were made by the development team to improve the process of development and delivery. One of them was to create one more column called "In Review". So, after the task is finished in the

“Dev – Doing” column, it goes to “In Review”, and when a task is moved to the “Dev – Done”, it means it was done and approved by the tech team. The developers know when the task is finalized in the “Dev – Doing” when the developer responsible for the task requests a review in the GitHub Platform. After that, the developer responsible for testing moves the tasks to “In Review”, and if approved, they move it to “Dev – Done”. This change happened during a Kanban Meeting.

Regarding the improvements that came from the Retrospective meeting, Figure 24 below shows the cards added by the team during the first Retrospective Meeting. The Tech Team, the Agile Project Manager, responsible for facilitating the meeting, and the Product Leader were present at the meeting.

The Dynamic of the meeting is the following. The team has 10 minutes to add cards in the “Well” and “Not so Well” columns. After this moment, the group presents their cards and discusses why they presented those points and if they agree with them or not. Then, the team thinks about new ideas for improving the topics they showed in the “Not so Well” column. Other ideas, not related only to the improvement of the previously discussed points, can also be added to the discussion.

The new ideas will become actions, and, if the team agrees, the actions will be implemented in the next cycle of the Kanban. What works and improves the process will become part of the Kanban System. What does not work will be discarded. The decision regarding that is discussed in the next Retrospective, when the team evaluates how the work was in the cycle, proposing new ideas and how the Kanban System is continuously improved.



Well	Not so Well	New Ideas
Doesn't feel that it is in the way	Late and short UI UX	Create a tag for blocked cards
It was enough/productive having 2 kanban meetings	Not so easy to identify blocked cards	Devops card goes directly to delivery or approval when done
Transparency - don't need to ask anyone to know what I need to do.	Remember to request review in the github	Create a tag when it is ready to test - Dev-Done
Collaboration in the tech team	More syncing on the tech team	

Figure 24 - 1st Cycle of Improvement - Retrospective Board

In the first cycle of improvement, marked by the retrospective meeting, the points that were going well, as shown in Figure 24, are:

- It (the Kanban Method) does not feel that it is in the way;
- It was enough/productive having two Kanban Meetings per week;
- Transparency – we do not need to ask anyone to know what we need to do;
- Collaboration in the tech team.

The not-so-well points were:

- Late and short UI/UX (User Interface and User Experience design);
- It is not so easy to identify blocked cards;
- Remember to request a review on the GitHub platform;
- More syncing on the tech team.

The new ideas or actions to implement were:

- Create a tag for blocked cards;
- DevOps cards go directly to delivery or are approved when done;
- Create a tag when the task is ready to test – Dev–Done.

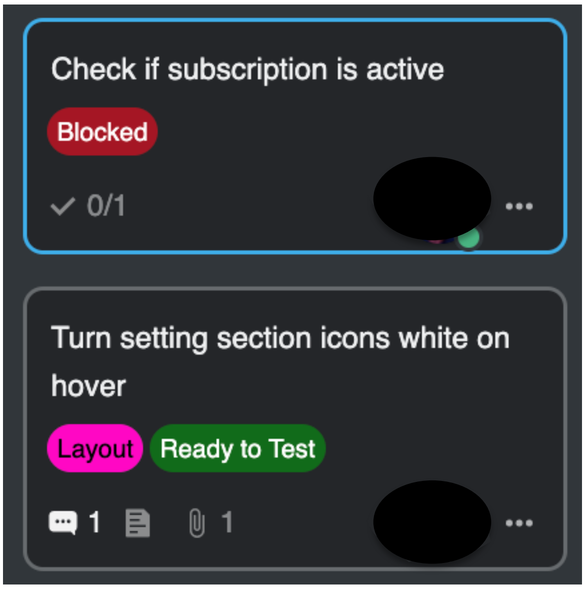
This first retrospective meeting was crucial to show the first impressions of using the Kanban Method and how the team was feeling regarding managing the tasks. The team pointed out that they could see Kanban’s transparency and that the tech team’s collaboration has improved. Another point is that now they do not need to ask another person what comes next, because it is possible to visualize it in the Kanban Board.

The first impressions match what Ahmad et al. (2018) discovered during their literature review about Kanban, that the method benefits the improvement in visibility, better management of the workflow, and a good impact on team cohesion and team building.

One interesting comment is the first card in the “Well” column when a developer says that the Kanban Method does not feel to be in the way. One of the team’s concerns before selecting the agile methodology to be implemented in this context was a large number of meetings that could take up the time invested in software development. On another topic, they mentioned that two Kanban Meetings per week was enough because the team was discussing having two or three meetings, so it was defined to test two, and if it was not enough, it could be implemented one more.

In the Scrum Methodology, for example, the meeting to check the work status, as defended by Schwaber and Sutherland (2020), happens daily. It is called Daily Scrum. The definition for two or three meetings for this team took into consideration that two out of three programmers worked only part-time (around 20 hours per week), and if it was needed to have more meetings, it could be done asynchronously.

In the not-so-well, the team pointed out the need to improve the visualization of the system, as it was challenging to identify blocked cards. To solve that, in the new ideas, the team suggested implementing new tags in the cards to signal when a card is blocked or when a card is ready to be tested by the Product Development Team. An example can be checked in Figure 25 below.



Check if subscription is active

Blocked

✓ 0/1 [redacted] •••

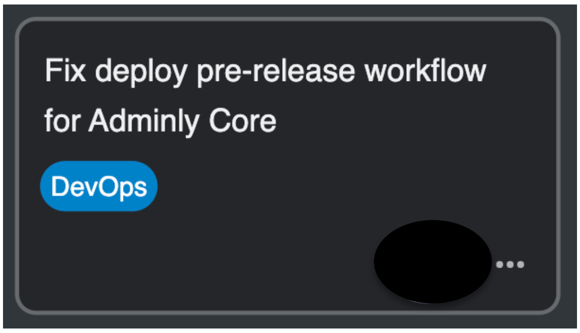
Turn setting section icons white on hover

Layout Ready to Test

💬 1 1 [redacted] •••

Figure 25 - Tags for blocked and ready to test cards

Another relevant point is that it was defined that tasks related to DevOps would be moved, when done, directly to the "Delivery" column, once it does not need approval by the Product Team. The DevOps tasks are related to the system's setup or improvement of the release and other technical processes. In this case, it could be classified as a new class of service, but the impact on the project is low once the number of tasks related to DevOps is small. It was defined that the only way of identifying them would be by using a tag with the name "DevOps", as shown in Figure 26 below.



Fix deploy pre-release workflow for Adminly Core

DevOps

[redacted] •••

Figure 26 - DevOps card

One more point for improvement was the "More syncing on the tech team". The team identified that they communicated better during the meetings but did not try to reach out to the others, when they were working on the development tasks. During the discussion, they agreed to try to communicate better in other moments, not only during the Kanban Events.

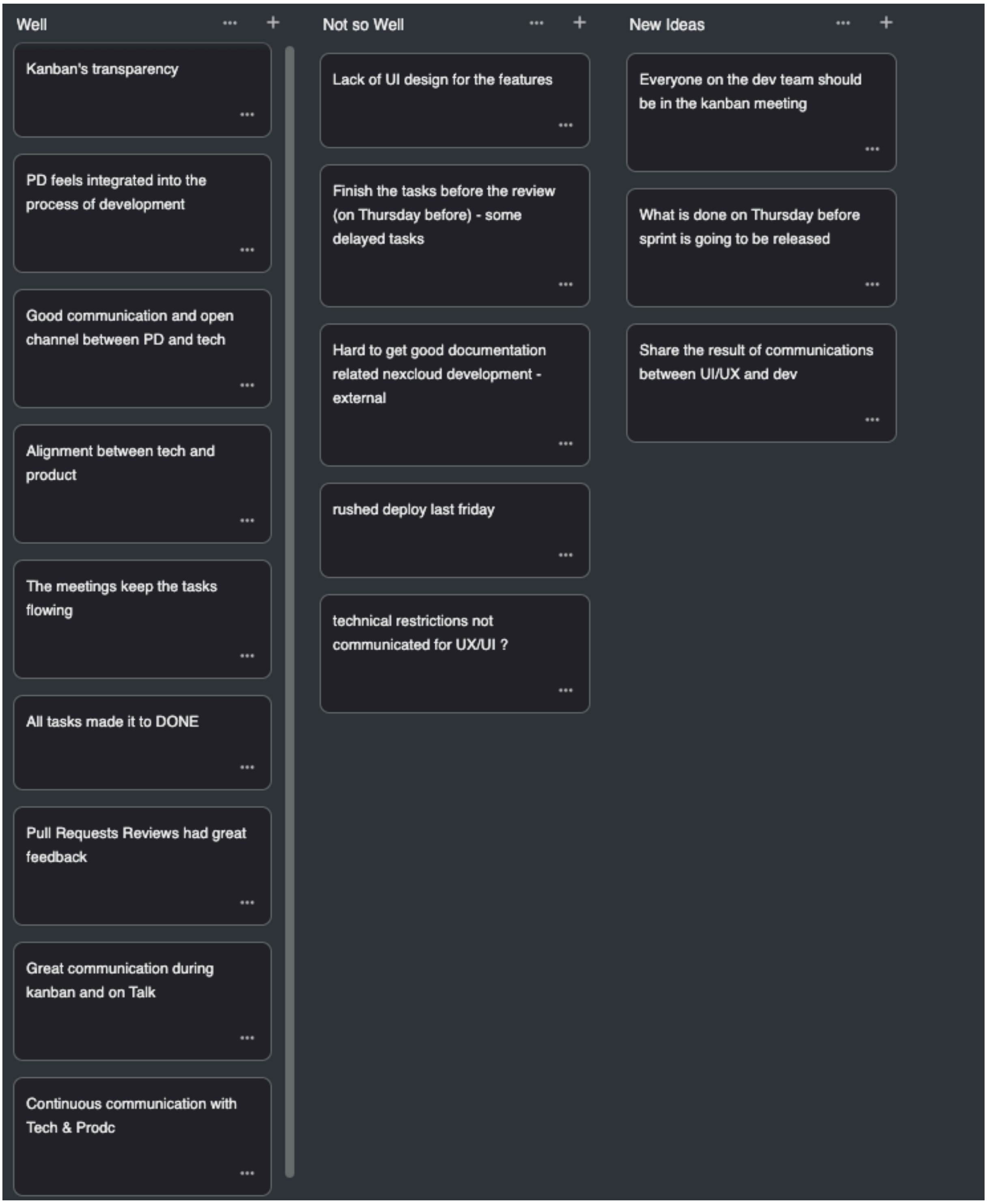
2. 2nd Cycle of Improvement

Right after the first Retrospective Meeting, the actions discussed during the meeting were implemented, and all of them became part of the process. The tags were implemented in the Kanban Board, and the Tech Team members started to have more one-on-one meetings and pair programming to sync the communication. The cards regarding DevOps tasks began to have a new process and were moved directly to the “Delivery” column when done.

The second Retrospective happened on April 8th, 2022, around two weeks after the first one. The team’s contribution in this meeting was more significant, and it is possible to notice it by the number of cards added to the Retrospective Board.

In the “Well” column, the team mentioned, as it is presented in Figure 27:

- Kanban’s transparency;
- PD (Product Development Team) feels integrated into the process of (software) development;
- Good communication and an open channel between PD and Tech Teams;
- Alignment between Tech and Product Team;
- The meetings keep the tasks flowing.



Well

Kanban's transparency

PD feels integrated into the process of development

Good communication and open channel between PD and tech

Alignment between tech and product

The meetings keep the tasks flowing

All tasks made it to DONE

Pull Requests Reviews had great feedback

Great communication during kanban and on Talk

Continuous communication with Tech & Prodc

Not so Well

Lack of UI design for the features

Finish the tasks before the review (on Thursday before) - some delayed tasks

Hard to get good documentation related nexcloud development - external

rushed deploy last friday

technical restrictions not communicated for UX/UI ?

New Ideas

Everyone on the dev team should be in the kanban meeting

What is done on Thursday before sprint is going to be released

Share the result of communications between UI/UX and dev

Figure 27 - Second Cycle of Improvement - Retrospective Board

- All tasks made it to done (The “Dev–Done” column);
- Pull request reviews had great feedback;

- Good communication during Kanban Meetings and on Talk (The messaging app used by the organization, explained in Chapter 3.2);
- Continuous communication between Tech and Product Teams.

A few topics were repeated a few times in the Retrospective Board, especially regarding the integration between the Product Development Team and the Tech team. The reason for that is that different team members added those topics in the 10 minutes they had to add their inputs on the board.

Regarding the methodology, the team mentioned Kanban's transparency again. They also noted that the meetings were helping the work to flow. During the Kanban Meetings, it is the primary time to visualize the Kanban Board and talk about impediments the team can find while developing a feature, and it helps the team to solve an issue or to understand the task better, making the tasks flow better.

In the "Not so Well", the team mentioned, as is shown in Figure 27 above:

- Lack of UI (User Interface) design for the features;
- Finish the tasks before the Review Meeting (on Thursday before) – some delayed tasks;
- Hard to get good documentation related to NextCloud development – external issues;
- Rushed to deploy last Friday;
- Technical restrictions not communicated for UI/UX.

At this moment of development, a UI/UX developer was being integrated into the team, and there was a lack of communication and misalignment between the software development and the design. The team also reported difficulty finding documentation for the NextCloud. Regarding the development process, the group mentioned the issue of going to the Review Meeting without all the tasks finished on the development side being tested by the Product Team because the Tech Team had not deployed them. The team decided to improve the process by determining that all tasks done by Thursday would be deployed, so the Product Team could test them before the Review Meeting that happens on Fridays.

In the new ideas, the team also suggested:

- Share the communication between the developers and the UI/UX designers with the rest of the team;
- Everyone on the Tech Team should attend the Kanban Meetings.

This last topic was suggested after a few absences by the team members, which made it difficult to have a complete overview of the Kanban Board.

In the second improvement cycle, the ideas and enhancements tested in the previous cycle proved effective, becoming part of the process and Kanban System. Additionally, the Retrospective has shown that the team has accepted the Kanban Method and that improvement in the just-implemented software development process has already been made. A few more ideas were suggested to be implemented in the next improvement cycle.

3. 3rd Cycle of improvement

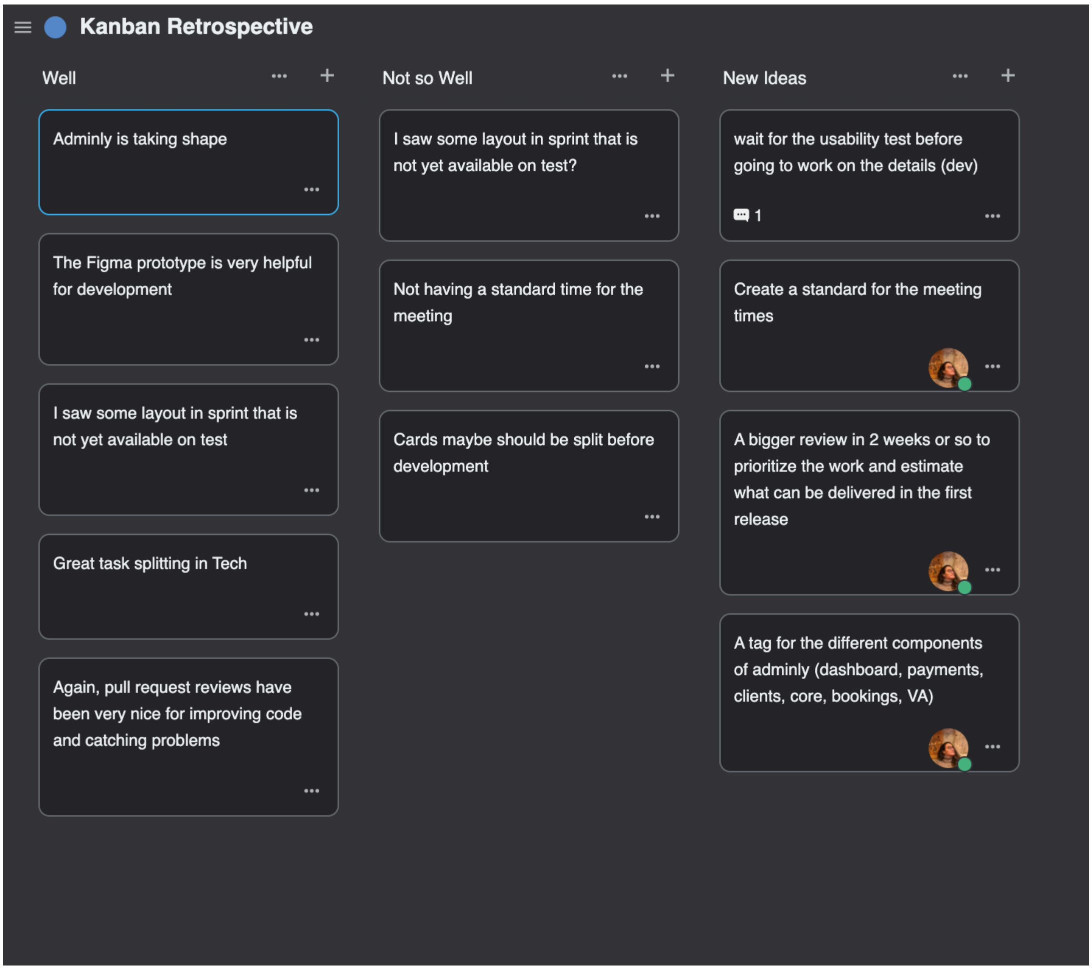
The third improvement cycle started after the Retrospective Meeting, implementing the proposed actions. From that moment on, it was decided that if one of the developers was not available to participate in the Kanban Meeting, the meeting should take place at another time. Still, at the same time, it was defined as a priority for the team to participate in the meeting, as it was noticed that the participation of the Teach Team was fundamental to making the Kanban Method function well.

The third Retrospective Meeting happened on April 25th, 2022. The team identified the following topics in the "Well" column, as seen in Figure 28.

- Adminly is taking shape;
- The Figma prototype is very helpful for development;
- Great tasks splitting in Tech;
- Again, pull request reviews have been very nice for improving code and catching problems.

The interesting point to see as a result of this Retrospective Meeting is the topic regarding the Figma prototype. In the last two meetings, the team complained that the lack of UI/UX design or the lack of alignment between design and development was compromising the project. This week, after the prototype was adjusted, the team could see the benefits of the process. This kind of improvement in the process could happen or could have been addressed without the Kanban Method, but the cadences of the Kanban promote an agile response to the issues and constant process improvement as promoted by Anderson and Carmichael (2016).

Another important topic was that the team mentioned the need to send a pull request review on GitHub in the first retrospective. In this retrospective, they pointed out that the process is working and has been helping improve the code and catching problems.



Kanban Retrospective

Well

- Adminly is taking shape
- The Figma prototype is very helpful for development
- I saw some layout in sprint that is not yet available on test
- Great task splitting in Tech
- Again, pull request reviews have been very nice for improving code and catching problems

Not so Well

- I saw some layout in sprint that is not yet available on test?
- Not having a standard time for the meeting
- Cards maybe should be split before development

New Ideas

- wait for the usability test before going to work on the details (dev)
  💬 1
- Create a standard for the meeting times
- A bigger review in 2 weeks or so to prioritize the work and estimate what can be delivered in the first release
- A tag for the different components of adminly (dashboard, payments, clients, core, bookings, VA)

Figure 28 - Third Cycle of Improvement – Retrospective Board

In the “Not so Well” column, the team said, as presented in Figure 28:

- I saw some layout in the sprint that is not available on the test;
- Not having a standard time for the meetings;
- Maybe the cards should be split before the development.

The three presented points are essential to be improved with the Kanban. The first one shows that the team presented a layout to the organization before aligning with the Product Team. The second is regarding the Kanban Meeting that every week was changing the scheduled time, because of team availability and other meetings, so it was defined that from that moment on, the meeting would start having an established time to happen twice a week, on Wednesdays and Fridays. The third topic is to start assigning tasks before developing, so in the planning meeting, a few cards began to be already assigned to the developer responsible.

In the new ideas, the following point was suggested:

- Wait for the usability test before working on details (Software Development);

- Create a standard time for the meetings (Kanban Meetings);
- A bigger review in two weeks to prioritize the work and estimate what can be delivered in the first release;
- A tag for the different components of Adminly (dashboard, payments, clients, core, bookings, VA – or virtual assistance, etc.).

At this moment, it was defined the creation of tags to identify what group of features the task comes from, if it is: layout, clients, dashboard, bookings, payments, and so on, as shown in the prototypes in chapter 3, to improve the visibility of the Kanban Board.

Two relevant decisions were made in this meeting. First was waiting for the platform to be tested by a client before developing details. The second one was to schedule an extensive review to prioritize what comes next and estimate what would be delivered in the first big release. For the objective established at the beginning of the project of having a functional MVP to start testing with the clients, it is crucial to be careful not to develop features that will not deliver value to the final customer.

During this meeting, it was decided that from this moment on, the Retrospective Meeting would happen monthly, not biweekly anymore. The team decided that because the Kanban System is already integrated into the team routine, the process works well, and no significant improvements were identified. The communication is good between the Product and Tech teams.

4. 4th Cycle of Improvement

The fourth improvement cycle is the first one, with a duration of one month. The process started, as the previous one, with the implementation of the suggested ideas. All the Kanban meetings were scheduled to begin at a specific time on Wednesdays and Fridays from that day until August 21st. It helped to improve consistency and participation in the meetings.

The tags to signalize the correspondent group of features in the system were created. The tags have been helping the Product Team visualize which feature the development is focusing on and the priorities. Figure 28 shows all the tags used to support the visualization of the work.

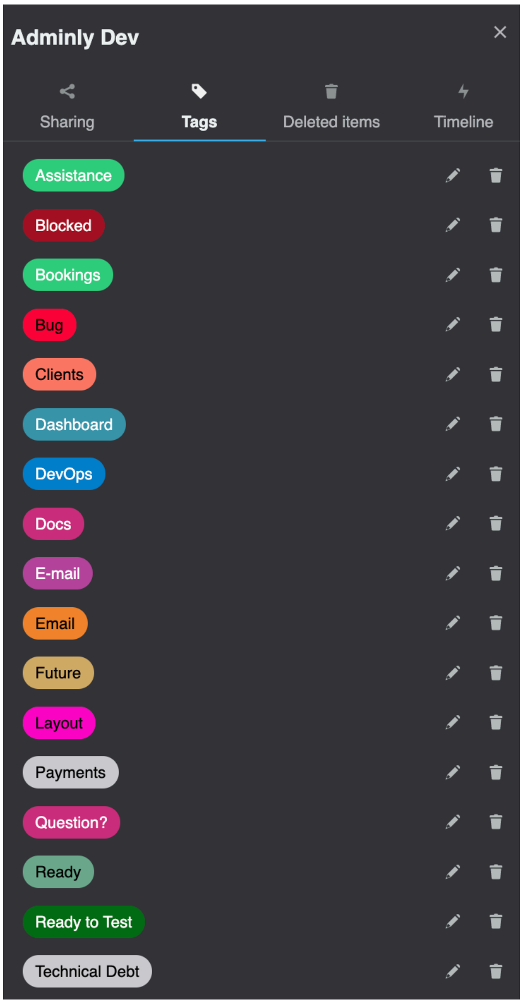


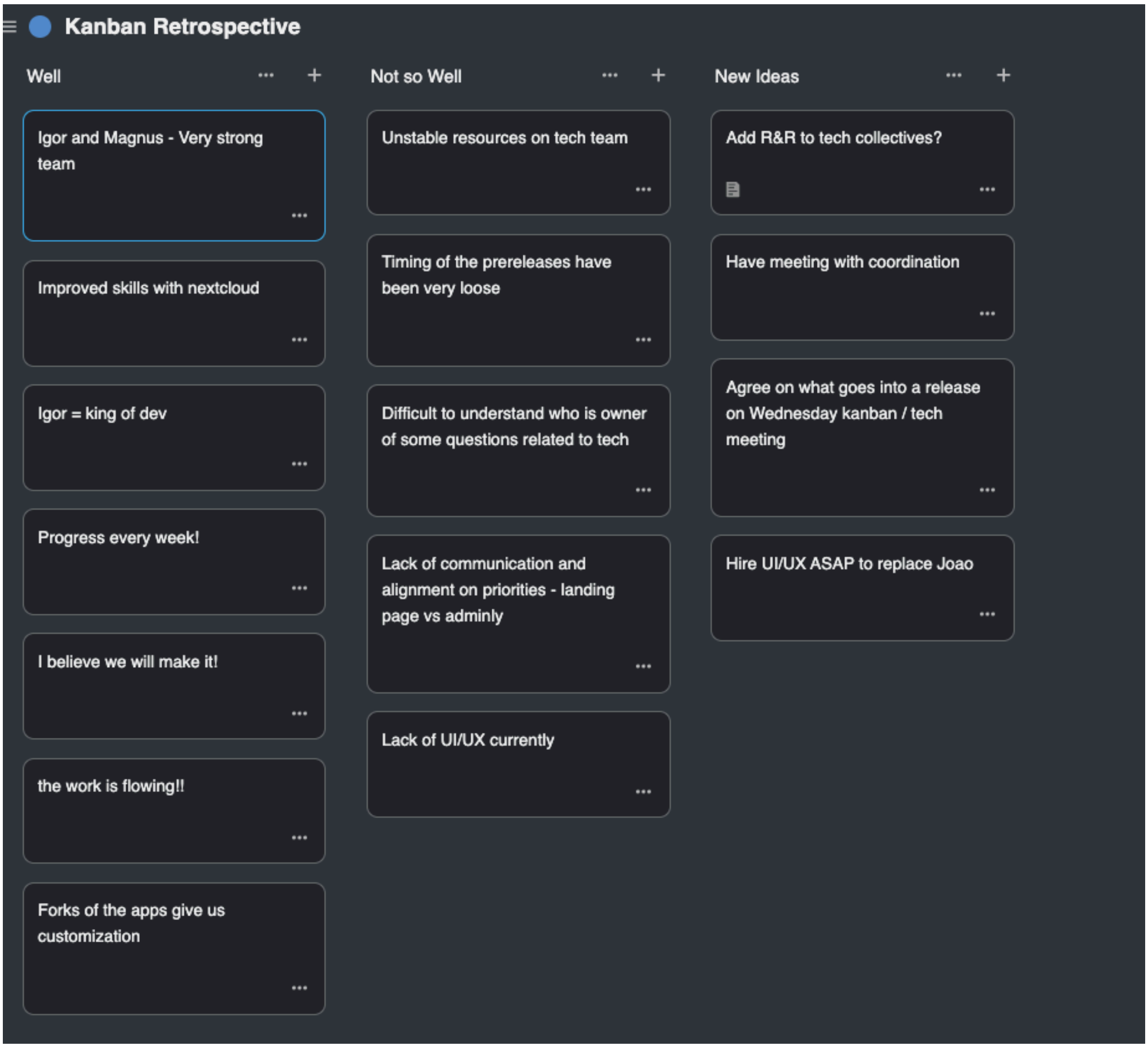
Figure 29 - Tags of the Kanban Board

The fourth retrospective happened on June 3rd, 2022. In the "Well" points of this retrospective, the team pointed out, as shown in Figure 30:

- Igor and Magnus (two members of the Tech Team responsible for the software development) are – Very strong team;
- Improved skills with NextCloud;
- Igor = king of dev (software development);
- Progress every week;
- I believe we will make it;
- The work is flowing;
- Forks of the apps give us customization.

The Tech Team significantly improved and delivered good work in this cycle, especially the two developers mentioned. The team used the Retrospective Meeting as an opportunity to acknowledge them.

Again, one of the topics is related to the fact that it is possible to see the work flowing, and the team also mentioned that they believe the product will be delivered on time.



Kanban Retrospective

Well

Igor and Magnus - Very strong team

Improved skills with nextcloud

Igor = king of dev

Progress every week!

I believe we will make it!

the work is flowing!!

Forks of the apps give us customization

Not so Well

Unstable resources on tech team

Timing of the prereleases have been very loose

Difficult to understand who is owner of some questions related to tech

Lack of communication and alignment on priorities - landing page vs adminly

Lack of UI/UX currently

New Ideas

Add R&R to tech collectives?

Have meeting with coordination

Agree on what goes into a release on Wednesday kanban / tech meeting

Hire UI/UX ASAP to replace Joao

Figure 30 - Fourth Cycle of Improvement – Retrospective Board

In the “Not so Well” the team mentioned:

- Unstable resources on the tech team;
- The timing of prereleases has been very loose;
- It is difficult to understand who the owner of some tech-related tasks is;
- Lack of communication and alignment on priorities – landing page or Adminly;
- Lack of UI/UX currently.

To add context to the points mentioned related to the Tech Team, the position for a PHP programmer mentioned in session 3.8 was never fulfilled. This overloaded the team between

Adminly development and the other demands from the organization. The prioritization of the work was not well communicated between teams, which also disturbed the resource availability. Other than that, the UI/UX designer left the organization during this cycle.

On the “New Ideas”, the team suggested that the Tech Team could create a Roles and Responsibilities document to address the lack of clarity regarding the ownership of the tasks and responsibilities. One issue was caused by the fact that the organization, as mentioned before, is a flat organization, so the leadership of some activities was unclear.

Another point was for the Tech Team to meet the organization’s coordinator and define the priority of the work.

To solve the problem of the looseness of the prereleases, the Tech Team will define on Wednesdays what will be released on Thursdays.

The last topic is to hire an UI/UX designer because the Product Leader was overwhelmed filling the lack of this position to reduce the impact of not having this person in the organization.

One clear issue in this cycle was that not hiring a PHP position, which was opened at the beginning of the project, made the Tech Team work more disorganized, since the team is also responsible for other demands of the organization. With that, it is urgent to hire someone to fill this gap.

Other than that, the ideas implemented at the beginning of this cycle improved the work and will continue as part of the Kanban System.

5. 5th Cycle of Improvement

The fifth improvement cycle is the last one that will be presented in this thesis. After this, it will discuss the metrics, how the proposed Kanban System was modified after five Retrospective Meetings and the impact of it on delivering the functional MVP.

The ideas of the last cycle were implemented or at least tried to be implemented. The RH did not find an UI/UX designer yet, so the position was still open during this fifth cycle. The Tech Team created a RACI (responsible, accountable, consulted, and informed) Matrix to define the roles and responsibilities within the team. They also started defining the release on Wednesday, which is working better. In the last topic, they determined the priorities of the organization coordinator.

So, in the “Well” column presented in Figure 31 on the next page, it is possible to see the impact of the actions that came from the last retrospective. They were:

- Forks of the apps give us customization;
- Adaptable/Agile;

- R&R RACI (roles and responsibilities RACI Matrix);
- Communication improved regarding absence and availability;
- Good communication;
- Product and Tech (teams) are aligned in the way to approach the work;
- Good prioritization of features;
- Not so many bugs.

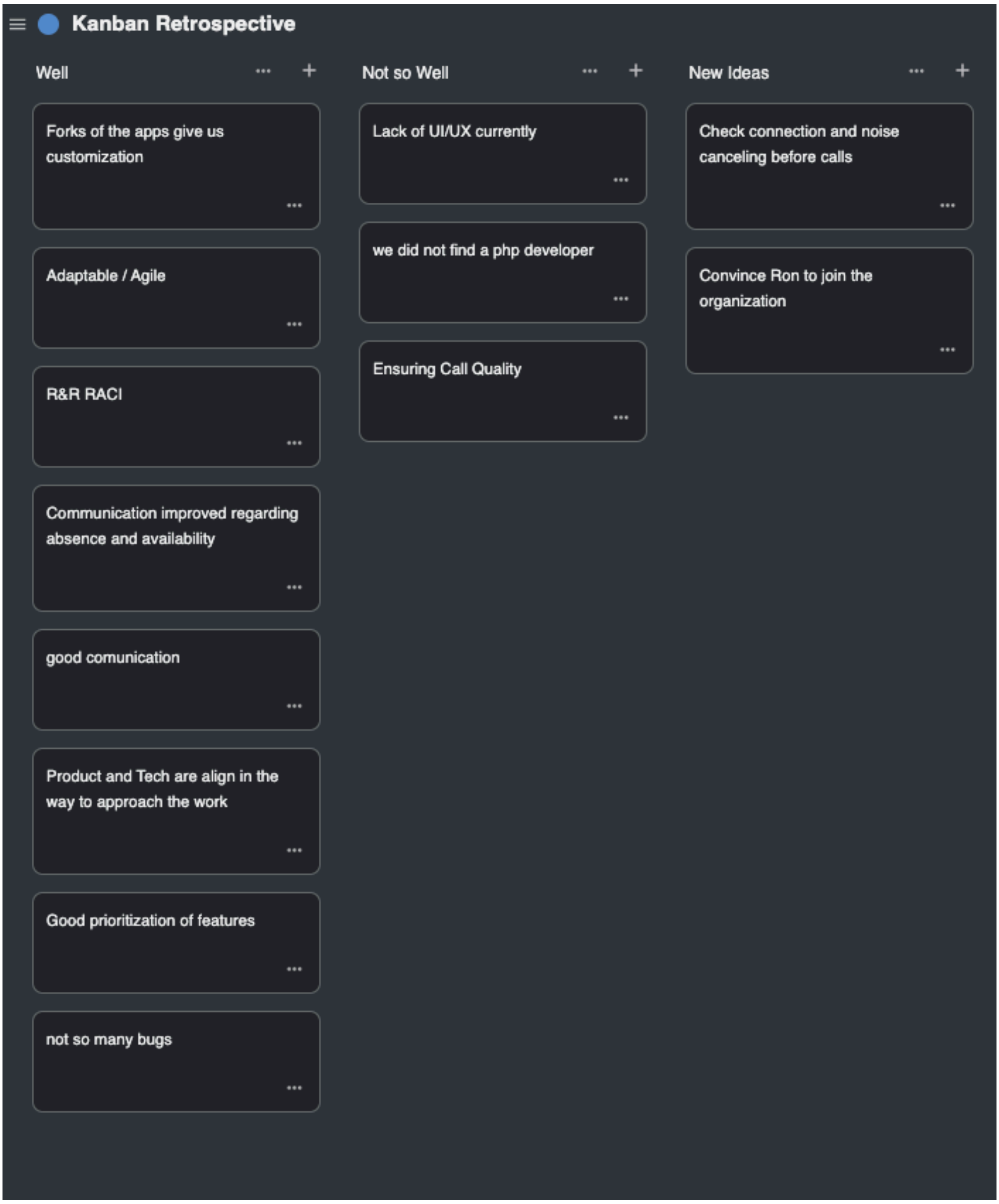
In this retrospective, it is possible to notice the considerable improvement in the communication and alignment between the two teams. The prioritization of the work has also improved. They also mentioned the tremendous impact of the method and classified the work as agile and adaptable.

In the "Not so Well" column, the absence of a UI/UX continues to be an issue, as well as the lack of a PHP programmer, which has not yet been found. Another topic is, as the team works remotely from different countries, sometimes it is difficult to ensure the quality of the internet connection or the noise in the places people are working from, especially if the person is traveling, so it was pointed out that it was disturbing the meetings.

In the "New Ideas", they mentioned two things:

- Check connection and noise cancellation before calls;
- Convince Ron to join the organization.

Ron is a PHP developer working on an external project related to the MetaProvide organization. As the other project has finished, the team thinks he could help develop Adminly, so the group decided to talk to him to see if he is interested in joining the Tech Team.



Kanban Retrospective

Well

- Forks of the apps give us customization
- Adaptable / Agile
- R&R RACI
- Communication improved regarding absence and availability
- good comunication
- Product and Tech are align in the way to approach the work
- Good prioritization of features
- not so many bugs

Not so Well

- Lack of UI/UX currently
- we did not find a php developer
- Ensuring Call Quality

New Ideas

- Check connection and noise canceling before calls
- Convince Ron to join the organization

Figure 31 - 5th Cycle of Improvement – Retrospective Board

The team did not give suggestions regarding the Kanban structure in the last two Retrospective Meetings. But in this previous cycle, the team decided that it would be better to plan weekly with the last week's review on Mondays. And from that moment on, every Monday, the team would have a meeting starting with the last week's review and then plan the work for the next week. The retrospectives will continue to happen monthly, and the Kanban Meeting is on Wednesdays and Fridays.

5.2 Results

As results of this thesis, this topic presents the final Kanban structure, as the result of the cycles of improvement, the metrics of the Kanban and the project time and budget. The main changes suggested and successfully implemented during the cycles of improvement are presented in Table 2. The changes impacted the Kanban structure that will be presented in the next topic.

Table 2 - Main changes made during the improvement cycles

First Cycle	Second Cycle	Third Cycle	Fourth Cycle	Fifth Cycle
Create tags to blocked tasks.	Releases happens always on Thursdays.	Create a standard time for the meetings (Kanban Meetings).	Define Roles and Responsibilities of the Tech Team members.	Planning Meeting and Review will happen weekly on Mondays.
DevOps cards go directly to delivery or are approved when done.	Share the communication between the developers and UI/UX designers with the rest of the team.	Retrospectives happens monthly.	Define on the Tech Wednesday meeting what will be deployed on Thursday.	
Create a tag when the task is ready to test – Dev – Done.	Everyone on the Tech Team should attend the Kanban Meetings.	A tag for the different components of Adminly.		

5.3 New Kanban Structure

1. Kanban Events

a. Kanban Planning

At the beginning of the project, the Kanban Planning happened every two weeks when the team managed to plan the work for two weeks of development. There was no estimate associated with the tasks, but the idea was to create small tasks to shorten the time a card was on the board, facilitating the workflow.

It was not mandatory to finish all the planned cards in two weeks, but it was crucial to work at a good pace and to guarantee the quality of what was delivered. Other than that, it was also essential to respect the priority defined in Kanban Planning.

In the last improvement cycle, it was decided that the Kanban Planning would happen weekly and alongside the Review Meeting. So, every Monday, the two teams would start the meeting by reviewing what was delivered last week and planning and prioritizing the work for the next week.

b. Kanban Meetings

Kanban Meetings continued to happen twice a week, as decided in the first improvement cycle. With the tests for what would fit best the team availability, it was established to have a fixed time to happen every Wednesday and Friday, as the team noticed that the participation of everyone from the Tech Team was fundamental for the week’s status.

c. Review Meeting

The Review Meeting, which in the beginning happened every two weeks with the Retrospective Meeting, then, when the Retrospective became monthly, it started to happen alone on Fridays. In the last cycle, as already mentioned, it was integrated into the Planning Meeting, happening every Monday.

d. Retrospective Meeting

In the beginning, the Retrospective Meeting happened every two weeks to adjust the Kanban structure to the team’s reality. After three cycles, the team decided to change it to happen every month, and it continued to add value and improvements to the team processes. The communication and the Kanban System were already well established, reducing the need to meet biweekly.

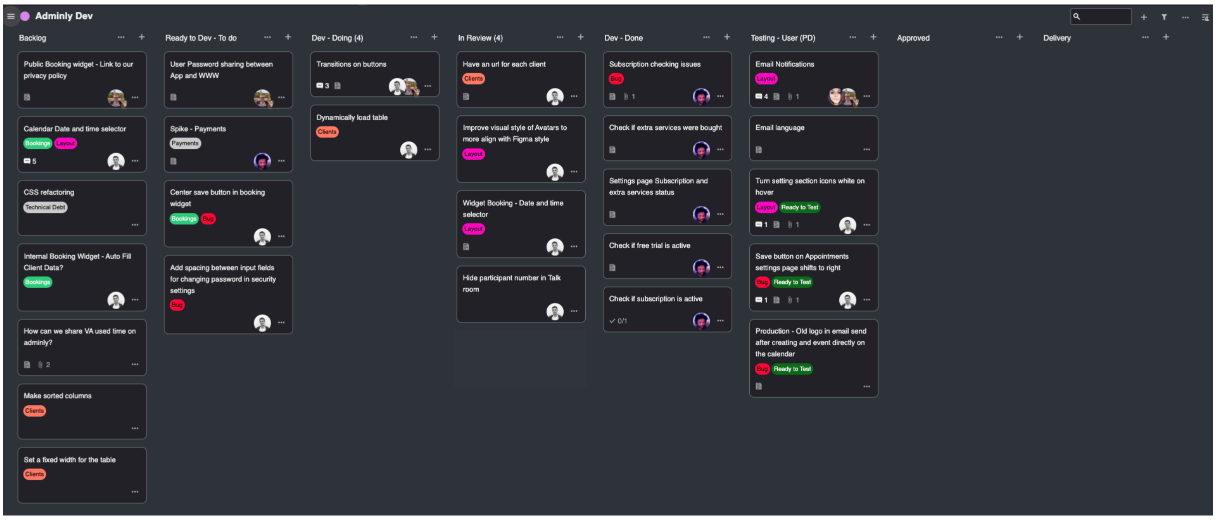
2. Kanban Board

Regarding the Kanban Board, presented in Figure 32 below, some changes happened to improve the visibility and to attend to the changes in the software development process.

The first change was adding the “In Review” column between “Dev–Doing” and “Dev–Done”. So, a card only moves to the “Dev–Done” column when it is developed and approved by the Tech Team.

The second change was the rule of the “DevOps” tasks, which could move from Dev–Done to Delivery when finished by the Tech Team, as it does not need to be approved by the Product Team.

The third change was the creation of tags to facilitate the visualization, prioritization, and management of the tasks.



Adminly Dev

Backlog
- Public Booking widget - Link to our privacy policy
- Calendar Date and time selector — Bookings, Layout — 5
- CSS refactoring — Technical Debt
- Internal Booking Widget - Auto Fill Client Data? — Bookings
- How can we share VA used time on adminly? — 2
- Make sorted columns — Clients
- Set a fixed width for the table — Clients

Ready to Dev - To do
- User Password sharing between App and WWW
- Spike - Payments — Payments
- Center save button in booking widget — Bookings, Bug
- Add spacing between input fields for changing password in security settings — Bug

Dev - Doing (4)
- Transitions on buttons — 3
- Dynamically load table — Clients

In Review (4)
- Have an url for each client — Clients
- Improve visual style of Avatars to more align with Figma style — Layout
- Widget Booking - Date and time selector — Layout
- Hide participant number in Talk room

Dev - Done
- Subscription checking issues — Bug — 1
- Check if extra services were bought
- Settings page Subscription and extra services status
- Check if free trial is active
- Check if subscription is active — ✓ 0/1

Testing - User (PD)
- Email Notifications — Layout — 4, 1
- Email language
- Turn setting section icons white on hover — Layout, Ready to Test — 1, 1
- Save button on Appointments settings page shifts to right — Bug, Ready to Test — 1, 1
- Production - Old logo in email send after creating and event directly on the calendar — Bug, Ready to Test

Approved

Delivery

Figure 32 - Kanban Board

With that said, the policies of the Kanban are:

- Backlog: the work from the Product Team to be developed by the Tech Team;
- Ready to Dev–To Do: In this column are the cards containing the week’s work defined and described during the Planning Meeting. The Ready to Dev is also the commitment point of the Kanban Board;
- Dev–Doing: Tasks that a developer of the Tech Team is developing. The limit of WIP for this column is four cards;
- In Review: Tasks that were developed and are being reviewed by a member of the Tech Team. The limit of WIP for this column is also four cards;

- Dev–Done: Cards developed and approved by the Tech Team. Every Thursday, the cards are released in the testing environment, and the feature is available to be tested by the Product Team;
- Testing–User: In this column, the Product Team tests the features. There is no limit, as all the approved tasks are moved to the "Approved" column during the Review Meeting.
- Delivery: The card will move to the "Delivery" column when deployed to production, which means it is available to the final client. It only happened three times in the six months of the project, when the project achieved its milestones.

5.4 Kanban Metrics

The selected metrics to measure the performance of the Kanban were the Lead Time, Throughput, or Delivery Rate, and the WIP, or work in progress, based on Anderson and Carmichael (2016).

The Delivery Rate of this project was 152 cards delivered from March 8th to August 16th, as presented in Table 3 below. When the project was delivered, it represented around 25 cards per month of work. The project took 23 weeks to finish, which means about seven cards per week. There were weeks when the team delivered more cards and others when part of the team was on vacation or unavailable because of other demands, which reduced the delivery rate. Still, on average, the team delivered seven cards per week.

Table 3 - Metrics analysis and results

Metric	Analysis	Result
WIP	Analyzed during the Kanban Meeting	Limit of 4 WIP was respected during the project
Delivery Rate	Number of cards per period.	-152 features delivered during the project; - 25 features/month; - 7 features/week.
Work Age Item	Analyzed during the Kanban Meeting to prioritize the work	The metric was used to prioritize the work.
Lead Time	Lead Time = "Approved" date – "Ready to Dev–To Do" date	Unable to calculate due to a problem on the NextCloud database

The Work Age Item was analyzed during the Kanban Meetings when we could prioritize the cards that were taking longer to be finished on the Board. As the team had meetings Mondays, Wednesdays, and Fridays and discussed the blockers, it was possible to visualize the cards that were on the Board for more than usual and ask them to finish them first and then move to the newer cards.

Most of the cards were moving fast through the Board. They were moved to the commitment point on Monday and were delivered on Friday or the following week. In the last two months of work, more cards needed to be added to the "Ready to Dev–To Do" column in Wednesday's Kanban Meeting.

One thing that was possible to understand with the Kanban Board was that it is crucial to define well when planning the task what is the definition of done or the list of requirements for the task to be considered finished. The work flows better when big tasks are broken into more than one card. If the card has too many requirements, it will probably be blocked at some point. In one case, the list of requirements was too long, and it was not approved by the Product Team twice, and every time it was returned, the list was increased. This card took 35 days to be finished and after those two things started to be analyzed: first, when planning the activity, make sure that it is not too big to be delivered in a week of work; second, when a card is not approved, and it needs to increase the list of requirements to finish it, another card will be created instead of increasing the size of the task.

The WIP was analyzed in every Kanban Meeting. It was required from the Tech team needed to respect the limit of WIP defined at the beginning of the project and it could be changed if it was blocking the flow of the work, after analysis. The WIP limit worked well, and was exceeded only three times during the project, two in the "Dev–Doing" column and one in the "In Review" column. Two of the three situations that happened after the vacation of one of the members of the Tech Team were rapidly solved after the Kanban Meeting.

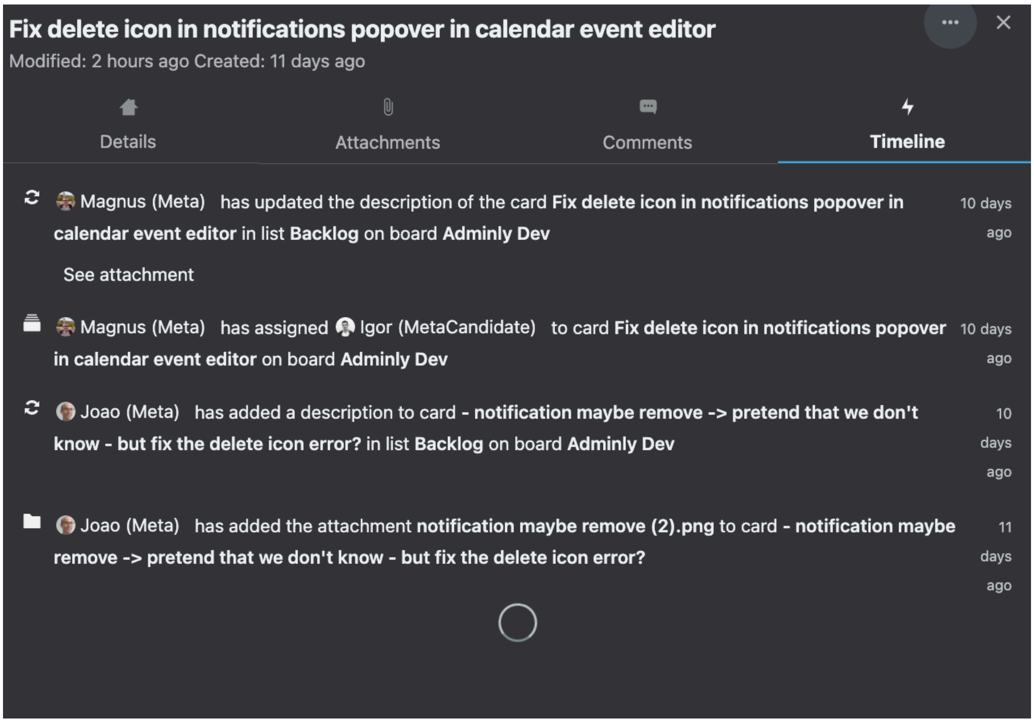
The Lead time would be calculated by using the formula:

Lead Time = "Approved" date – "Ready to Dev–To Do" date

As it was mentioned in session 4.4, the features would only be delivered or deployed to the production instance on three specific dates, as defined in the roadmap. In this phase of development, the platform was not accessed by the final customer, and it has no impact on them. For the next phase, it will probably change.

At the end of the project, the data was to be extracted from the NextCloud, by consulting the activity function, as shown in Figure 33 below. But in the recent cards, this data was not presented in the card activity. Only a few cards had the movement between columns registered.

The Tech Team investigated to see if the data could be extracted from the Data Base and they discovered that none of the movements between columns in this board were registered from May, when one of the users was removed from the team. The Tech Leader mentioned that the NextCloud could present instabilities when it has a high amount of data being used simultaneously. In the last few months, all the organizations' areas increased the use of the Deck app, which manages tasks such as the Kanban Board.



Fix delete icon in notifications popover in calendar event editor
Modified: 2 hours ago Created: 11 days ago

Magnus (Meta) has updated the description of the card **Fix delete icon in notifications popover in calendar event editor** in list **Backlog** on board **Adminly Dev** — 10 days ago

See attachment

Magnus (Meta) has assigned Igor (MetaCandidate) to card **Fix delete icon in notifications popover in calendar event editor** on board **Adminly Dev** — 10 days ago

Joao (Meta) has added a description to card **- notification maybe remove -> pretend that we don't know - but fix the delete icon error?** in list **Backlog** on board **Adminly Dev** — 10 days ago

Joao (Meta) has added the attachment **notification maybe remove (2).png** to card **- notification maybe remove -> pretend that we don't know - but fix the delete icon error?** — 11 days ago

Figure 33 - Deck activity function

Most of the concepts of the metrics were analyzed during the Kanban Meetings to make the tasks flow better. In this phase, the metrics did not impact the use of the Kanban Methodology. The lack of Lead Time may impact the estimative in the next phase when the final client will use the features. For that, if the team feels this estimate is necessary, the Lead Time will be calculated by a parallel Excel spreadsheet or by implementing another platform for the Kanban. As most of the cards were delivered within a week, especially in the last two months of the project when the team was already used to the Kanban Method and had improved the Planning Meetings, this lack of information may not impact the team.

5.5 Time and Budget

The first customer was on boarded to the system on July 1st, 2022, after the first production release. This first user was on boarded to add a customer view and redirect the platform development to their actual use. At this point, it is possible to consider that the MVP was functional, but the interface was not according to the prototype, and the features were still basic to be sold to new clients.

As presented in the third chapter, the roadmap for the project was from February 21st to August 16th, 2022. The development started on March 8th and finished in the second release on August 28th. Considering the time of development, the project was delivered on time, but if we consider the date of August 16th, the project was delivered with a delay of two weeks.

Regarding the budget, from the 400,000 euros, the organization still has 58,000 euros left to spend on the improvements of the MVP. So, the development of Admily cost around 342,000 euros, 85.5% of the planned budget, as presented in Table 4, representing a saving of 14.5% of the total budget.

Table 4 - Time and Budget Results

Metric	**Planned**	**Real**	**% Real/Planned**
Time	6 months	6 months	100%
Budget	400,000 €	342,000 €	85.5%

6. Conclusions and Future Work

To conclude this master's thesis project, this last chapter will be presented if the project achieved its objectives and if it answered the research question, what were the limitations of developing the project, and suggestions for future work.

It is crucial to consider the research question and the main objective of this thesis project to analyze its outcome. The research question was: "Which agile methodology can better fit the presented startup environment with a low budget to help deliver an MVP in six months?". Regarding the objectives, the main goal was to implement an agile methodology to guarantee that the Minimum Viable Product of Adminly will be delivered on time and within budget by the end of phase two, starting in February and ending in August 2022.

6.1 Conclusions

The project started with the literature review as the first step to decide which agile methodology would fit better in the research context, being a startup with limited resources and budget. Another point presented was the team and the context of the project. It is a small team, and the project is not so complex, as it would use an open-source tool, the NextCloud, as the basis of the Adminly platform.

The Kanban methodology was implemented after being decided with the team, considering the size of the team, the size of the project, the time commitment of the team members, and the flexibility that the method offers when compared with Scrum, the second option. It was crucial to the success of this project to the involvement of the Product and Tech teams since the methodology selection and their participation in how to use it.

The retrospective meetings were shown to be fundamental to the improvement of the methodology, being also the feedback needed for the Action Research, as a way of finding out how to make the method fit in the team context and help the delivery of the platform on time and budget. The main changes to the proposed Kanban System happened after two first retrospectives.

The Kanban Method helped the improvement of the communication between the two teams involved, and it was clear since the project started, as well as the visual aspect of it. The board organization made the tasks flow at a good pace, considering the time we had to deliver the project.

Communication and development also improved between the members of the Tech Team. They used part of the Kanban Meetings to identify how they could help each other finish the tasks and deliver a good quality of work.

In the organization, it was clear that the Tech and Product Teams were working in a more structured and organized way and that communication flowed better between the two teams. Because of the synchronism of the two teams, I was invited to prepare a workshop for the rest of the organization to help them improve their task management on the Deck App with the concepts of the Kanban Method. The Operations Team started using the Deck and cards more to manage their tasks after that.

The NextCloud and Deck have proven that it does not need to have the most robust platform to use Kanban, create boards based on processes, and improve the platform. In a physical office, it could be done on a physical whiteboard. Regarding the visual aspect of the Kanban Method, the Deck worked well, supporting the flow of tasks throughout the process.

A giant bug compromised the metrics on the NextCloud, which made the platform not register the dates that the cards moved through the board. At the same time, the Deck worked well for the board and visualization, but it proved not to be reliable to store the card data. Because of that, in future works, the suggestion is a parallel spreadsheet to calculate the metrics. But, in this specific project, the lack of metrics did not impact the project, as it was the cards were constantly analyzed during the meetings, project was not so complex, and the team was small.

Creating small tasks and breaking big tasks into more than one card makes the tasks flow faster on the board.

Regarding the time and budget, the project was delivered under budget, with an economy of 14.5%, and it had a duration of 6 months of work, as planned. There are already customers testing the platform, as it is already functional.

The flow of communication created and the structure of meetings provided by the Kanban Method helped the teams to deliver the Adminly platform on time and budget. With the Kanban System, we could constantly analyze and improve the process and manage the changes, aligning everyone involved in the project.

The flexibility of the Kanban Method fit nicely with the team, as after each retrospective, we could improve the process to attend to the team's needs. After three retrospectives, we had a structure that was working well, and it was already part of the team's routine. After that, only minor changes were made to Kanban. The team adapted well and will continue using the methodology to improve the platform in the next development phase.

The main contribution of this master thesis is to have proved that the Kanban Method is effective to manage software development projects. Also, as we did not find any literature regarding the development of an MVP using the Kanban method, this work may contribute to this area of knowledge and be useful for other researchers and practitioners doing similar applications.

6.2 Limitations

The two main limitations of this thesis project are time and budget. Those two limitations were the ones that most impacted the project. The organization where the project takes place, MetaProvide, had a budget of 400,000 €. This budget was calculated to last around six months to cover all the organization's costs.

Another limitation was the difficulty of hiring another developer for the project. During the whole phase of development, the position was posted, and the team just found a good candidate for this role in August. Due to budget limitations and the type of technology MetaProvide uses, it wasn't easy to fulfill this position. When one of the developers needs to be absent for vacations or other reasons, it had a high impact on the deliveries of the week.

As NextCloud was already implemented as a collaborative tool of work for the teams and the organization has as values the use of open-source and protect the data privacy, it limited the selection of other platforms that are more used to work in agile software development, such as Jira and Trello. This limitation impacts the metrics of the Kanban, but for the visualization of the Kanban Board and for conducting the meeting, as it was not a very complex project and the team was small, it worked well.

6.3 Future Work

MetaProvide is a startup that has just started developing its products and aims to impact people's lives. Implementing the Kanban is just the start of implementing high-quality processes supporting the organization's growth and sustainability. The Kanban method will continue to be used and continuously improved in software development, and part of its concepts is also being applied to the rest of the organization.

A suggestion for future work is the implementation of the Kanban Method for other processes of the organization, especially in processes related to the Product and Tech teams, as in the development of future products.

For future research, I suggest implementing new ways of measuring the Lead Time and to study the real impact of this metric to this project, if the metric would impact the delivery time of the next phases of product development or not. After the implementation of Kanban to the other processes of the organization, the study could compare it to the use of Kanban for software development.

The Kanban Method started to be implemented in the Product Team for the Discovery Process, with a Kanban Board for the process, facilitating the visualization of the flow from ideation to the definition of a requirement to be developed for the platform. This process needs to be improved, and the creation of a system of meetings and a channel of communication, such as the one for software development, needs to be created and implemented in this process.

As the NextCloud platform was not reliable in the registration of the data regarding the cards, one suggestion is the implementation of commercial software, such as Jira or Trello, for the Kanban Board. As it is against the organization's values, another suggestion is the parallel follow-up of the metrics in an Excel spreadsheet. The suggestion is if the organization gets a more complex project, as the lack of metrics did not impact this project.

Referências Bibliográficas

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137, 96-113.

Alshamrani, A., & Bahattab, A. (2015). A comparison between three SDLC models: waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues* (IJCSI), 12(1), 106.

Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business.* Blue Hole Press.

Anderson, D. J., & Carmichael, A. (2016). *Essential kanban condensed*. Blue Hole Press.

Angioni, M., Carboni, D., Pinna, S., Sanna, R., Serra, N., & Soro, A. (2006). Integrating XP project management in development environments. *Journal of Systems Architecture*, 52(11), 619-626.

Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. S. (2017). Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2), 1-10.

Beck, K. (1999). *Embracing change with extreme programming. Computer*, 32(10), 70-77.

Beck, K. (2000). *Extreme programming explained: embrace change.* addison-wesley professional.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001a). Agile Manifesto - Principles. Retrieved December 9, 2018, from https://agilemanifesto.org/principles.html

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001b). Agile Manifesto - Values. Retrieved December 9, 2018, from https://agilemanifesto.org/

Bergmann, T., & Karwowski, W. (2018, July). *Agile project management and project success: A literature review. International Conference on Applied Human Factors and Ergonomics* (pp. 405-414). Springer, Cham.

Bredillet, C., Tywoniak, S., & Dwivedula, R. (2015). What is a good project manager? An Aristotelian perspective. *International Journal of Project Management,* 33(2), 254-266.

Bin, K. J., Higa, N., da Silva, J. H., Quagliano, D. A., Hangai, R. K., Cobello-Júnior, V., ... & Ono, S. K. (2021). Building an outpatient telemedicine care pilot using Scrum-like framework within a medical residency program. *Clinics*, 76.

Boehm, B. (2007). A survey of agile development methodologies. *Laurie Williams*, 45, 119.

Chadli, S. Y., Idri, A., Ros, J. N., Fernández-Alemán, J. L., de Gea, J. M. C., & Toval, A. (2016). Software project management tools in global software development: a systematic mapping study. *SpringerPlus*, 5(1), 1-38.

Coughlan, P., & Coghlan, D. (2002). Action research for operations management. *International journal of operations & production management.* Vol. 22, No. 2, 2002, pp. 220-240**.**

Conforto, E. C., Salum, F., Amaral, D. C., Da Silva, S. L., & De Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development?. *Project Management Journal*, 45(3), 21-34.

COVID, H. (19). has pushed companies over the technology tipping point—and transformed business forever. *McKinsey & Company (October, 2020). Available online:https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever (accessed on 10 Aug 2022).*

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of systems and software*, 85(6), 1213-1221.

Fabris,N.(2022).Impact of Covid-19 Pandemic on Financial Innovation, Cashless Society, and Cyber Risk. ECONOMICS,10(1) 73-86. https://doi.org/10.2478/eoik-2022-0002

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software development*, 9(8), 28-35.

Garel, G. (2013). A history of project management models: From pre-models to the standard models. *International Journal of Project Management*, 31(5), 663-669.

Holtzhausen, N., & de Klerk, J. J. (2018). Servant leadership and the Scrum team's effectiveness. Leadership & Organization Development Journal, 39(7), 873-882.

Hron, M., & Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *Journal of Systems and Software*, 183, 111110.

Hlaing, S. Z., & Ochimizu, K. (2018, December). An integrated cost-effective security requirement engineering process in SDLC using FRAM. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 852-857). IEEE.

J. Highsmith, *Agile Software Development Ecosystems*. Boston, MA: Addison-Wesley, 2002.

Kim, S. K. (2015). Lean initiative practice for supplier developments in Philippines. *International Journal of Lean Six Sigma.*

Lee, S., & Geum, Y. (2021). How to determine a minimum viable product in app-based lean startups: Kano-based approach. *Total Quality Management & Business Excellence*, 32(15-16), 1751- 1767.

Lindstrom, L., & Jeffries, R. (2003). Extreme programming and agile software development methodologies. In *IS management handbook* (pp. 531-550). Auerbach Publications.

Melton, T. (2005). The benefits of lean manufacturing: what lean thinking has to offer the process industries. *Chemical engineering research and design*, 83(6), 662-673

Morandini, M., Coleti, T. A., Oliveira Jr, E., & Corrêa, P. L. P. (2021). Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams. *Computer Science Review*, 39, 100314.

Mundra, A., Misra, S., & Dhawale, C. A. (2013, June). Practical scrum-scrum team: Way to produce successful and quality software. In 2013 *13th International* Conference on Computational Science and Its Applications (pp. 119-123). IEEE.

Nakazawa, S., & Tanaka, T. (2016, July). Development and application of Kanban tool visualizing the work in progress. In 2016 *5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)* (pp. 908-913). IEEE.

Ohno, T., & Bodek, N. (2019). *Toyota production system: beyond large-scale production*. Productivity press.

Parsons, D., Thorn, R., Inkila, M., & MacCallum, K. (2018, December). Using Trello to support agile and lean learning with Scrum and Kanban in teacher professional development. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 720-724). IEEE.

PMI. (2017). *A guide to the project management body of knowledge* (PMBOK ® guide). Project Management Institute. (PMI, Ed.) (6th ed.). Newtown Square, Pennsylvania 19073-3299 USA.

PMI. (2021). *A guide to the project management body of knowledge* (PMBOK ® guide). Project Management Institute. (PMI, Ed.) (7th ed.). Newtown Square, Pennsylvania 19073-3299 USA.

Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: an agile toolkit*. Addison-Wesley.

Saunders, M., Lewis, P., & Thornhill, A. (2016). *Research Methods for Business Students (Seventh)*. London: Pearson Education Limited.

Sastri, Y., Hoda, R., & Amor, R. (2021). The role of the project manager in agile software development projects. *Journal of Systems and Software*, 173, 110871.

Reich, B. H., Gemino, A., & Sauer, C. (2010). Exploring the Impact of Knowledge Management on Business Value in IT Projects. *European Academy of Management,* May, Rome, Italy.

Romano, B. L. & Delgado Da Silva, A.(2015) Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise, *12th International Conference on Information Technology - New Generations,* pp. 774-776.

Sánchez-Arias, L. F., & Solarte-Pazos, L. (2010). The body of knowledge of the Project Management Institute-PMBOK® Guide, and the specificities of project management: a critical review. Innovar, 20(37), 89-100.

Saynisch, M. (2010). Mastering complexity and changes in projects, economy, and society via project management second order (PM-2). *Project Management Journal*, 41(5), 4-20.

Schwaber, K., & Sutherland, J. (2020). The scrum guide. Scrum Alliance, 21(1).

Shenhar, A. J., & Dvir, D. (2007). Project management research—The challenge and opportunity. *Project Management Journal*, 38(2), 93-99.

Sugimori, Y., Kusunoki, K., Cho, F., & UCHIKAWA, S. (1977). Toyota production system and kanban system materialization of just-in-time and respect-for-human systems. *The International Journal of Production Research,* 15(6), 553-564.

The Scrum Framework Poster (2020), retrieved July 20, 2022 from https://www.scrum.org/resources/scrum-framework-poster

Todorović, M. L., Petrović, D. Č., Mihić, M. M., Obradović, V. L., & Bushuyev, S. D. (2015). Project success analysis framework: A knowledge-based approach in project management. *International Journal of Project Management*, 33(4), 772-783.

Vaara, E., Harju, A., Leppälä, M., Buffart, M., How to Successfully Scale a Flat Organization. *Harvard Business Review,* retrieved October 05, 2022 from https://hbr.org/2021/06/how-to-successfully-scale-a-flat-organization.

Vogelzang, J., Admiraal, W. F., & van Driel, J. H. (2019). Scrum methodology as an effective scaffold to promote students' learning and motivation in context-based secondary chemistry education. *EURASIA Journal of Mathematics, Science and Technology Education,* 15(12), em1783.

Womack, J.P., Jones, D.T. and Roos, D. (1990), The Machine That Changed The World, Free Press, New York, NY.

Yuliadi, B., & Nugroho, A. (2019). Integration between management capability and relationship capability to boost supply chain project performance. *Int. J. Supply Chain Manag,* 8(2), 241-252.

Zasa, F. P., Patrucco, A., & Pellizzoni, E. (2020). Managing the Hybrid Organization: How Can Agile and Traditional Project Management Coexist?. *Research-Technology Management*, 64(1), 54-63.

Zhong, S., Liping, C., & Tian-en, C. (2011, March). Agile planning and development methods. In *2011 3rd International Conference on Computer Research and Development* (Vol. 1, pp. 488-491). IEEE.

Appendix 1 – Kanban Presentation

KANBAN METHOD

Danna Dias - Agile Project Manager
MetaProvide

Figure 34 - Kanban Presentation

SUMMARY OF CONTENTS

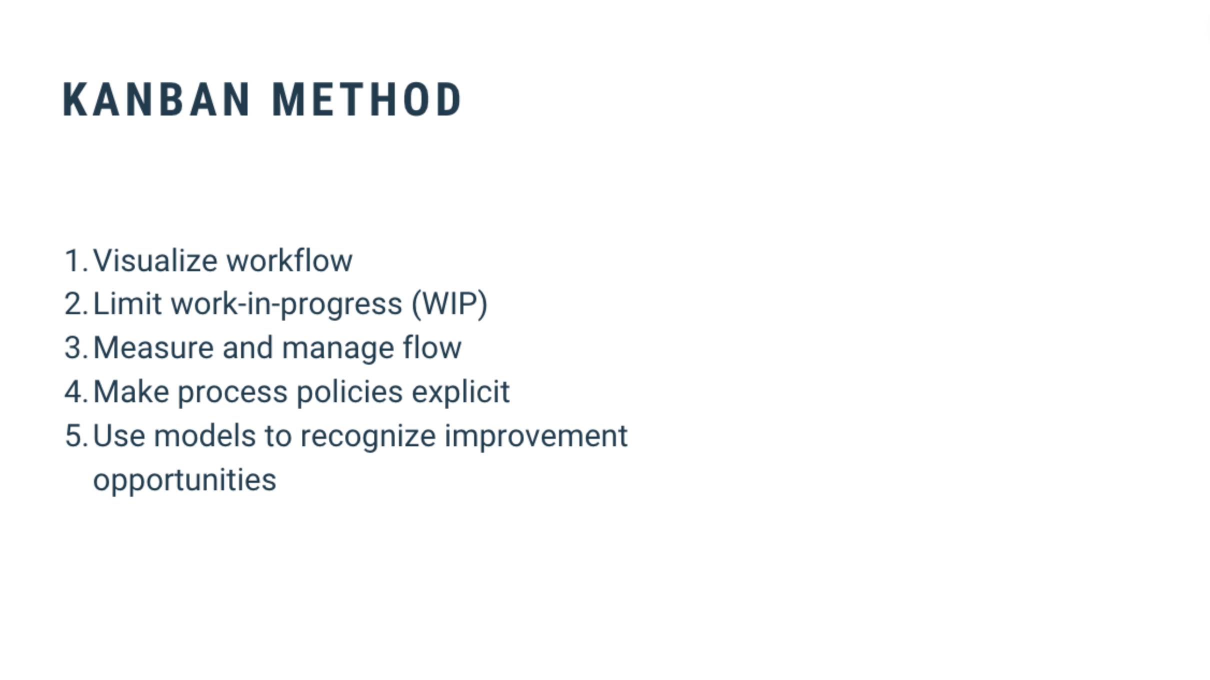
OUR MAIN TOPICS TODAY

1. Kanban Method : What is it?
2. Boards
3. Roles
4. Metrics
5. Roadmap

Figure 35 - Kanban Presentation - Summary

WHAT IS KANBAN METHOD?

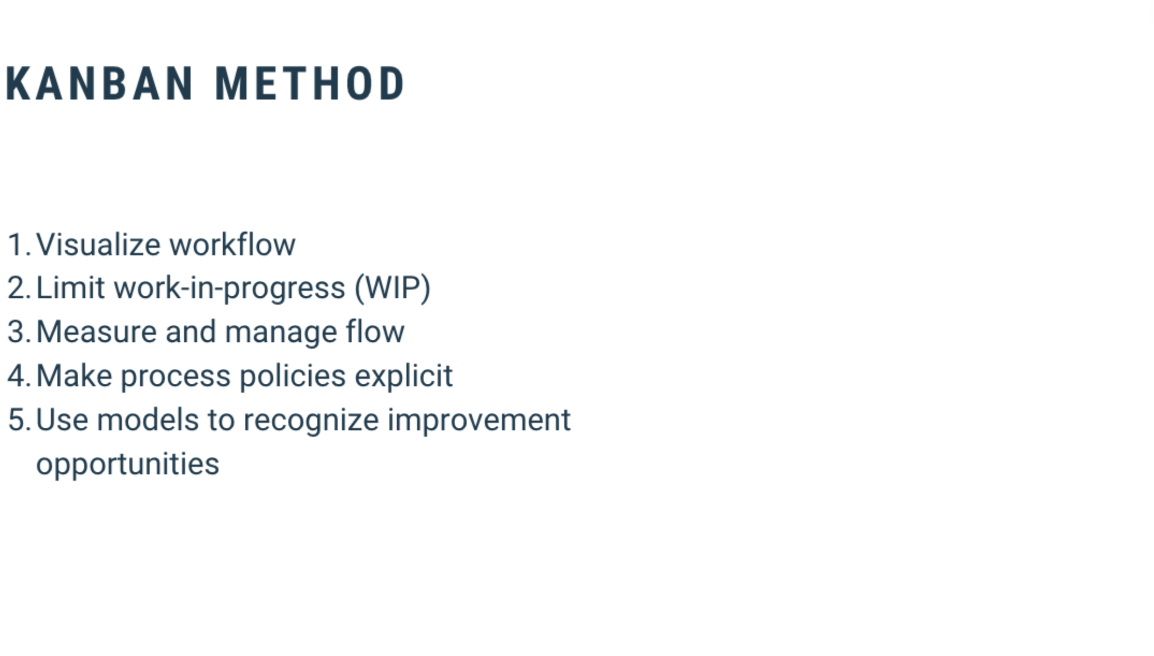
Figure 36 - Kanban Presentation - What is Kanban Method?



KANBAN METHOD

1. Visualize workflow
2. Limit work-in-progress (WIP)
3. Measure and manage flow
4. Make process policies explicit
5. Use models to recognize improvement opportunities

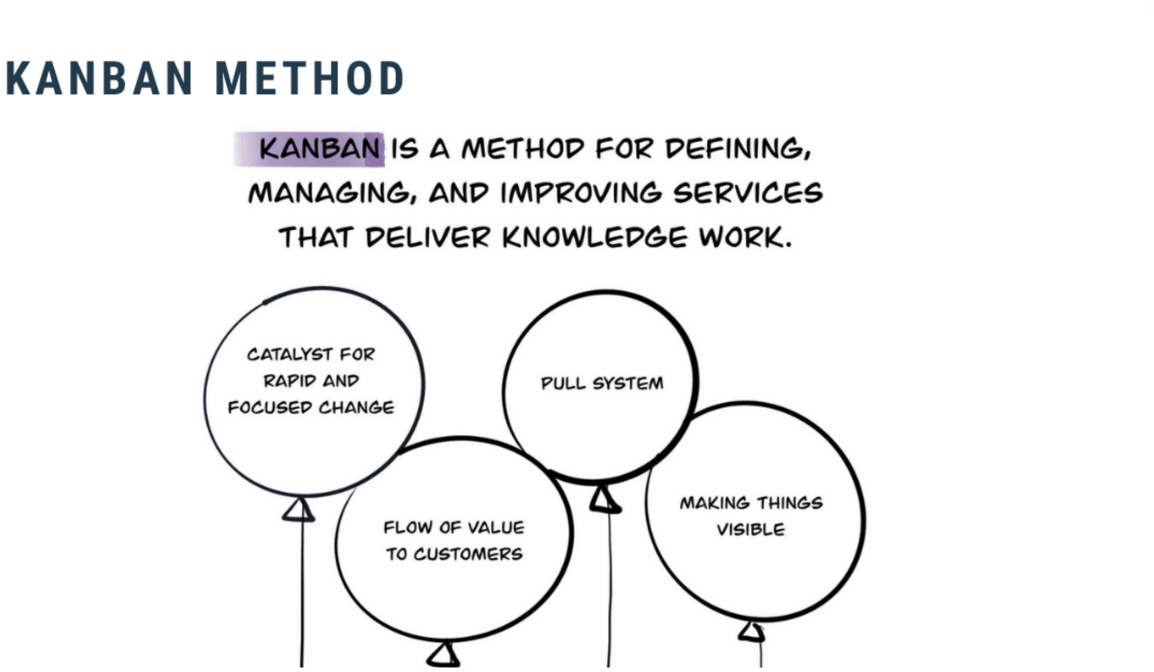
Figure 37 - Kanban Presentation - Kanban Method



KANBAN METHOD

1. Visualize workflow
2. Limit work-in-progress (WIP)
3. Measure and manage flow
4. Make process policies explicit
5. Use models to recognize improvement opportunities

Figure 38 - Kanban Presentation - Kanban Method Benefits



KANBAN METHOD

KANBAN IS A METHOD FOR DEFINING, MANAGING, AND IMPROVING SERVICES THAT DELIVER KNOWLEDGE WORK.

CATALYST FOR RAPID AND FOCUSED CHANGE
PULL SYSTEM
FLOW OF VALUE TO CUSTOMERS
MAKING THINGS VISIBLE

Figure 39 - Kanban Presentation - Kanban Method Benefits 2

KANBAN

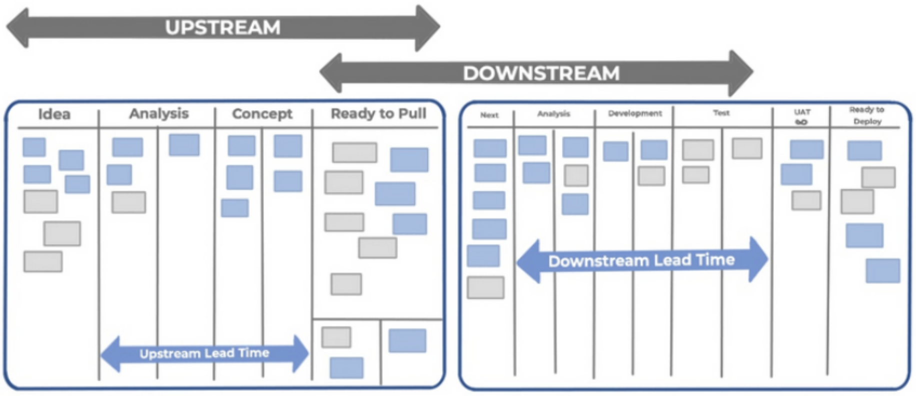
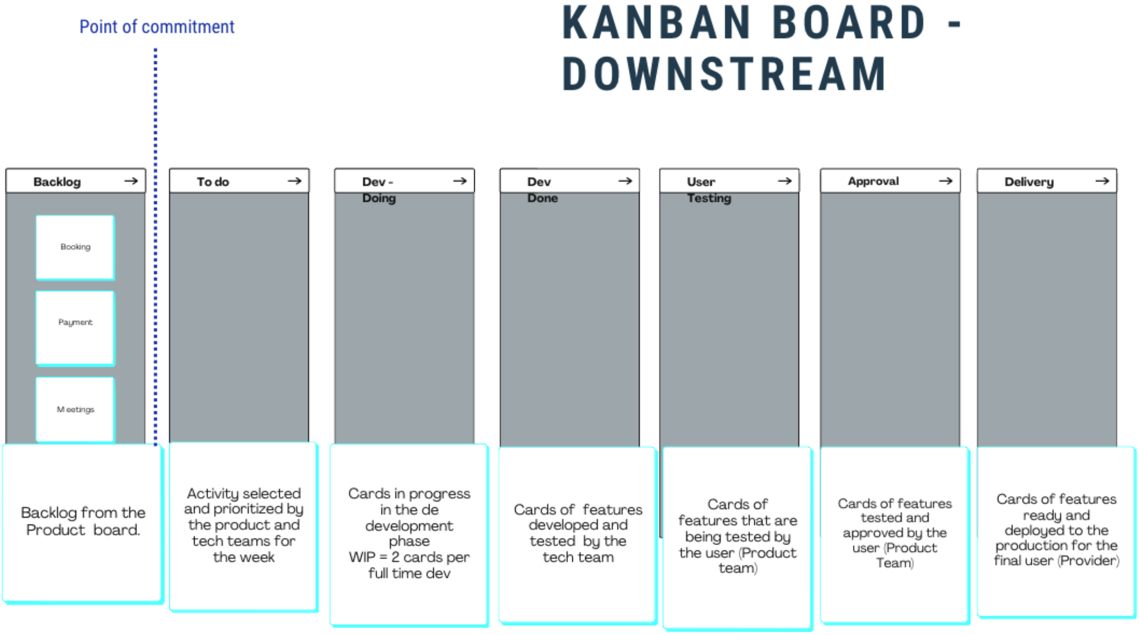


Figure 40 - Kanban Presentation - Kanban board

KANBAN BOARD - DOWNSTREAM



Backlog from the Product board.

Activity selected and prioritized by the product and tech teams for the week

Cards in progress in the de development phase
WIP = 2 cards per full time dev

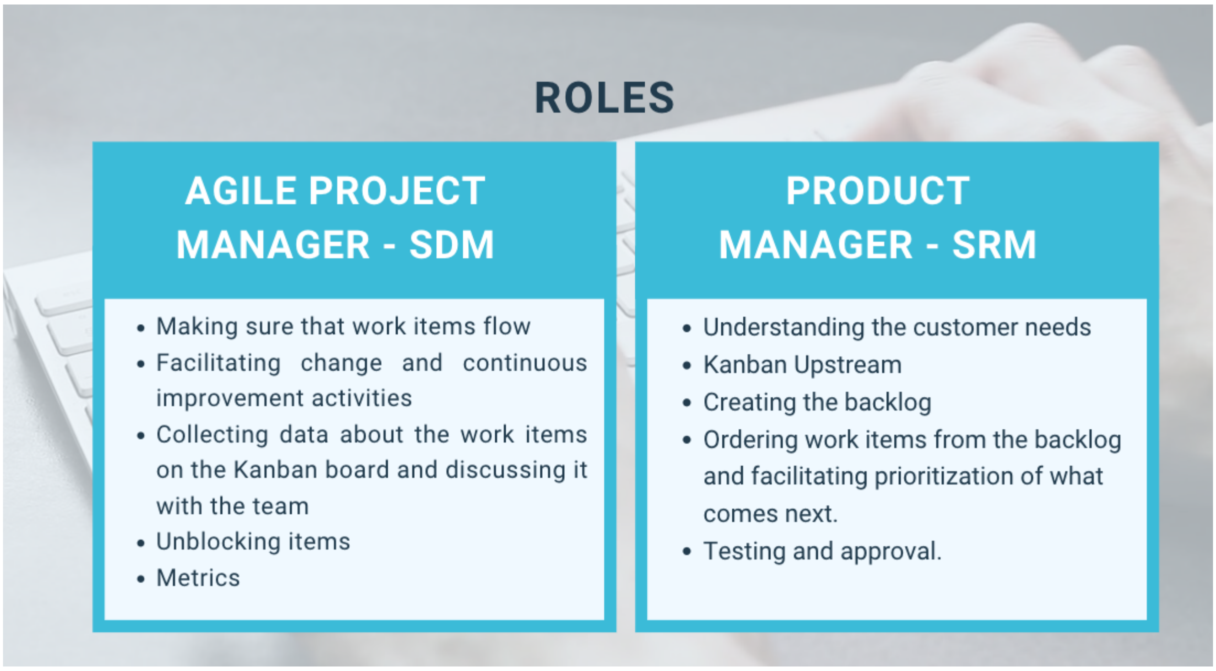
Cards of features developed and tested by the tech team

Cards of features that are being tested by the user (Product team)

Cards of features tested and approved by the user (Product Team)

Cards of features ready and deployed to the production for the final user (Provider)

Figure 41 - Kanban Presentation - Proposed Kanban Board



ROLES

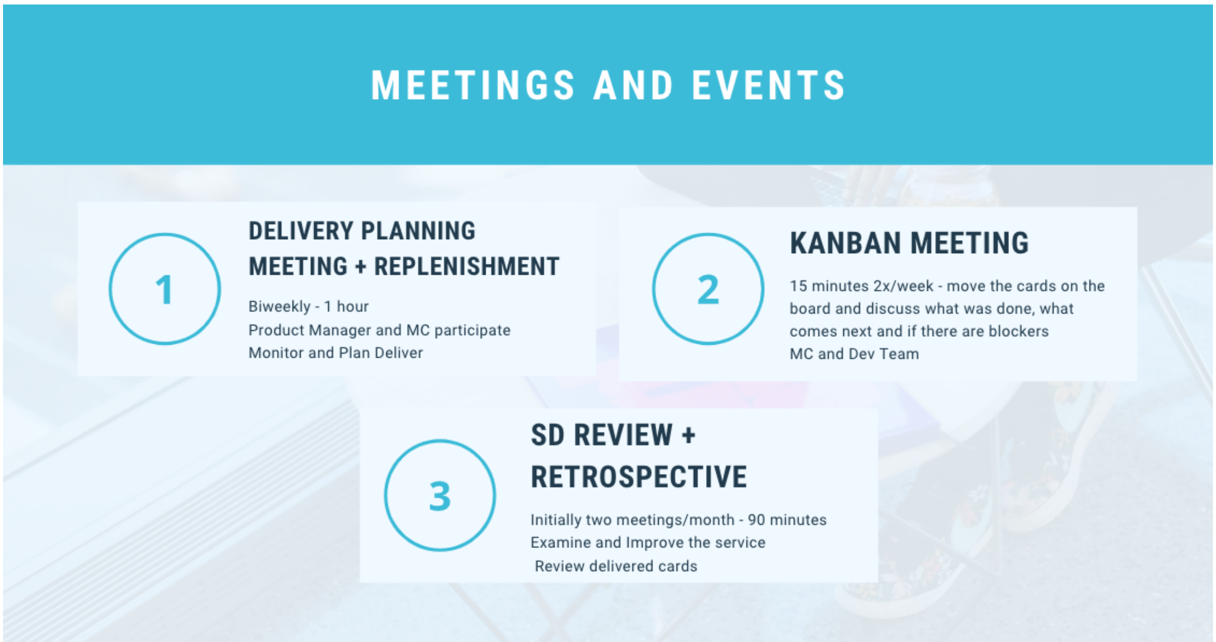
AGILE PROJECT MANAGER - SDM

- Making sure that work items flow
- Facilitating change and continuous improvement activities
- Collecting data about the work items on the Kanban board and discussing it with the team
- Unblocking items
- Metrics

PRODUCT MANAGER - SRM

- Understanding the customer needs
- Kanban Upstream
- Creating the backlog
- Ordering work items from the backlog and facilitating prioritization of what comes next.
- Testing and approval.

Figure 42 - Kanban Presentation - Roles



MEETINGS AND EVENTS

1 DELIVERY PLANNING MEETING + REPLENISHMENT

Biweekly - 1 hour
Product Manager and MC participate
Monitor and Plan Deliver

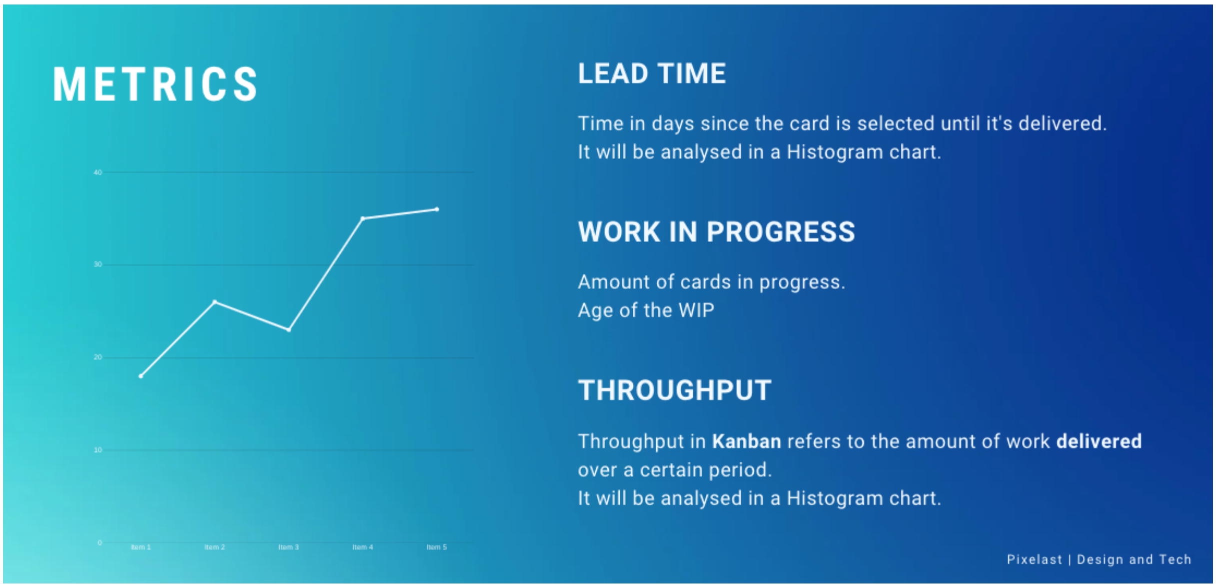
2 KANBAN MEETING

15 minutes 2x/week - move the cards on the board and discuss what was done, what comes next and if there are blockers
MC and Dev Team

3 SD REVIEW + RETROSPECTIVE

Initially two meetings/month - 90 minutes
Examine and Improve the service
Review delivered cards

Figure 43 - Kanban Presentation - Events


40
30
20
10
0
Item 1
Item 2
Item 3
Item 4
Item 5

METRICS

LEAD TIME

Time in days since the card is selected until it's delivered.
It will be analysed in a Histogram chart.

WORK IN PROGRESS

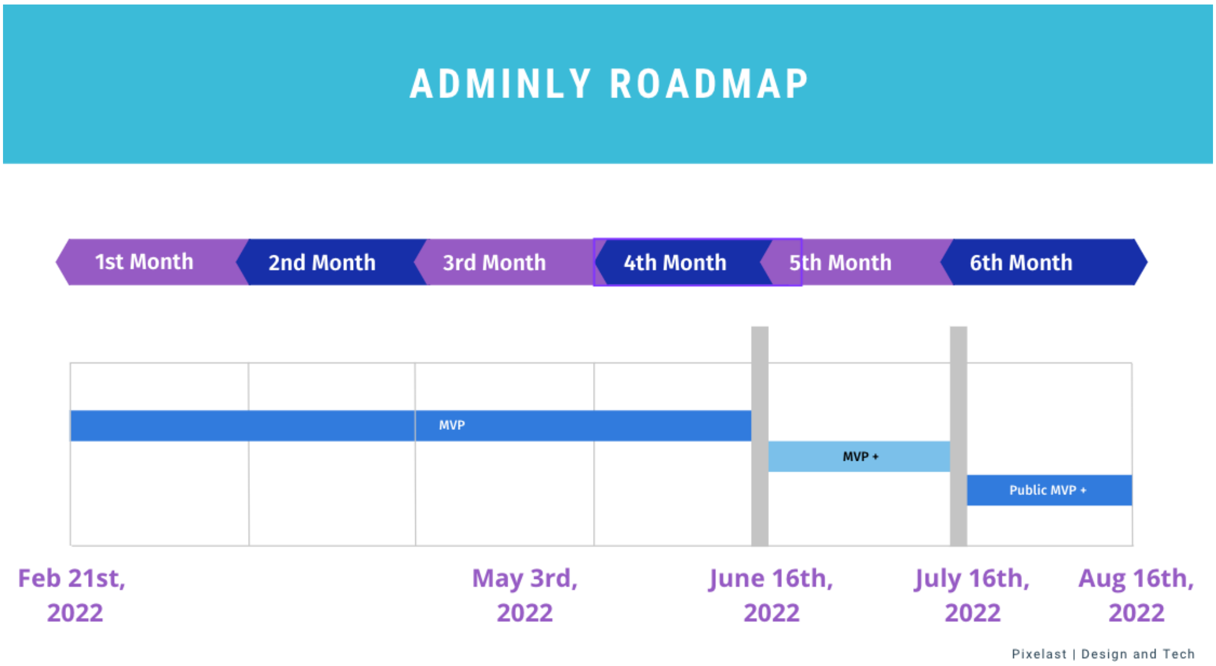
Amount of cards in progress.
Age of the WIP

THROUGHPUT

Throughput in **Kanban** refers to the amount of work **delivered** over a certain period.
It will be analysed in a Histogram chart.

Pixelast | Design and Tech

Figure 44 - Kanban Presentation - Metrics



ADMINLY ROADMAP

1st Month
2nd Month
3rd Month
4th Month
5th Month
6th Month

MVP
MVP +
Public MVP +

Feb 21st, 2022
May 3rd, 2022
June 16th, 2022
July 16th, 2022
Aug 16th, 2022

Pixelast | Design and Tech

Figure 45 - Kanban Presentation - Adminly Roadmap