MESG

MESTRADO EM ENGENHARIA DE SERVIÇOS E GESTÃO

Analysis and Improvement of Software Development and Project Management Processes

Manuel Rui Fernandes

Master Thesis

Supervisor at FEUP: Prof. Américo Azevedo

Supervisor at TBFiles: Eng. Álvaro Magalhães

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

2020-07-05

Ao meu filho

Abstract

The present dissertation was developed within the context of the completion of the master's degree in Services Engineering and Management, belonging to the Faculty of Engineering of the University of Porto. The objective of this thesis is to present a case study in which the software development and project management processes of a company certified in the provision of integrated document management solutions and administrative back-office outsourcing were analysed.

The company's main strategic objective is to achieve operational excellence in order to guarantee the company's adaptation to the dynamic commercial reality in which it operates. Thus, speeding up their processes by reducing in 20% the average time to market, and increasing their quality by reducing rework about 57% and increasing on time deliveries to 75% were defined as specific project objectives.

The project was executed in a period of five months, which was organised in four phases, each one with its intermediate goals. The processes were mapped, the inefficiencies were identified and suggestions for improvements were presented and analysed. Thus, activities such as characterising the reality of the company and the current state, quantifying current performance, defining a set of objectives, identifying the root causes of problems, developing and analysing solutions emerged from these phases.

It was concluded that the main problems with the project management and software development processes are related to the lack of visibility of the team load, lack of standardisation and inefficient management processes. These causes result in problems such as IT being unable to plan work in advance and problems associated with quality that negatively influence the lead time of projects.

Therefore, suggestions for improvement were formulated and prioritised to address each of these aspects. A more agile approach to software development and the redesign of the processes for creating customised solutions were the solutions developed. There was also a need to develop changes in the existing project management software to adapt it to these changes.

Acknowledgments

The performance of this work, despite its individual character, was impossible without the collaboration and encouragement of several people. In this way, I would like to express my thanks and gratitude to all those who allowed me to realize this dream.

To Prof. Américo Azevedo, for the scientific education, teachings, detail, and, above all, for the dedication, care and availability that he provided me throughout this work.

To Profª. Lia Patrício and the FEUP institution for all the support, collaboration along this path, but mainly for the opportunity granted.

To Eng. Álvaro Magalhães for his precious tips and professional growth for working "at your side".

To the company and especially to all those who, in a responsible and dedicated way, participated as if it were their own project.

To all my friends, especially the "Masters" Francisco, Eduardo and Miguel.

And of course, my family stands out, In particular to my mother, my support! Which always supports me unconditionally! Without you, it would be impossible!

To Márcia, for having been the driver of this good change in my life, for the endless hours of torture that my doubts provide you, your advices, helps and availability. You are definitely a reference for me.

To you son, who without realizing you are my greatest motivation.

To all who have contributed to my growth and success ... this is also yours!

… To all my sincere THANK YOU! I dedicate to you ...

Table of Contents

List of Tables

List of Figures

List of abbreviations

SD - software development

PM - project management

IT – Information Technology

SaaS - Software as a Service

CPM - Critical Path Method

PERT - Program Evaluation and Review Technique

SDP - Software Development Process

IEEE - Institute of Electrical and Electronics Engineers

PMS - Project Management Software

PMI - Project Management Institute

PMBOK - Project Management Body of Knowledge

IS – Information Systems

DLC – Development Life Cycle

PLC – Project Life Cycle

XP - Extreme Programming

BCG - Boston Consulting Group

PDCA - Plan Do Check Act

TQM - Total Quality Management

TDD - Test-Driven Development

CRM - Customer Relationship Management

EVM - Earned Value Management

1 Introduction

The present dissertation is inserted in the master's in services engineering and management developed in a specific context focused on the analysis and improvement of processes of software development (SD) and project management (PM). In this part, we want to give a general overview of the context, the project, the research objectives, the methodology in use and address the structure of the report.

1.1 Context

In response to market needs, to stand out and create a competitive advantage, many companies choose solutions based on digital transformation.

It is commonly accepted that with the growing digitalisation of "virtually everything" driven by and based on the success of the Information Technology (IT) megatrends social, mobile, analytics and cloud computing, enterprises need to pursue digital innovation to improve or change their business models. As technology change accelerates and new digital solutions emerge, many companies feel the pressure to perform a digital transformation in order not to lose their competitive advantage.

For Downs and Nunes (2013) digital transformation is the integration of digital technology into all areas of a business, fundamentally changing how companies operate and deliver value to customers. This is one of the biggest challenges that companies are facing today. No industry or organisation can stay indifferent to digital transformation. Therefore, entire business models are being rethought and reformulated.

So, technology is, of course, one of the most important elements of digital transformation and it evolved rapidly these days to support the businesses in delivering the services to customers, but often it's more about abandoning outdated legacy processes than adopting new technologies.

A study from Sebastian I. et al. (2017) confirm the ideas from previous authors and revealed three essential elements for a successful digital transformation: a digital strategy, an operational backbone, and digital service platforms.

In the opinion of Fitzgerald et al. (2013) technological trends such as social media, big data, the internet of things, mobile computing and cloud computing significantly influence processes, products, services, and business models. The impact is even greater if these technologies are combined. The consequences are almost unlimited possibilities in the innovative application of IT.

Urbach, Ahlemann and Böhmann, (2017) corroborate that the business activity gets not only more efficient, but is also no longer imaginable without IT. So, more and more companies are realising that the use of cloud computing and service-oriented architecture provides excellent and innovative opportunities. New service architectures and business models began to be widely adopted, such as Software as a Service (SaaS). This represents a shift from traditional software package license purchases to remotely "renting" specific software services where a business model is paid for usage. For Horlach, B., Drews P., and Schirmer I. (2016) these changes should allow the business to be better informed, more flexible, and faster in adapting services and products to market opportunities and customer needs.

Realising these needs for new business models and strategies is important not only for companies who are facing these changes but simultaneously to understand the challenges of companies that support this digital transformation. These types of companies, through SaaS models and customised software solutions, in addition to being responsible for the development, are responsible for the security, performance, availability and stability of the service.

However, according to Agarwal, N. and Rathod, U. (2005) in the world of software projects, success is found relatively rare. This success is assessed in terms of its ability to attain the target cost, time, and the desired level of products' quality. Therefore, a software project is successful if it delivers the product with a pre-agreed level of quality within the given time and cost.

So, the more information technologies are applied to drive business innovation, the more IT teams need to proactively, and early on, cooperate with business departments, with the clear goal of developing and implementing innovations together. Besides ensuring the regular IT operations, IT team is requested to proactively identify technological innovations and to rapidly transfer them into marketable solutions.

Experience tells us that, unfortunately, just as a technologically inferior product can win in the market due to superior marketing, a technologically superior software project can fail if PM is lacking.

So, in addition to changes in technology and the competitive environment, it is important to highlight the evolution of mentalities. According to Kerzner (2006), PM came to be a means to achieve the objectives. The same author points out that nowadays companies are under pressure to introduce new products on the market, because the product life cycles are getting shorter and shorter.

This new context creates an urgent need to adjust software development and PM processes currently used in this context.

1.2 Problem Description

The company where this project was developed is a Portuguese company, named TBFiles, with more than 15 years in the market of dematerialization services and file custody.

Over the past recent years, the company has experienced a significant growth in terms of business volume with the consolidation of its position in the domestic market.

At the same time, the company has focused on developing new services and improving the base product, offering a wide range of resources and customised solutions to maintain its competitive advantage.

Historically, the software solutions produced by TBFiles have served to automate the document management of its clients. Today, TBFiles' technological and logistical solutions also optimize processes and workflows of companies from different sectors. This company provides business process outsourcing services based on customised software solutions to ensure the digital transformation from other companies.

As we have already mentioned, the market has been pressuring companies to make this digital transformation so as not to lose their competitive advantage.

So, this increased demand and the development of customised solutions also brings other challenges for TBFiles, such as timely delivery and the compliance with the costs and requirements established with the customers.

To adapt to this "new" reality and guarantee the success of services in this constantly changing technological world, the company has focused on the need to increase the productivity and internal efficiency of the IT department and specifically in the development and management of its clients' software projects .

So, to achieve operational excellence in the development and management of this type of projects the main objective of this research was to identify problems in current processes and practices and develop possible solutions to improve.

1.3 Research Questions

The following research questions aim at guiding and focusing this project.

- Research question 1 (RQ 1) – What are the relevant metrics in the analysis of the company's software development and project management processes?
- Research question 2 (RQ 2) – What types of problems and inefficiencies are affecting the company's software development and project management processes?
- Research question 3 (RQ 3) – What are the real causes of problems identified?
- Research question 4 (RQ 4) –How to overcome the problems identified?

After understanding the motivations and nature of the project, the first of our research questions arose to enable the quantification of current performance and obtain a clear definition of the objectives of its execution.

The other questions formulated came from the main objective of this project, identifying problems, understanding their causes, questioning solutions for improvement, in order to achieve operational excellence in the software development and project management processes.

1.4 Methodology

At this stage of the report, the aim is to show how the whole project was planned and organised to answer the research questions raised.

The first phase of the project aimed to understand the operation of the entire company in terms of structure, processes, surrounding context and evolution over the past few years, through collecting information, holding informal workshops with the main stakeholders and observation of the process.

This allowed to obtain qualitative data, specifically textual descriptions of observed facts, useful for the development of a generic characterisation of Management processes. Then, narrowing the focus of the analysis on the PM and software development process (SDP), the informal workshops, the analysis of documents and information existing in the current software were used to obtain information that allowed the extensive representation and diagnosis of these processes. At the same time, the process of searching and analysing the relevant literature on the topics of the project was initiated.

In the second phase, a deductive approach through the application of concepts and methods already developed and studied by a vast set of authors in order to identify opportunities for improvement and discover their causes. At this stage, was used workshops in order to complement and enrich the information we were collecting.

In the final phase of the project, a set of solutions were designed and analysed in terms of the benefits and difficulties of their implementation with the objective of mitigating or eliminating the problems previously identified. Again, knowledge obtained from the literature consulted was used and complemented by additional workshops conducted.

To do this and to structure this project *kobetsu kaizen* was chosen as a methodology. According to Coimbra, A. (2016) this structured problem solving methodology can be used to achieve different types of objectives such as: reducing defects or errors, costs, delivery times, increasing productivity or even the motivation of employees.

So, our project went through different stages:

- Problem selection - in the first stage our main objective was to understand the motivations for the realisation of this project.
- Problem description - The purpose of this step was the characterisation of the initial situation, understand the problem and its dimension.
- Objectives definition - This step consisted of defining the improvement objective that was clear and that answered three questions: what is the indicator to achieve? What is the value to achieve? When should it be achieved?
- Root-cause analysis - We use the Ishikawa diagram to perform this analysis. This tool consists of dividing the causes into five types: labour, material, machine, method, and environment. After the causes are identified, the root causes arrive, successively asking "Why?". With this approach we can identify opportunities for improvement and find out the real causes of the problems.
- Solution of the problem - After all the causes were identified, a set of suggestions for improvement was analysed in terms of the benefits and difficulties of their implementation and the ideal solution to mitigate or eliminate the problems previously identified was designed.

1.5 Report outline

This report is structured to allow the understanding of the literature of interest related to the subject of the study, and better perception of the project.

In this introductory section is presented a brief presentation of the context and motivations for the realisation of this project, as well as the research questions that we established for it.

The second chapter presents a deeper analysis on these main subjects: PM for software projects and the most used models for software development.

The aim of the third chapter is to do the presentation of the company used as a case study divided into three subsections: the description of the evolution of the business model, the detailed characterisation of the initial situation of TBFiles' IT processes, and going deeper in

the development of customised software solution by TBFiles. Collecting data, and outlining performance objectives for this project, will also be addressed throughout the chapter.

After we have a clear picture, in this fourth chapter we will analyse the current situation, the possible root causes of the problems and the improvement opportunities.

The fifth chapter is devoted to developing possible solutions to mitigate or eliminate the negative effects of previously identified problems by comparing possible approaches and methods addressed in the second phase of this study.

Finally, chapter six discusses the project's findings and reflects on the main difficulties and suggestions for future improvements.

2 Theoretical Background

It is commonly accepted that PM activities typically take place in parallel with software development activities. For Sheetal Sharma et al. (2012) software practices are techniques for "how" to develop software products or manage software projects effectively.

To Cadle J. and Yeates D. (2008), is important to differentiate the system development life cycle (DLC) and project life cycle (PLC). According to this authors, the system DLC often covers only the technical deliverables while the PLC is concern with all aspects leading to the delivery of project's objectives, not only the technical aspects but also the associated management and quality aspects necessary to a successful project.

So, the present chapter goal is to deliver background knowledge on the subjects relevant to the dissertation such as the most used models for software development, but also the processes and the mains topics inherent to the management of software projects like the business context of projects, manage the project execution, the tasks related to delivering and the area related to managing people.

2.1 Project Management for Information Systems

According to Kerzner H. (2002), to understand PM and specifically software project management first, it is necessary to know how to recognise what a project is.

So, to Project Management Institute (PMI) (2013), in Project Management Body of Knowledge (PMBOK), a project is a temporary effort undertaken to create an exclusive product, service, or result.

Kerzner H. (2013) corroborates that the temporary nature of the projects indicates that they have a clearly defined beginning and end. According to the same author, this is an enterprise with an identifiable objective, which consumes resources and works under pressure from deadlines, costs and quality. It is then a progressive work that has a temporary, non-repetitive character and that often involves a certain degree of uncertainty.

PMI (2013) defines PM as the application of knowledge, skills, tools and techniques to project activities to meet your needs. PM is performed through the application and integration of processes grouped logically into five groups. These five groups of processes are:

- Initiation and planning - Guidelines and criteria for adapting the set of standard processes and procedures of the organisation in order to meet the specific needs of the project; Specific organisational standards such as policies (e.g., human resources policies, health and safety, ethics, and PM), product life cycles and of the project, and quality policies and procedures (eg, process audits, goals improvements, checklists, and standardised process definitions for use in organisation); and Models (eg, risk register, analytical structure of the project, network diagram of the project schedule and contract templates).

- Execution, monitoring and control - Change control procedures, including steps to change standards, the organisation's policies, plans and procedures or any project documents and how the changes will be approved and validated; Financial control procedures (for example, reporting of time, expenses and mandatory analysis expenses,

accounting codes and standard contractual clauses); Problem and defect management procedures that define your controls, identification and resolution of problems and defects and monitoring of its action items; Organisation's communications requirements (eg, specific communications technology available, authorised communication media, record retention policies and requirements of security); Procedures for prioritising, approving and issuing work permits; Risk control procedures, including risk categories, statement templates of risks, definitions of probability and impact, and matrix of probability and impact; and Standardised guidelines, work instructions, proposal evaluation criteria, and performance measurement criteria.

- Closure - Project closure guidelines or requirements (eg, lessons learned, final project audits, project evaluations, product validations and acceptance criteria).

So, PM typically includes, but is not limited to:

- Requirements definition.
- Addressing the different needs, concerns, and expectations of stakeholders in the project planning and execution.
- Establishment, maintenance, and execution of active, effective and collaborative communications between interested parts.
- Stakeholder management aimed at meeting project requirements and creating deliveries.
- Balance of conflicting project constraints that include, but are not limited to: Scope, Quality, Timeline, Budget, Resources, and Risks.

“Project management” is a broad field that can be applied to any development scenario. However, building software presents some unique challenges.

PM for information systems (IS) has evolved in the past decade and deviated from the original methodology of PM more orientated to industrial engineering (Cervone, H. 2011).

According to Cadle J. and Yeates D. (2008), the general principles of managing any project are essentially the same but are some differences in the dynamics of each type that the project manager needs to keep in mind.

In the book from 2008, Project Management for Information Systems the authors, Cadle J. and Yeates D., advocates although the general principles of PM are broadly common to all IS projects there are nonetheless ways in which the different types of IS project do diverge from one another. They group IS projects in software development, package implementation, system improvement, business consulting and analysis designation, systems migration, infrastructure implementation, outsourcing (and supply), disaster recovery and smaller IS projects, and they question the need to apply the principles and management techniques necessary for each one. For these authors the right approach is the one who ensures success.

Furthermore, Cervone H. (2011) added that the usage of traditional PM for software development has clear disadvantages such as the extensive planning phase and late development stage when requirements could have already changed. For this author, agile project management is one of the most effective means of improving the development cycle within projects. By focusing on eliminating unnecessary bureaucracy, process and practice in PM, this methodology allows for an emphasis on really productive work.

Spundak, M. (2014) states that to fully answer the question of how the methodology can be designed for a specific project, the challenge is to define which characteristics of the project are important for this decision. This is because, different methodologies, like traditional or agile approaches for example, have their advantages and disadvantages. In addition, the challenge is to define a PM methodology that can be based on different PM approaches and highly customisable for each project in a specific organisational context. For this author, the case of the custom software development project shows that it is necessary to combine different approaches.

According the work of Keil, M., Lee H., and Deng T. (2012), IT projects continue to suffer a high failure rate. They reported a "marked reduction in project success rates", with 44% of all IT projects being delivered late, over budget or not meeting requirements, and an additional 24% being cancelled before completion or delivered, but never used.

So, managing IT projects to achieve a successful outcome requires overcoming a multitude of different risks that are encountered along the way

Teams spends countless hours and sleepless nights thinking about the best way to design software. Their efforts directly impact the parameters for measuring project success:

- On time delivery
- Completed within budget
- Delivery with all features completed.

Paulk, M., Weber, C. and Chrissis, M. (1991) report that the success of a software project is closely related to the ability to achieve the desired cost, time and desired level of product quality and therefore a software project is considered successful if it delivers the product with an agreed quality level on time without exceeding the budgeted cost. The project success could also be measure by the number of post-release bugs, the support needed after release, the customer rating, the revenue generated and the client's satisfaction.

For the same authors, a successful software organisation has the ability to manage software development and maintenance processes. For these authors Key characteristics, that are directly related to management are: precise process communication, activities are carried out as planned, work should be based on consistent processes, but updated and improved where necessary, roles and responsibilities are clearly defined, and quality of the process and product is monitored. In general, a disciplined process is consistently followed because all participants understand the value of doing so, and the necessary infrastructure exists to support the process.

2.1.1. Related Activities

Having a good and well adapted management is crucial to improve profit, reduce waste (everything thing that the client is not willing to pay for) and optimise all processes (Bhuiyan and Baghel, 2005).

For Cadle J. and Yeates D. (2008), there are four main subjects intrinsically related to specific activities that management must consider and study:

a. THE BUSINESS CONTEXT OF PROJECTS (TYPES OF PROJECTS, THE RELATIONSHIP OF PROJECTS TO BUSINESS STRATEGY AND LIFE CYCLES)

Cadle J. and Yeates D. (2008) ask if it is necessary to go through all the rigours of estimating, planning, monitoring, and reporting in small IS projects. For authors is fair to adjust the PM approach to the size of the project. What they suggest about smaller projects is the need some method of baselining the project and other typical PM deliverables, such as plans and reports, should be abbreviated so that they are proportional to the size of the project being undertaken.

For Shenhar (2001) several organisations treat all projects the same way, regardless of the type of project or business strategy chosen. Ideally, when an organisation's business strategy is translated into project-level objectives such as speed to market, superior product quality, and other factors. By understanding the challenges involved in aligning PM and business strategy, professionals can effectively manage their projects in today's competitive environment.

Every project is different, of course, but the differences when managing IS projects are especially so, due to its distinct life-cycle processes. For Cadle J. and Yeates D. (2008) selecting an appropriate system DLC model is important for the project. It is probably true to say that it is best to go into the project with a clear idea of the general form that development will take. Furthermore, for these authors there are only two basic system DLC models: the waterfall model and the spiral model. All other well-used models tend to be variants or refinements of these two.

Therefore, for the same authors, there are several types of IS projects that can be found and, although the general principles of managing any project are essentially the same, there are some differences in the dynamics of each type that the project manager needs to maintain. mind. So, some projects call for strong stakeholder planning and management while others for better interpersonal skills.

When looking at the business context, it is important to mention, according to Cadle J. and Yeates D. (2008), that the analysis of a business case is essential before the execution of any project, as it defines the scope of the project in terms and establishes the costs and benefits of its execution. Ideally, the project manager should be involved in preparing the business case or at least have his or her opinion. The business case provides a baseline against which possible changes in the scope or direction of the project can be assessed and decided. One the other hand, it is vital to the success of an IS project that there is a clear understanding of who the customer is, who will make the major decisions about the scope and direction of the project and who will ultimately accept responsibility for the project.

b. THE PROJECT EXECUTION (PLANNING AND CONTROL OF PROJECT)

Kerzner H. (2013) defines that planning is determining which needs must be solved, by whom, and when. There are four reasons for planning projects: eliminate or reduce uncertainties, improve the efficiency of the operation, obtain a better understanding of the objectives, and Provide a basis for job control.

Cadle J. and Yeates D. (2008) ask: So why do we need to plan? There are some compelling reasons: Developing an information system is a complex undertaking, generally involving the synthesis of various elements – hardware, software, data capture, user training and so on. Making something of this complexity is likely to succeed only if it is planned carefully in

advance. The people involved in a project need to know exactly what their role is, what they are expected to produce and when it is wanted. The project plan communicates this information to all concerned. Customers want to be confident that the developers know what they are about. The plan is a tangible demonstration that thought has gone into the work and that the developers have a clear idea of where they are going. And unless there is a plan, how can the project manager know whether the project is on schedule, ahead or behind, and whether corrective action is needed?

Lakein A. (1973) says "Planning is bringing the future into the present so that you can do something about it now". Planning involves organising tasks and schedules to align SD activities with requirements, in order to make timely progress through the phases of the development process. Creating tasks and schedules requires identifying who should do the work and estimating how long that work will take.

According to Cadle J. and Yeates D. (2008) planning is essential to the successful execution of an IS project, because involves thinking hard about the project, what it is to achieve and how the team will go about it. The starting point for a good plan is a proper understanding of the requirements. The work to be done is analysed using either a work breakdown structure or a product breakdown structure. In either case, the product is a set of activities that will need to be performed to complete the project. With the activities identified, the next step is to understand the dependencies between them.

In the opinion of the same authors, the starting point for a good project plan is a proper understanding of the requirement: what is it exactly that the project is supposed to achieve? If an IS project is to be successful, then all concerned must know in detail what they are trying to do. Unfortunately, in too many instances, this is far from being the case.

Pereira, C. (2008) summarises the idea of several authors when says that for planning and control the most common techniques and tools in PM (referenced by reference guides) are: Work Breakdown Structure, network analysis, Gantt charts, Critical Path Method (CPM), Program Evaluation and Review Technique (PERT) and Earned Value Management (EVM.

Using the ideas of Cadle J. and Yeates D. (2008), we can summarise these points above in: the starting point for a good plan is an adequate understanding of the requirements. The work to be performed is analyzed using a work breakdown structure or a product breakdown structure. in both methods, the product is a set of activities or tasks that will need to be performed to complete the project. The need then arises to understand the dependencies between the identified activities. Using a network diagram shows these dependencies and allows you to identify the critical path in the project. Bar charts, like Gantt, provide a visual way to show the sequence of activities in a project. The project manager needs to understand the tolerances - of time, cost and quality - within which the project must operate. Project planning software can take much of the work out of the planning process, produce high-quality results and facilitate hypothetical analysis of various scenarios.

Others important subjects in many projects is the estimating, monitoring and controlling activity. What often happens is that a project is originally underrated, and over time, exceeds its budgets and dates. Although there are some specific problems that arise in IS projects, most of the estimation difficulties can be attributed to the lack of an adequate approach to the process, as practised in more mature engineering disciplines.

For Pereira, C (2008) another factor to take into account as it is a very frequent difficulty in organisations is the allocation of resources. An organisation can hardly survive through a single project. It is necessary to conduct several projects simultaneously, in order to raise funds to cover the costs, especially when the projects do not proceed as planned.

c. THE TASK RELATED TO DELIVERING (MANAGING RISKS AND STAKEHOLDERS)

About managing for Hall E. (1998) the primary source for a failed software project is the project, process, and product risk when risk management is not done.

Risks are defined by Kerzner (2013) as possible events that in case of occurrence may affect project timeline, costs, and performance.

Heldman et al. (2005) divide project risks in four categories:

- Technical, quality, or performance risks – these risks include "risks associated with unproven technology, complex technology, or changes to technology anticipated during the project. Performance risks may include unrealistic performance goals"
- PM risks – this includes risks related to "improper schedule and resource planning, poor project planning, and improper or poor PM disciplines or methodologies".
- Organisational risks – this category includes "resource conflicts due to multiple projects occurring at the same time in the organisation; scope, time, and cost objectives that are unrealistic given the organisation's resources or structure; and lack of funding for the project or diverting funds from this project to other projects"
- External risks – "the external risk category includes those things external to the project, such as new laws or regulations, labor issues, weather, changes in ownership, and foreign policy for projects performed in other countries"

d. THE SUBJECT RELATED TO MANAGING PEOPLE

For Cadle J. and Yeates D. (2008) SD requires close and constant collaboration between all the parties involved in the project. In practice, this means users and developers working very closely together all the time. Management requires managing customer expectations, how to deal with customer relations issues around changes and variations to a project, the need to manage conflict and the importance of networking. All of this becomes theoretical, however, unless you have the skills you need, so we also consider some practical interpersonal skills such as managing conflict, establishing rapport, active listening, and networking.

Furthermore, it is vital to the success of an IS project that there is a clear understanding of who the customer is, who will make the major decisions about the scope and direction of the project and who will ultimately accept responsibility for the project (Cadle J. and Yeates D., 2008).

For Institute of Electrical and Electronics Engineers (2006) Software PM Activities usually include:

- Creating a process, because provides from the beginning a structure of expected activities and work which help to guide a team through the development
- Setting standards to ensure the team's activities are performed consistently

- Managing risks
- Performing estimations enables development timelines
- Allocating resources to ensures that people, time, money, and equipment are deployed to do a required task
- Making measurements to track and assess the quality of the product and/or the process
- Improving the process, test and review the process is so important as it is top test and review the product

2.2 Process Models for Software Development

According to Pinto, P. (2008) a SDP is a set of activities, methods and practices involving transformation that people use to develop and maintain software.

Process models for SD have evolved alongside advancements in computer science. While it may be tempting to simply adopt the latest innovations in software process models, not all process models fit all projects' needs. Some process models are more complex or comprehensive than needed for a specific project.

Something that, according to Zamudio et al., (2017), is still very common today, is the use of common sense and experience rather than proven formal methodologies in SD.

To same authors, ad hoc development occurs in the absence of structured processes and happens in a development scenario where work is done reactively to inputs such as developer ideas or customer requirements. Ad hoc development can be inefficient as developers can waste time on poorly designed features. Sheetal Sharma et al. (2012) reinforces that there is no single methodology that is suitable for any project.

So, to Sheetal Sharma et al. (2012) in software DLC, there are two main considerations, one is to emphasize on process and the other is the quality of the software. The SD activities such as planning, analysis, design, coding, testing, and maintenance need to be performed according to the requirements and customer's needs.

So, what is a process? The Institute of Electrical and Electronics Engineers (IEEE) defines a process as "a sequence of steps performed for a given purpose". A secure software process can be defined as the set of activities performed to develop, maintain, and deliver a secure software solution. For Pinto, P. (2008) a SDP is a set of activities, methods and practices involving transformation that people use to develop and maintain software.

In his theses, Zamith, M. (2019) reinforces that a process is not a strict prescription to SD. The process should be an adjustable approach that should be taken by the team to choose the right activities taking into consideration the resources they have. He refers the final purpose is the on-time delivery of a software with the quality and cost that stakeholders expect. In his work, he resumes in planning, specification, modeling, development, validation, and maintenance the set of activities common in all development models.

Pinto, P. (2008) states that to be able to deliver successful software projects, a focus on the process is essential. This focus would help in predicting the results, trends, and characteristics of the project. It also reinforces that when several people work cooperatively on a common project, they need to coordinate their work in some way. However, the author argues that the way of coordinating the work may differ according to the size of the project or tasks. For small

projects or simple tasks, this can be done informally, but with a large number of people or with more sophisticated activities, more formal arrangements are needed.

According to Sheetal Sharma et al. (2012) there are several SD models like spiral, waterfall, RAD which has their own advantages.

Models for the DLC provide a guide of crucial importance for the success of any project, however, it is always debatable which model will be indicated below. In fact, perhaps the only imprudent choice is that of absolutely standardising and imposing the same development model for all situations related to SD

Taking this last sentence in consideration in the following topics, an overview of the major known SD models and how they can be used in a project context will be done. The description of each model will try to emphasize his characteristics, advantages, disadvantages, and applications.

2.2.1. Linear Models

Linear process models follow a series of phases, one by one, with no return to prior phases. The product is essentially specified, design, developed and release in that order, with no opportunity to revisit earlier phases. This kind of processes are best suited to projects where very little feedback or refinement is expected.

According to Wong, K (2019) in his notes from Software Product Management Specialisation the three principal linear models are: the waterfall model, the V-model, and the Sawtooth model. For this author these models can still be useful for certain situations, like when teams are developing simple, well-defined products.

a. WATERFALL

For Pinto, P (2008) this model is considered the traditional approach to SD.

Cadle J. and Yeates D. (2008) explain that in this model, the system development is broken down into a number of sequential stages represented by boxes, like it is show in figure 1, with each stage being completed before work starts on the following one. The outputs from one stage are used as an input to the next.

For Zamith, M. (2019) corroborates the literature by defending the increased importance of the activity of gathering and specifying requirements with the customer in the beginning, where the functionalities are identified and the restrictions to be provided by the final solution are evaluated.

One important consideration for Pinto, P. (2008) is that in this model the fixes or modifications only occur in the maintenance phase. This gives big importance to the requirements stage. For this author, this model can be successful when

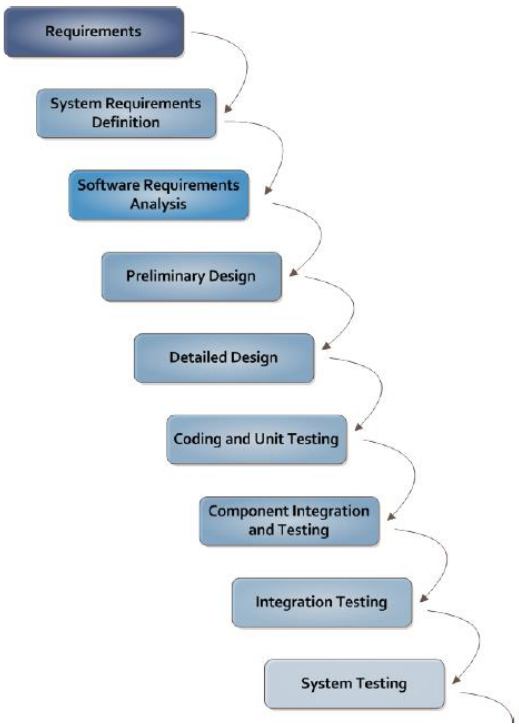


Figure 1 - Waterfall Model (Cadle, J. and Yeates D., 2008)

requirements are well understood and documented since the beginning of the project. However, there are some disadvantages like the feedback only occurs between phases, corrections often wait for the last stage, progress are not observable until the later stages and no working software is produced until late during the development process, between others issues.

Zamith, M. (2019) adds that despite being intuitive and easy to understand, with clearly defined documentation and milestones, it is not agile enough to react to change in complex projects, and can generate bottlenecks when elements of the team are dependent on others.

Wong, K. (2019) resumes that this linear process model in which each phase produces an approved work product that is used in the next phase, faces one major flaw: it is not very adaptable to change.

b. V-MODEL

According to Zamith, M. (2019) one of the main adaptations of the waterfall model is the called V-Model in which the main difference is the planning of different test activities in parallel with the development stages. The weight given by this model to the different validation steps has the consequence of producing detailed evidence of the quality of the process.

According to Wong , K. (2019) this model has the same advantages and disadvantages as the waterfall model, however allowing the development team to verify the product at multiple levels in each phases is an important improvement over the waterfall model. In both models the clients only see finished product at everything is complete.

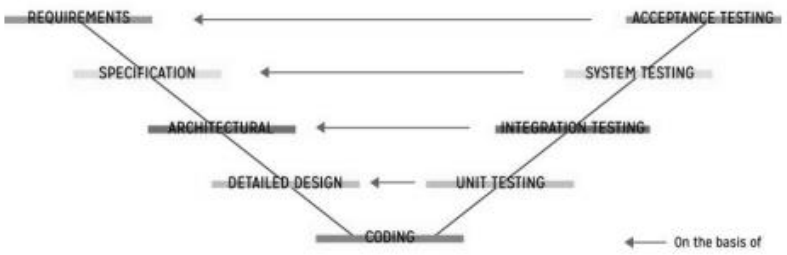


Figure 2 - The V-Model (Wong, K., 2019)

c. SAWTOOTH MODEL

This is also a linear process model, but unlike the previous ones, the client is involved to review intermediate prototypes of the product during the process.

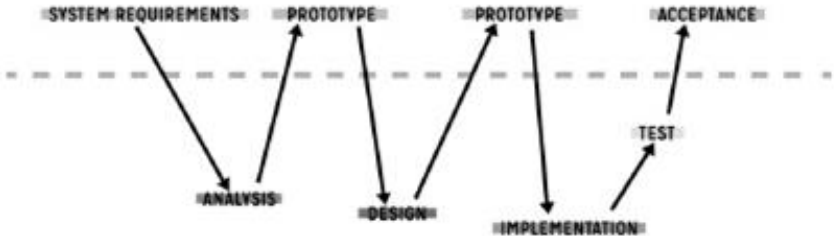


Figure 3 - The Sawtooth Model (Wong, K., 2019)

In the upper section of the diagram showed in figure 3, we can see all the interactions of the client and their respective involvement. So, this is the great advantage of this model because it allows increasing the probability of the product to satisfy the customer's needs as referred by Wong, K. (2019).

2.2.2. Iterative and Incremental Models

For Zamith, M. (2019) linear models do not respond to the current characteristics of modern software, as it is often characterized by continuous changes, short schedules, and an unavoidable need to satisfy the customer or the user. Incremental development, on the other hand, involves producing and delivering software in increments, rather than in a single installation. Each iteration of the process produces a new increment.

According to Pinto, P. (2008) the incremental model is an intuitive approach to the waterfall model and consists essentially in a series of waterfall cycles. The requirements are known at the beginning of the project and divided by groups to develop incrementally and to facilitate management, the cycles are divided into smaller iterations.

The same author explains that until all requirements are met, the SDP is repeated, with each version adding more functionality. Each development cycle acts as the maintenance phase for the previous version of the software. Although new requirements are discovered, during the development of a given cycle, they can be implemented in later cycles.

Cadle J. and Yeates D. (2008), reinforces that this incremental model is not appropriate where the scope of the project is poorly defined or undecided or where there is lack of clarity of some of the requirements.

Pinto, P. (2008) in his research refers that despite of the usual improvement of the final requirements more in line with real expectations of the clients, is also usual an increase in both cost and schedule because there are more activities and changes. The literature seems to demonstrate the increased risk in several areas like temptation to defer requirements definition or vulnerable to delays in funding approvals.

For Zamith, M. (2019) despite being more flexible than the linear model, since the reaction to the change implies fewer costs, defects are identified more prematurely in the process, and the risk is lower because it assumes constant feedback from the customer, this model has some disadvantages, including problems in the definition of the system architecture associated with the degree of uncertainty of the requirements and the difficulties in estimating a project completion date.

For Wong, K. (2019) this type of models differ from de linear process models in that they are designed for repeating stages of the process.

According to the same author, the advantage of iterative processes is the ability to loop and revisit previous phases as part of the process. Each “loop back” is a iteration that allows for client feedback.

Iterative and incremental models are also closely linked to evolutionary concepts, that implies the iterative delivery of an increasingly complete product. Among the evolutionary options for SD, two stand out: Prototyping and Spiral Model.

a) PROTOTYPING

For Zamith, M. (2019) a prototype is an example, a model or a version of the product to be developed, which aims to test and validate a concept. The prototype goes through specification and testing activities and only implements confirmed requirements. However, it is used as a strategy for raising additional requirements. Iterations occur as the prototype evolves to satisfy the needs of stakeholders, in addition to clarifying for everyone the global understanding of the solution.

To Wong, K. (2019) there are five types of prototypes: illustrative, exploratory, throwaway, incremental, and evolutionary. The core idea behind all prototyping is to gain feedback on versions of the product, to make informed decisions about the product's future.

b) SPIRAL MODEL

According to Pinto, P. (2008) the Spiral model, that we show in figure 4, was developed with the aim of reduce risk in software life cycle. Risk management is a key element of this model because in each round of the spiral the problems with the highest risk are identified and a set of solutions is developed. So, this model has been used extensively in the commercial world because of its performance in a market-driven environment.

Wong, K. (2019) explain that the Spiral model has four phases, which have associated goals: determine objectives, identify and resolve risks, develop and test, and plan the next iteration. Early iterations lead to product ideas and concepts, while later iterations lead to working software prototypes.

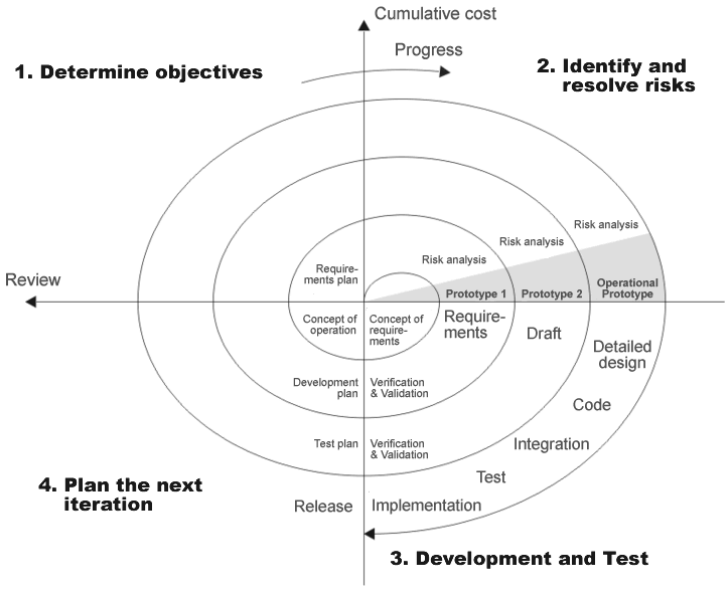


Figure 4 - The Spiral Model (Pinto, P., 2008)

According to these ideas for Pinto, P. (2008) better risk management, better definition of requirements and the system more responsive to user needs are the advantages related to the application of this model. However, for the same author this model is hard to manage and usually increases development costs and schedule.

Wong, K. (2019), on the other hand, agrees and adds that in this model, estimating deadlines becomes a huge challenge and that risk management may consume too much time and resources to do it correctly.

c) UNIFIED PROCESS MODEL

For Wong, K (2002), The Unified Process model is an iterative model of SD. Its basic structure has sequential phases in a repetitive cycle. In most phases of this model, the work takes place in small iterations until a specific point in the project is reached, then the phase is completed.

For Arlow, J. and Neustadt, I. (2002) there are four phases within the Unified Process: inception, elaboration, construction, and transition. Within each phase, there are five 'workflows': requirements, analysis, design, implementation (building the software) and test.

Wong, K. (2002) considers this model as a parallel process and in spite of using a similar style of iteration as previous models addressed, this model allows for more concurrency of activities and tasks. In other words, the general structure of the Unified Process is iterative, however, the model allows for tasks done in one phase to also occur in another. For example, a requirements task or activity can happen throughout the phases instead of in just one.

The same author reinforces with the idea of activities like requirements, design, and implementation can happen at the same time, and they all drive the product to a "release" stage. However, once the product is released, the process can begin another cycle to refine and improve the product. So, this Unified Process is well suited to large projects.

2.2.3. Agile Methodologies

Zamith, M. (2019) explains to us that until the beginning of the 90s, the general point of view in the software community was that the best approach to achieve quality consisted of careful planning, quality assurance and, globally , in a controlled and rigorous engineering process. This vision came from experience in the development of critical systems. However, when this approach started to be applied to smaller business systems, the overhead involved became so significant that it completely dominated the development process. Most resources were invested in fulfilling the process, to the detriment of construction and testing activities. Dissatisfaction with plan-driven approaches has led recognised software developers, writers and consultants to sign the *Manifesto for Agile Software Development*.

According to Sheetal Sharma et al. (2012) to meet rapidly changing requirements and the short product cycles required by companies today, software developers have developed agile SD methodologies using iterative development, prototyping, models, and minimum documentation requirements.

Wong, K. (2019) details that the main values and principles that define Agile philosophy are: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

As Pinto, P. (2008) refers all agile models are based on an iterative workflow and the incremental delivery of work software in short iterations over time. An iteration is essentially a small version of the software.

To the same author the agile software development is a conceptual framework that promotes development iterations throughout the life cycle of a project. Agile processes were described as a style of SD characterized by an emphasis on people, communication, working software, and responding to change.

Zamith, M. (2019) also states that the basic aspect that makes agile methodologies so relevant is their ease of reaction to change. However, the philosophy of agility is more than an efficient response to change. Facilitated communication between teams and other elements related to the project is encouraged. It emphasizes the fast delivery of functional solutions. Communication with the customer is promoted because it is assumed to be an integral member of the development team. It is recognised that planning has limits and must be flexible.

According to Zamith, M. (2019) despite characteristics such as quick reaction to change, involvement between stakeholders, and the constant encouragement of cooperation between team members, these models lack a high level of skills from the participants and eventually the need to involve resources for customer training. Other topic to take into account is the prioritisation of items can also be problematic if the set of stakeholders is large and dispersed in nature. Different stakeholders often assign different priorities to change.

For Cadle J. and Yeates D. (2008), some circumstances justify an agile approach. The pressure to develop quickly due to an increase in the pace of changes in business, the redesign of traditional organizational structures, the increase in user expectations in the information system due to the constant technological evolution, and the fact that for the most diverse reasons customers are unable to make a detailed specification of their requirements at the beginning of the project.

Zamith, M. (2019) conclude that the agile movement is, in his essence, an effort to overcome the weaknesses in the conventional software engineering process. However, it is essential to realize that recognizing the challenges of the realities of modern development should not result in the abandonment of the basic principles described in traditional models.

In his study Pinto, P. (2008) identifies several agile models however, the ones that seem to be more popular nowadays are Scrum, Extreme Programming (XP), Feature Driven Development, Dynamic Systems Development Method, Adaptive Software Development, or Lean, between others.

Here three of the most used ones stand out: Extreme Programming, Scrum, and Lean. And in the next pages we will also analyze a management practice called Kanban that works well with these methodologies.

A. SCRUM

Scrum is the most popular methodology for SD according to agile ideology. This methodology is both an iterative and an incremental process. It is iterative in the sense that plan for the work of one iteration, to be improved upon in subsequent iterations. However, it is also incremental because completed work is delivered throughout the project.

According to Wong, K. (2019) Scrum is an Agile methodology that consists of management practices with frequent evaluations of the project, with the aim of increasing predictability and mitigating risks. These are light practices that have relatively little overhead. Although simple to understand, they are very difficult to master in full.

The same author refers that Scrum is based on three pillars: transparency, inspection, and adaptation. With transparency, everyone can see every part of the project, from inside the team and outside the team. Scrum encourages frequent inspection of work products and progress to detect undesirable deviations from expectations. When a deviation is detected, the team must adapt and adjust to correct the problem. Important is agreeing on standards for the project such common terminology and definitions such as the meaning of "done."

For Zamith, M. (2019) the agile movement is, in his essence, an effort to overcome the weaknesses in the conventional software engineering process. However, it is essential to realize that recognizing the challenges of the realities of modern development should not result in the abandonment of the basic principles described in traditional models.

Pinto, P. (2008) explains that Scrum projects progress through a series of thirty-day iterations called "sprints". At the start of each sprint, the team determines the amount of work it can accomplish during that sprint. Work is selected from a prioritized list called the "product backlog". The work the team believes it can complete during the Sprint is moved onto a list called the "Sprint backlog". A brief daily meeting, the "daily scrum", is held to allow the team to inspect its progress and to adapt as necessary.

Wong, K. (2019) conclude that to enable the three pillars, already mentioned, Scrum outlines four required techniques or events that happen within a sprint:

- Sprint planning (to set expectations for the sprint)
- Daily scrums (to ensure work is aligned with these expectations)
- Sprint review (to gain feedback on completed work)
- Sprint retrospective (to identify what to do better in the future)

For Pinto, P. (2008) the biggest advantages of this methodology are increasing productivity, continuous improvement, capability to leverage the chaos, and it is easy to learn.

B. EXTREME PROGRAMMING

According to Pinto, P. (2008) XP is one of the more well-known agile methodologies, based in an incremental and iterative development. This methodology was created in response to problem domains whose requirements change, where customers may not have a firm idea of what the system should do.

Wong, K. (2019) defends that this methodology consists of effective development practices to achieve client satisfaction. The practices, that we can see in figure5, focus on constantly delivering software, responding to change, effective teamwork, and self-organisation.

Everyone is valued and respected for what they individually bring to the project, be that business experience, domain expertise, or development skills. Everyone is considered equal, so that the client, software product manager, and development team members are all on the same level, working together, as Wong, K. (2019) mentions in his work.

According to Cadle J. and Yeates D. (2008), this method was created to deal with common project problems that quickly change requirements, defy deadlines and work with new or unknown technologies. The literature define that it is a great method when there is no attempt to scope the entire system from the beginning. It is also suggested that XP works best on relatively small projects, or at least projects that can be done by small teams. The XP concept is for the small team of developers who work side by side with their customers to create, test and implement relatively small software compared to deliberately challenging deadlines. The software is created from user 'stories' and is tested as it is built.

For Pinto, P. (2008), XP is characterized by the twelve practices described, with positive and negative aspects. At some point these practices look like rules, yet they are more like guidelines than rules.

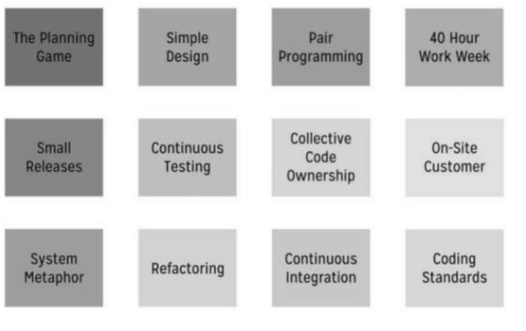


Figure 5 - The Principles of Extreme Programming (Wong, K., 2019)

C. LEAN SOFTWARE DEVELOPMENT

One Agile methodology that has gained increasing attention is Lean Software Development. as Wong, K. (2019) refer in his work. Its origin is inspired by the manufacturing industry where Lean production grew out of the “Toyota Production System.” At Toyota context, their approach was used effectively to reduce waste in the production process and increase the quality of their vehicles.

For the same author the Lean is based on seven principles: eliminate waste, amplify learning, decide as late as possible, deliver at fast as possible, empower the team, build quality in, and see the whole. These principles can be effective for both small and large projects.

D. KANBAN

According to Wong, K. (2019) Kanban involves a technique to organize and track project progress visually, which is widely used even outside Agile or Lean software development. Kanban is a Japanese word, which loosely translated means board or signboard. So, the Kanban technique uses a board, with a set of columns labelled by the stages of completion.

3 Case Study Description

The first objective of this chapter is to understand the company and the context in which it operates. Realising how it is strategically organised and positioned considering its mission and values.

The second section of the chapter provides an overview of the information technology processes and what is the impact of these processes on the precession of the quality, cost and delivery of the services provided by the TBFiles' IT department.

Then, to select the process to be analysed in the next phases, the processes were prioritised taking into account the impact and their weight on billing.

In a third phase and after collecting information, goals were defined to be achieved for customised software solutions.

3.1 Company Business Model

TBFiles is a company certified in the provision of integrated document management solutions and administrative back office outsourcing.

TBFiles' mission is to provide innovative outsourcing services looking to the dematerialization and automation of administrative processes, with maximum quality, professionalism and efficiency, completing the entire information life cycle.

Satisfaction and trust are values that TBFiles wishes to share with its employees and customers, through a professional and capable attitude.

Thus, TBFiles is guided by the following objectives:

- Support for technological innovation so that it is possible to unlock processes in organizations in an evolutionary way.
- Reduction of costs related to the relief of non-core tasks.
- Increase in the productivity of your customers' employees.

In the national market since 2003, Mozambique since 2006 and Angola since 2012, TBFiles main concern is the safe and confidential management of data and documents, keeping them protected and with maximum accessibility.

Regardless of maintaining an ambitious organic growth since its creation, in 2006 TBFiles acquired Triarquivo. This strategy was replicated in 2015 with the acquisition of 100% of KeepItSafe's capital, reinforcing its presence in the national market.

Strategically established in Portugal, Porto and Lisbon, TBFiles is aware of the logistical needs of large companies, responding nationally and internationally. With logistical capacity to set up operations 24/7 and collect documents anywhere in Europe, providing data in Portuguese, English, Spanish (Catalan and Spanish) and French, through the most secure tools via the web.

To ensure this level of services, the company is currently organized into a functional organizational structure with 5 hierarchical levels. Consider the following scheme:

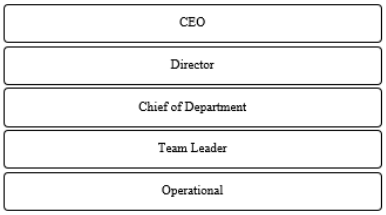


Figure 6 – TBFiles' Organizational Structure

Guidance is clearly the added value, maintaining the "free licensing" strategy, offering a service that ensures that these web platforms are adequately fed with service levels that are much more demanding than those normally established internally. TBFiles guarantees the availability of data and all the automation that its customers request.

In summary, TBFiles has evolved over the years from a document custody company with all the inherent logistical skills to position itself in the value chain, ensuring all the activities inherent in document management. Activities ranging from the collection of physical documents to a structured data delivery service to its customers supported by software developed according to the needs of its customers.

In order to analyse the TBFiles' product portfolio, we use the Boston Consulting Group (BCG) matrix described by Buttle, F. and Maklan, S. (2015) designed so that companies can develop strategies that increase their profit and cash flow. This analysis considers two criteria, relative market share and market growth rate. Figure 7 details the BCG matrix of TBFiles.

Table 1 - BCG Matrix for TBFiles

Market grow		
High	**STAR** Documentary workflow Automatic data extraction Process dematerialization BPO	**QUESTION MARK** Customized Software Solutions
Low	**CASH COW** Documentary custody Digital files Document indexing Scan on demand	**DOG** Stock management (picking) Non-standard formats custody Physical files
	10x — Relative market share — 0,1x	

3.2 "As-Is" Characterisation

After understanding the company and its products, the purpose of this step was to do a deeper characterization of the initial situation of the IT department.

3.2.1. IT Services

First it is important to comprehend how TBFiles provides its customers a wide range of services based on information technology systems, namely:

- Capacity to Develop Web Portals based on Big Data
- B2B Solutions: (Credit, Accounts Payable with workflow, etc.).
- B2C Solutions: (Rent-a-car; Student applications, etc.)
- Internal Collaboratives: Customer Relationship Management (CRM), Digital Archive, Human Resources, Quality, Purchasing, etc.
- Robotics: Possibility of creating ROBOTS to automate processes based on predefined rules.
- Outsourcing of Administrative Backoffice
- Ability to outsource critical information extraction procedures from documents / emails / databases to feed business processes or administrative automatisms.
- Mail Management / Data Entry / etc

3.2.2. IT Organisation, Resources, and Tools

The information systems team is divided into three teams, one focused on internal projects, the other on external projects (customers) and support. Managed by an IT Manager under the supervision of the operations department director.

There is a vision at TBFiles of a similar approach between internal and external projects, including in relation to the customer view. However, we realise that the specifications are more likely to be changed in projects with internal customers.

Although recent, there is still the figure of the project manager, but its scope of action is mixed between project, team, and resource management.

Because it does not have a license cost, the development team uses it to develop open-source software and the internal collaboratives are internal develop. This offers the opportunity for greater investment in services and training, ensuring a greater and better return on IT investments. Simultaneously the tools used are:

- Code writing - Visual Studio / PHPStorm / NotePad ++
- GIT- SmartGit / GitLab
- Database - Navicat / WorkBench
- APIs- Insomnia / PostMan
- WebServers - XAMPP / Docker
- WebBrowsers- Chrome / Firefox

3.2.3. IT Fundamental Processes

TBFiles divides these processes into six fundamental processes that characterize the global services of the IT department: parameterisation of physical files, parameterisation of digital files, developing customised solutions (projects), budgeting, developing robots and problem solving (support and maintenance).

Although there is an idea within the company that the main problems would be in customised solutions, we chose to briefly know each of these processes using a process modeling tool called SIPOC.

SIPOC is s visual tool used to represent processes at a high-level, and it is usually applied by a team to identify all relevant elements of a process improvement project. These diagrams present a high-level description of processes and do not contain too much detail. The main objective of SIPOC is to ensure an understanding of the scope of the process, realizing the beginning and the end (Hamza, S., 2008).

Table 2 - SIPOC of TBFiles' IT processes

Suppliers	*Inputs*	*Processes*	*Outputs*	*Clients*
Operations	Order Data and Files	**Parameterise Digital Files Portals** Creation of tables; Robot parameterisation; Test; Portal Parameterisation	Customised portal Files available on the portal Integration with the customer's IS.	Digital Archive Clients
Operations	Order Data and Files	**Parameterise Physical Files Portals** Portal parameterisation (client image) Test	Customised portal	Physical Archive Clients
Operations Commercial	Data Order email Specifications	**Develop customised solutions (Project)** Planning Back End Front End Integrations and Tests Go Live	Portal Modules API Data base Customised Software	Clients Operations
Commercial	Order Specifications	**Budgeting** Requirements Analysis Budgeting Functional analysis	Functional Dossier Budget	Commercial
Operations	Order Data and Files	**Develop Robot** Operative Analysis Development Test Monitoring	RPA Script	Operations
Internal Departments Clients	Order	**Troubleshooting (Support)** Problem analysis Resolution Test	Support	Internal Departments Clients

To have a clear vision and an unbiased analysis, all stakeholders in the IT department are responsible for the result of this brainstorming from IT Manager to Junior Developer. This

analysis will be important throughout our work even to decide who will be the different players in the most diverse exercises.

So, after this high-level analysis of the existing processes in the company's IT department, we try to understand the importance for invoicing. By doing this analysis, we can see, as shown in the following table, that customised IT solutions represent more income than all the other processes together.

Table 3 - Processes impact in invoicing

Processes	*%*
Parameterise Digital Files Portals	15
Parameterise Physical Files Portals	5
Develop customised solutions (Projects)	60
Budgeting	5
Develop Robots	10
Troubleshooting (Support)	5

But does this turnover translate into profit? This is one of the big questions that we want to address over the next pages of this analysis and work.

3.2.4. Prioritisation

After understanding the processes, we try to understand what impact each of these processes had on the perception of quality, cost and delivery of the service provided by the TBFiles' IT department. The results of this workshop, showed in table 4, attended by not only the IT department, but also senior management, including the CEO and high-level directors of interested areas such as operations, sales, and quality.

Table 4 - Perception of processes impact on TBFiles' IT services

	Quality (3-1)	*Cost (3-1)*	*Delivery (3-1)*
Parameterise Digital Files Portals	1	2	1
Parameterise Physical Files Portals	1	1	1
Develop customised solutions	2	3	3
Budgeting	3	3	3
Develop Robots	2	1	2
Support	2	2	3

In this exercise the processes are scored from low impact (1) to high impact (3). After a lot of discussion in several points the team finally reached an agreement. So, regarding the parameterisation of a physical archive portal, the perception of quality, cost and compliance with service delivery deadlines is maximum. In the opposite direction, there is a perception of these same points in relation to budgeting.

But does perception reflect reality?

Before clarifying this doubt, we did a prioritisation exercise giving different weightings also considering the impact of these processes on invoicing, referred above, and their alignment with the company's strategy. So, priorities were defined and accepted, which we show in the figure 7.

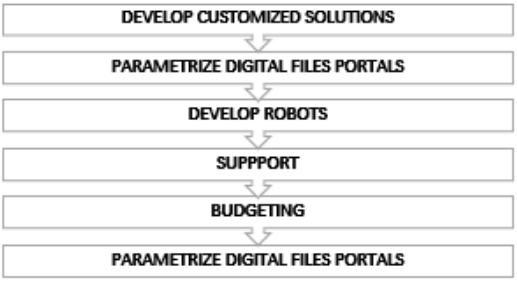


Figure 7 - TBFiles' processes improvement priorities

3.2.5. Software customised solutions at TBFiles

According to Johnston, R, Clark G. and Shulver, M. (2012) are several tools that are very important for designing new processes and to help assess and improve the design of existing ones. These include process mapping, lean thinking, kaizen, six sigma, and value stream mapping.

The same authors refer that the essence of process mapping is to capture all activities and their relationships. This is usually done by a team of people who understand the various aspects of the process. It may not be surprising that this activity is time-consuming, but it can produce some significant results - the first of which is usually the emergence of a shared vision and an understanding of a process and the role of each involved.

As previously referred, informal workshops with key stakeholders and observation of team performing their daily tasks and processes were useful to develop a generic picture of this IT process to develop customised software solutions, represented in Appendix A. The following processes and their relationships were identified in the figure 8.

Figure 8 - Phases for developing customized software solutions at TBFiles

For Sharp, A. and McDermott, P. (2009) the process workflow design is the work plan for responding to a triggering event. It shows the sequence of steps, decisions, and handoffs carried out by the process's actors between the initial event and the result. Having a workflow model can certainly uncover problems in the design of the workflow, but just as or even more important, it allows you to assess the other enablers in an organized fashion, step by step and actor by an actor.

In the company under study, this process is triggered by the visit of the sale account to client's company. After receiving the information about the client needs, the sales account does the process diagram and requirement specification. After validating this information with the client, a budget request is sent to IT department to analyses and to estimate the costs of the solution. This time is usually iterative and incremental between the two departments. In several situations, several contacts are necessary until the IT department understands the request and, simultaneously, the commercial department converts the budget into a proposal for the client. Then, the sales account makes the proposal to be analysed and discussed with the client. Every activity related to documentation and planning happen after the customer's decision to move forward. However, it is reported by the team that several times the documentation, planning or even development phase does not start immediately after the acceptance. Another situation raised by the team is the start of development without having the documentation and planning fully defined.

After the development is completed, tested, and the deploy is done the responsible sales account is informed and at this stage inform the client that can test and give feedback before entering in pilot mode. There are two important considerations about these moments, one is the directly pass from the development staging to the product staging and, the other is that several times there is no demonstration of the solution developed to sales account.

This process finishes after the pilot with the moment pre-defined to go live with a kick-off meeting.

Regarding this process, it is also important to mention that, when it comes to communications between departments, they occur informally by phone or email. Regarding the necessary documentation, it is carried out differently by each of the stakeholders in the different departments. We can conclude that in relation to the process itself, the procedures are mainly ad hoc, which made this mapping a big challenge because the different activities and tasks are carried out in a specific way by the different stakeholders. That is, the process itself is not standard, nor is their normalization.

3.2.6. Talking with data

In order to support our project and to answer the question raised in 3.2.4. in relation to the difference between the perception and the reality of the numbers, we decided to collect information on the field. In fact, *Talking with data* is one of the *Kaizen* principles addressed in the book by Coimbra, E. (2016), which is based on the conviction that in order to discuss and analyze any problem, it is necessary to have data collected on the *gemba (the actual place, the place where value is created).*

To gather the information that we think was important was an enormous challenge, since this when it exists, is dispersed and disorganised. There is no tool with all this information. To do this exercise, we needed to analyse existing communications in the CRM, proposal dates, the various versions of existing proposals, and even signed contracts.

As previously mentioned by Agarwal N. and Rathod U. (2005) the performance of a software project is evaluated in terms of its ability to achieve the desired cost, the time and the desired level of quality of the products and, therefore, a software project is considered successful if deliver the product efficiently and with a pre-agreed quality level within the given deadline and cost.

It was important to limit the scope of our analysis, therefore, we collected information about the 10 representative projects for customized software solutions developed by the company between January 2019 and March 2020. Projects with more than 10 days and up to 90 days in the budget were defined as representative characteristics of the case under study. Thus, below, analyses of time, quality and cost of these projects will be presented.

TIME

Regarding delivery-related issues, our approach was based on temporal issues, however, it is important to distinguish some definitions on this subject. According to Guedes, A. (2019) the most conventional definition for lead time in supplay chain management is the time between the moment of the customer's order and the arrival of the product to him. For the same author, the process time is the average time that a unit is worked on.

In summary, considering the working days, the lead time of the project and his different phases are systematized in the table 5.

Table 5 - Lead Time in days per project

LEAD TIME / *PROJECTS*	*REQUIREMENTS SPECIFICATION*	*BUDGETING*	*ACCEPTANCE*	*IMPLEMENTATION*	*PROJECT*
1	4	31	9	233	273
2	12	2	68	113	183
3	3	62	21	47	130
4	41	7	21	204	232
5	5	12	16	49	77
6	1	14	16	77	107
7	1	25	27	155	207
8	3	30	0	35	65
9	1	11	17	79	107
10	3	14	39	63	116
Average	**7,4**	**20,8**	**23,4**	**105,5**	**150**

We can observe that more than 7 days elapse between the moment when the contact with the customer takes place and the moment when the sales account asks the IT department for a budget, with the requirements and diagrams already validated with the customer. However, it seems important to mention that in one of the cases analysed, there was a delay of 41 days.

Apparently, the budgeting phase would be TBFiles' sole responsibility, in a process between the sales account and IT, however, we realised that there are several situations that required new contacts with the sales account or even with the customer. As a result of these situations, despite observing a single case in which the budget was made and delivered to the sales account by IT in 2 days the average approaches 21 days with several versions of the proposal.

The acceptance of the proposal starts in the moment when the sales account sends the final proposal to the client and his award response. We can observe an average of more than 23 days. In the situation where the proposal was accepted on the same day, we were able to observe the existence of different versions and adjustments of the budget.

We had a need given the lack of data to group the phases of documentation, planning, development, tests, and pilot in a phase that we call implementation. All these phases are the responsibility of the TBFiles' IT department. This process kicks off with the communication

by the sales account of the acceptance of the proposal and ends with the deploy of the developed solution. It is possible to observe on average almost 106 days for these processes.

It is also perceived an average of 150 working days required since the customer requests the visit of the commercial until the solution is in a production environment and available for full use.

Table 6 - Weight of each phase on Lead Time

PROJECTS	*WEIGHT BUDGETING*	*WEIGHT ACCEPTANCE*	*WEIGHT IMPLEMENTATION*
1	11%	3%	85%
2	1%	37%	62%
3	48%	16%	36%
4	3%	9%	88%
5	16%	21%	64%
6	13%	15%	72%
7	12%	13%	75%
8	46%	0%	54%
9	10%	16%	74%
10	12%	34%	54%
Average	**17%**	**16%**	**66%**

Observing the table 6, it is possible to recognise the weight each phase has on the lead time of the different projects. Only in one project did the implementation show a lower weight compared to the other phases. In a global view, implementation shows a weight of 66%, while acceptance and budgeting present very similar percentages, 16% and 17%, respectively.

To measure efficiency related to delivery, another metric that we believe is pertinent to analyse is the on-time delivery. According to the literature, on time delivery is a measure of process efficiency which measures the amount of finish goods or services delivered to customers on time and in full.

Looking at the project data from our study, shown in Appendix B, we realised that 80% of the projects did not finish in the estimated time. In fact, analysing the figure 9, of these 10 projects, 50% finished more than 61 days after the deadline, 30% more than 16 days after and only 20% were delivered on time or the delay was considered residual.

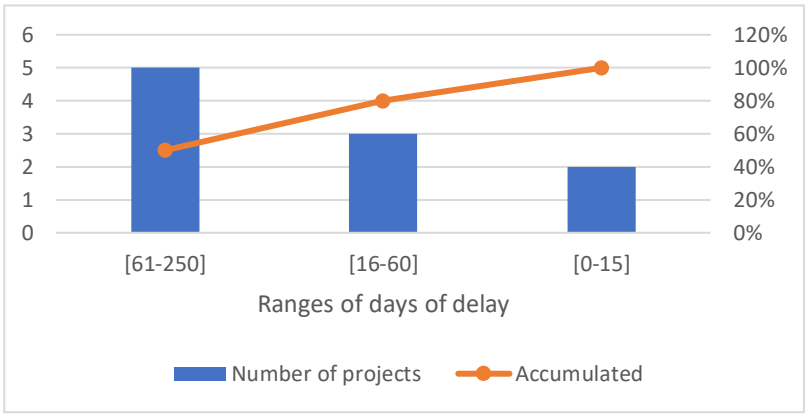


Figure 9 - Days of delivery delay

QUALITY

First, what is quality? The international standard ISO 8402:1991 offers the following definition of quality: 'The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs'.

For PMI (2013) there are some points in relation to quality that are fundamental:

- To assess customer satisfaction, it is necessary to assess the compliance of requirements (to ensure that the project produces what it was created to produce) and the suitability for use (the product or service must meet real needs).
- The quality must be planned, designed, and created, and not inspected in PM or project deliverables.
- Should exists process improvement models such as Plan Do Check Act (PDCA), Total Quality Management (TQM), Six sigma and Lean six sigma to improve the quality of the PM as well as the quality of the product from the project.
- The costs of quality of work must be considered throughout the entire life cycle of the delivery.

So, according to the literature, all projects must have a quality management plan. Project teams must follow the quality management plan and have data to prove compliance with it. However, once again we struggle with the lack of documentation in the case of TBFiles.

Taking these last paragraphs into consideration we realise that there is no quality plan, improvements models or data that we can use to evaluate the quality costs. However, it is possible to see in the table 7 that in 70% of the projects there was a need after the delivery of new developments, supports or improvements so that the project actually met the expectations and needs of the customer.

Table 7 - Percentage of projects that needed rework

Rework	*Number of Projects*	*%*
No	3	30
Yes	7	70

In our analysis of issues related to quality, more specifically customer satisfaction, we also tried to understand whether there was any document or communication about the projects in the study that could lead us to more subjective approach about the customer's perception of the work performed by TBFiles. There are only a few communications in CRM that could refer to this approach, however, there is no such standardized procedure. Sometimes it is performed and registered autonomously by the different sales account, but it is not something constant, consistent, or standard.

Based on the literature that argues that the level of service directly impacts customer satisfaction and that this is the result of dividing the number of articles delivered on time by the total number of articles, we realize as we have already observed that only 20% were delivered on time.

COST

According to Pereira C. (2008), for an efficient PM, with a minimum of three indicators:

1. The project budgets.
2. The physical progress of the project.
3. The costs spent in the period for carrying out activities (incorrect costs of work performed).

The same author also addresses a methodology that differentiates traditional control systems, EVM. Traditional budgets only compare budgeted amounts against amounts actually spent, while EVM compares the following indicators:

- Schedule planned tasks (Tasks) versus tasks actually performed.
- Planned costs vs costs actually incurred.

With this information, it is possible to analyse or the current status of the project in order to allow preventive and correct decisions to be taken.

In our case study, there is, in fact, little data on this topic.

Considering the estimated time as budget and the duration of the implementation time as the real cost, we realised that in 100% of the projects analysed the costs overlapped the budget, as we can see in the table 8.

Table 8 - Delay in days per project

Projects	*Estimate*	*Implementation Lead Time*	*Delay*
1	14	233	219
2	50	113	63
3	12	47	35
4	40	204	164
5	36	49	13
6	16	77	61
7	70	155	85
8	33	35	2
9	21	79	58
10	24	63	39

The fact that there is no standardisation of documentation limits the analysis in relation to the progress of the project or its performance at each moment.

3.3 Improvement objective

This step consisted of defining the improvement objective that was clear and that answered three questions: what is the indicator to achieve? what is the value to achieve? when should it be achieved?

Improving the efficiency and effectiveness of the IT department when it comes to custom SD is the general motto. However, and considering all the data described above, and after a meeting

to present them to the top managers of TBFiles, the three main indicators were defined. Two of them more related to time such as project lead time and on-time delivery and one related to quality, rework.

In fact, we understand that this decision is also due to the expectation that the impact that solutions for reducing the project's lead time or rework and increasing on-time deliveries, will replay in improvements in the other major areas, cost of projects.

We were then able to list some of the reasons that came out of this meeting that underlie this choice:

- related to TBFiles' mission, to provide services with the highest quality.

- related to the values of TBFiles, satisfaction and trust.

- related to return on investment.

Looking at the data we can see that the average lead time of the projects developed by TBFiles is 150 working days and in the vision, taking into account the resources available, there will have to be greater availability to develop more projects and thereby increase the return on investment in the development department.

TBFiles then intends to reduce lead time by 20%, as this reduction would allow it to move from a capacity to develop close to 9 projects per year to 11 projects per year.

As for the remaining indicators, on-time delivery and rework, the management were adamant in its desire to reverse the order of results.

For these, the 80% projects delivered after the deadline will have to be delivered to the client within the estimated time. However, they dropped the usual 7 days of tolerance to 15 days as an acceptable delay, due to the unique characteristics of SD.

In questions related to rework, they only accept that 30% of projects are in need after the delivery of changes or new developments.

In conclusion, there is an expectation that the impact of the solutions in reducing lead time by 20%, reducing rework by about 57% and increasing on time deliveries to 75% will have a positive impact on the cost of projects.

4 Problems and root cause analysis

After having a comprehensive view of the company and its operation, specifically on the processes selected for the next phases of the project, this chapter is dedicated to identifying opportunities for improvement (problems) and root cause analysis, thus responding to RQ2, and RQ3.

The data collected in the last weeks of work and the three major indicators defined by the TBFiles, addressed in the previous chapter, were presented, and explained to the team that had carried out the process mapping. So, the question arose that would serve as a guide for the next sessions:

Why are we in the current state and not in the proposed objectives?

The result of this work was 28 causes or opportunities listed by the team that were affecting the failure to achieve the expected results in each phase of the process.

After the causes have been identified, and bearing in mind the objective of this phase, to understand and analyse the root causes, we chose the diagram of Ishikawa also known cause-effect diagram. This tool consists of dividing the causes into types and the root causes arrive, successively asking "Why?".

The team's exercise in realising a problem was always to debate the problems to try to understand why this happens until the time when the root cause was almost confused with the solution.

As already mentioned, the 3 indicators are related, so it was decided to execute the Ishikawa diagram analysing only as a cause for the current situation of the lead time. With this approach we can identify opportunities for improvement and find out the real causes of the problems.

For a better understanding of the problems, we will divide into the problems that affect each phase. However, we realise that there are problems in the earlier stages of the process that stem from later stages and where the root cause is the same as in that earlier stage.

Despite the use of the Ishikawa method and this has a very particular presentation, we decided to facilitate the understanding to present and systematised the results in tables with the problems, root causes and rationale (Appendix C, D, E, F, G and H).

4.1 Budgeting phase

As we can see in the table 9, the main problems at this stage are mostly related to the low standardisation of the request and response to the budget. In fact, problems that we apparently associated with the customer or the commercial department are due to the lack of a fragile budget request.

When we look at a specific case like the problems associated with the lack of information (ID 1), these are related to the lack of adequate budget request questions.

Other issues are related to do not understand what the client needs (ID 4) or requirement specification (ID 2). These problems lead to a bad estimation of hours or even the final solutions are presented that do not meet expectations which in turn leads to rework to meet the customer's expectations

The absence or poor communication between departments that leads to not understanding all the requirements and poorly defined scope are also related to the lack of standardisation of the budget request, as we can see in the table 9.

Table 9 - Analysis of problems and root causes in the budgeting phase

Problem ID	*Problem*	*Root Cause*
P1	Missing information from the customer/sales account	Lack of standardisation of budget request
P2	Difficult interpretation in requirement specification	Lack of training
P3	Deficient IT / commercial communication	Lack of standardisation of budget request
P4	Subjective criteria or lack of sensitivity to estimate hours	Lack of standardisation of budget request
P5	Lack of detail in the budget	Lack of standardisation of budget request
P6	Difficult interpretation of the budget given	Lack of standardisation in the response to the budget
P7	Constant hourly renegotiation	Lack of standardisation of budget request
P8	Lack of visibility about the acceptance from the client	Lack of processes after customer acceptance
P9	IT does not give deadlines before acceptance	Lack of visibility of the team's capacity load
P10	Lack of technical specifications	Lack of the process of creating documentation

Other root causes identified as a result of the problems of this phase are the lack of standardisation in the response to the budget, lack of processes after customer acceptance, lack of the process of creating documentation and lack of visibility of the team's capacity load.

About this last root cause addressed in the paragraph above, comes up from the problem identified that IT does not give deadlines before acceptance (P9) and it happens because the IT manager is unaware of the team's capacity and load.

4.2 Documentation phase

The problems observed at this stage are actually related to the absence of a robust and more effective process for creating and accepting the functional dossier by the stakeholders, and simultaneously to the fact that there is no standard document to document the solution, as we can see in the table 10.

Most of these problems will impact the solution's delivery time because the absence of a dossier built and accepted by all stakeholders would increase the probability of the solution being delivered meeting the customer's quality expectations.

Table 10 - Analysis of problems and root causes in the documentation phase

Problem ID	*Problem*	*Root Cause*
P11	Little detail	Lack of standardisation of the functional dossier
P12	There is no acceptance by the client	Lack of acceptance of the functional dossier by the client
P13	Functional dossier is not carried out	Lack of the process of creating the functional dossier with stakeholders
P14	There are several versions of the same document	Lack of the process of creating the functional dossier with stakeholders
P15	Document executed simultaneously with the development	Lack of the process of creating the functional dossier with stakeholders

4.3 Planning phase

Regarding the planning phase, we realised that the problems are due to the lack of processes and simultaneously the burden of the team, either for the department itself or for the other departments of the company.

Table 11 - Analysis of problems and root causes in the planning phase

Problem ID	*Problem*	*Root Cause*
P16	There is no time set aside for unforeseen events	Lack of processes of planning
P17	Planning is not carried out in advance	Lack of visibility of the team's capacity load
P18	IT does not always participate in the roadmap	Lack of visibility of the IT department's capacity load

The first problem identified is more obvious, there is no time set aside for unforeseen events (P16) and with unforeseen circumstances will lead to more work.

About the lack of visibility is important to understand that means that it is not possible to plan and that other departments think IT are able to start immediately and that it is not exactly true.

4.4 Development phase

In this phase, we are able to more clearly realize that the problems that occur are not just due to processual problems or lack of documentation of tasks in the phase itself, but problems of the entire process in previous or later phases as we can see in table 12.

The main problem of a “bad” development is the result in the end. If in the end the solution does not work will force rework.

Table 12 - Analysis of problems and root causes in the development phase

Problem ID	*Problem*	*Root Cause*
P19	There are no priorities for tasks	Lack of standardisation of task documentation
P20	Unclear tasks	Lack of standardisation of task documentation
P21	"Support" or other situations corrupt the fulfilment of the planning	Lack of documentation all developments
P22	There is no definition of dependencies between tasks	Lack of the process of planning tasks and dependencies
P23	Tests when done are only performed at the end	Lack of test processes
P24	Code review is centralized in one person	Lack of team organisation

If, on one hand, the development of solutions that do not satisfy the customer directly impacts the lead time, this problem still gives rise to wrong priorities and inability to perceive the team's load and consequently bottlenecks. In our opinion this happens because of the lack of standardisation of task documentation.

Other important issue that is related to the long lead time is that the support that should oversee one of the developers does not solve and assigns the task to another colleague who, given the priority of the request, may lead to interrupting the planning (P21). In fact, if all developments were documented, this problem would not occur as often.

The problem related to the code review gets stuck in one person, which slows down the process (bottleneck) happens because the lack team organisation and simultaneously the managers do not have visibility of these bottlenecks.

4.5 Testing phase

The root causes of the problems identified in the testing phase, as we can see in the table 13, are mostly related to others phase of the project.

Table 13 - Analysis of problems and root causes in the testing phase

Problem ID	*Problem*	*Root Cause*
P25	Features do not match customer requirements	Lack of the process of creating the functional dossier with stakeholders
P26	Tests fail due to unforeseen operational needs	Lack of the process of creating the project with internal departments

The lack of the process of creating the functional dossier with stakeholders and the lack of the process of creating the project with internal departments will bring problems to this phase of

the project. These problems will affect the lead time because if the output of this phase may lead to the need to get back to work on the solution.

4.6 Pilot phase

The team mentioned that there are two problems at this stage, one is that the pilot period is not defined at the beginning and therefore extends beyond the delivery period. Therefore, the main problem with this is that this time to test is not estimated when sending the IT budget to the sales account, therefore, the delivery time presented to the customer is less than the reality. The other problem is that sometimes the customer does not clearly understand what has been delivered and therefore takes time to test and see improvements. Make late suggestions to interrupt IT a lot with questions. This is because the presentation meeting does not effectively empower the client.

Table 14 - Analysis of problems and root causes in the pilot phase

Problem ID	*Problem*	*Root Cause*
P27	Most times there is no closed period for pilot	Lack of standardisation of budget request
P28	Presentation meeting does not effectively empower the client	Lack of customer training processes

5 Proposal and Developed Solutions

After all, the main causes have been identified, and in order to answer the last of our research questions (RQ4), in this chapter, a set of suggestions for improvements was thought and analysed in terms of the benefits and difficulties of their implementation.

For each problem, that has its root cause, some suggested solutions are presented with the aim of improving the SD and PM daily activities.

In addition, given the limited project period and the available internal resources, not all proposed solutions were selected to be designed. In this sense, before the final design of the solutions, we analysed the possible solutions in terms of two factors - the implementation effort and the expected impact - to prioritize those that will be put into action.

5.1 Proposal solutions

To solve the problems identified in the previous chapter, the following possible solutions were suggested in the table 15.

Table 15 - Suggested solution per problem identified

ID Solution	*Suggested Solution*	*Problem ID*
S1	Standardize budget request	1; 3; 4; 5; 27
S2	Create a training plan for the current team of sales accounts;	2
S3	Standardize the response to the budget	6; 7
S4	Create handoff processes between the sales team and the IT	8
S5	Create a support tool for team management	9; 17; 18
S6	Create a process of documenting developments	10; 21
S7	Standardize the functional dossier	11
S8	Create handoff processes between the IT and client	12; 28
S9	Re-design the functional dossier creation process	13; 14; 15; 19; 20; 25
S10	Re-design the planning process	16; 22
S11	Create a process of testing	23
S12	Create a training plan for developers	24
S13	Re-design the project creation process	26

There are two main considerations about this table: one is the relation between one solution with the different phases of the project, the other is the benefit of a solution from the implementation of the others.

So, while the problems are related to the specific stages of the process, the suggested solutions may have an impact on the different parts of the process. For example, when we suggest

standardising the budget request it will impact not only the overall lead time or the budgeting lead time but in fact will impact the other phases of the process.

When we refer to the relationship between solutions, we are thinking, for example, if we create a support tool for management in which the team can see the carrying capacity or the challenges it faces, we can act on the training needs of the team itself.

5.2 Impact- Effort Analysis

After presenting the various improvement solutions that address the problems encountered, the next stage of this project included the prioritisation of such solutions through the execution of an impact-effort analysis.

An impact effort matrix is a decision-making tool that assists people to manage their time more efficiently. An organisation, team, or individual assesses activities based on the level of effort required and the potential impact or benefits they will have.

Ideally, solutions are expected to generate increased positive benefits for the company and, simultaneously, not be difficult to implement - do not take long to be implemented, does not have significant costs associated, and does not require the collaboration of many employees, namely from other functional units. However, natural trade-offs may be considered.

The result is a visual representation of where best to assign time and resources. Activities fall into one of four categories:

- Quick wins – Give the best return based on the effort (high impact – low effort).
- Major projects – Provide long term returns but may be more complex to execute (high impact – high effort).
- Fill ins – Do not require a lot of effort but neither do they offer many benefits (low impact – low effort).
- Time wasters – Time-consuming activities with low impact that should be avoided (low impact – high effort)

So, the suggested solutions to improve TBFiles SD and PM regarding the level of impact in lead time and the effort to implementation are represented through the figure 10.

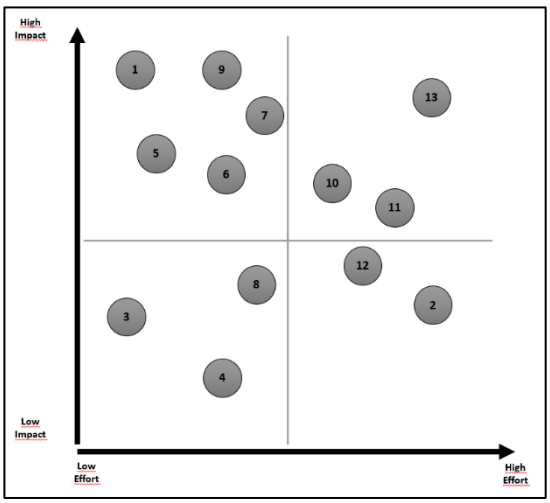


Figure 10 - Effort impact analysis for proposed solutions

Taking into account the assessment made for each proposed solution, the standardisation of the budget request (S1), the creation of a support tool for team management (S5), the creation of a process for documenting developments (S6), the standardisation of the functional dossier (S7) and the reformulation of the process of creating the functional dossier (S9) was selected to be designed, given the expected impact it will have on all stakeholders.

Another important consideration is that after looking to the results of our impact effort analysis and the problems that each solution address we can conclude that these 5 solutions cover 17 out of 28 problems found throughout the process.

5.3 Developed Solutions

The last step of this project is to design the solutions. Therefore, taking into account the several authors who address the different methodologies of software development and the characteristics and related activities for successful project management, we chose to analyze the solutions to be developed, such as team organization, standardization and process redesign.

Before going deeper into each of the solutions, we understand that it is important to remember the problems identified, their root cause and relate them to the general idea of each solution.

So, we believe that the problems identified as IT do not meet deadlines before acceptance (P9), planning is not carried out in advance (P17), and IT does not always participate in the roadmap (P18) has its main causes is the lack of internal and external visibility of the team capacity load. In fact, IT is unable to plan work in advance because it is unaware of the workload it faces.

So, we believe that with a more agile approach to SD, specifically using some of the techniques and events of Scrum we managed to improve the current state of the situation in relation to the team management (S5).

Inefficient processes and lack of standardisation are problems associated with quality, which we perceive that in the end negatively influence the lead time of projects.

So, regarding the lack of processes to create the functional dossier (S9) and documentation of developments (S8), we opted for the solution of redesigning the entire process of creating customised solutions by TBFiles. With the reformulation of these processes we believe we can address problems like there is no acceptance by the client (P12), functional dossier is not carried out (P13), there are several versions of the same document (P14), document executed simultaneously with the development (P15), the fact that there are no priorities for tasks (P19), unclear tasks (P20), features do not match customer requirements (P25), and the presentation meeting does not effectively empower the client (P28).

About the problems of missing information from the customer/sales account (P1), deficient IT / commercial communication (P3), lack of detail in the budget (P5), most times there is no closed period for pilot (P27), and the fact that the functional dossier has little detail (P11) we assume these problems are related to the lack of standardisation of documentation related to the budget (S1) and the functional dossier (S7) in TBFiles.

For Sharp, A. and McDermott, P. (2009) because successful process redesign projects usually involve major information systems efforts, and the implementation of a significant information system usually involves redesigned processes.

So, a more agile methodology for SD, the creation of new functionalities in the project management software (PMS), process redesign, and the standardisation are the proposed solutions that we will develop in the next points.

5.3.1. Scrum Solution

According to Wong, K (2019) Scrum is an Agile Methodology consisting of lightweight management practices that have relatively little overhead. This methodology is based on three pillars: transparency, inspection, and adaptation.

Our solution was based on the pillar of transparency, as Wong, K. (2019) says, everyone, can see all parts of the project, whether the team or not.

As already mentioned according to Pinto, P. (2008) Scrum projects progress through a series of iterations called “sprints”, and at the beginning of each sprint, the team determines the amount of work it can do during that sprint. The job is selected from a prioritised list called "product backlog". The work that the team believes it can complete during the Sprint is moved to a list called "Sprint Backlog". A brief daily meeting, the “daily scrum”, is held to allow the team to inspect their progress and adapt as needed.

According to the same author the scrum outlines four required techniques within a sprint: sprint planning, daily scrums, sprint review and project retrospective.

To implement this methodology in TBFiles we must adapt the ideas from the different authors to the reality of the company. So, to design the solution we need to consider the definition of roles and responsibilities, the ceremonies of the development cycle, and the strategies for implementing the model.

5.3.1.1. ROLES AND RESPONSIBILITY DEFINITION

According to the literature review scrum defines some roles besides the software developers, the product owner, and the scrum master. The product owner is the person who is responsible for the product, all the product requirements and the priorities gathered in a product backlog, come through this role. The scrum master helps the product owner and the development team by suggesting techniques to manage product backlog and attain requirements.

As already mentioned, we feel the need not to radically break with the current reality of the company. So, despite maintaining the responsibilities of each of the roles in the new development methodology, we designate that the product owner responsible for the project and this role was assigned to the current IT manager. This is because he is the most experienced person and able to list and prioritise the features better. The current project manager assumes the functions and denomination of the Scrum Master in new projects.

New fields have been added to the TBFiles’ PMS as we can see in the project creation view in Appendix I.

5.3.1.2. SPRINT

According to Cadle J. and Yeates D. (2008) the duration of the sprint should be 30 days. For Wong, K (2019) the duration of 15 days seems to be ideal. However, according to Zamith, M. (2018), the duration of the sprint will depend on the reality of the company, project or product that is being developed.

For Zamith, M. (2018) each sprint has a goal to increment the product and in the end of each sprint a functional solution should be the output.

In this specific case, it was defined that in a meeting the allocation and evaluation of tasks will be carried out every 7 days. At the end of this period, a more formal meeting should be scheduled with the interested parties to demonstrate the developments so that, in case of approval, the functionalities can be placed in a production environment.

We realised then that there is little consensus regarding the duration of the sprints. Thus, in order to increase the flexibility of the software solution, in the new features developed there is the possibility to configure the dates and durations of the sprint as we can see in Appendix J.

5.3.1.3. DOCUMENT USER STORIES

According to Wong, K. (2019), user stories are a consistent format for expressing requirements that are easy to write, read and evaluate. Therefore, these are the main techniques for expressing requirements. They take the following format: "As a **who**, I want to **what**, so that **why**". That is, who is the role of the stakeholders for whom the requirement is being formed. What is the specific task or functionality that the interested party wants to accomplish using the product. And Why provides the benefit of requirements. With this method, we can have a clear and structured way to express the requirements without needing too much technical detail.

The several authors refer that user stories should be short, clear, understandable, and actionable.

Although in these methodologies the customer is often requested to document the requirements in user story format, in the case of TBFiles this request was made to the sales team, with acceptance tests being on the side of the IT team.

A workshop, in which the criteria of correction, quality, clarity, consistency, viability, and traceability, were addressed, was then held with the entire sales and development team for the teams to be aligned on how the requirements should be documented.

5.3.1.4. SCRUM EVENTS

SPRINT PLANNING

According to Zamith, M. (2018) each sprint starts with a planning meeting. This ceremony aims to define the scope of the sprint and the breakdown of the tasks that make it possible to achieve it. The Product Owner has the responsibility to present a product backlog ordered by priority and updated to that the team can make estimates for each task to include in the sprint.

For Wong, K (2019) a product backlog is a list of features that the development team plan to develop for a product over the course of a project. Tasks, bugs, and user stories are the features included in the product backlog.

Our proposal went through changes in the current software with features that responded to this planning need, as we can see in Appendix K. This new version allows the project manager (IT manager) to view the tasks in the backlog and their time estimated completion time, assign tasks to different developers according to different priority criteria and evaluate their completion. The main functionality of the developers is to be able to view the tasks assigned to them.

DAILY MEETINGS

For Cadle J. and Yeates D. (2008) daily Scrum is the daily 15-minute meeting of the development team in which problems and issues are resolved and priorities redefined.

For Wong, K. (2019) this meeting usually occurs at the beginning of the day with the objective of synchronize the development team. Zamith, M. (2018) adds that it must have the objective of coordinating the entire team and stimulating communication between everyone, which should happen on a daily basis.

To organise these meetings, our proposal was a creation of a team table, as we can see in APPENDIX L, which includes a work plan for the days of the week, the task pipeline, and the waiting tasks.

Despite how the literature advocates each person answers questions such as what did you accomplish yesterday? What will you do today? Are there any impediments in your way?

To prepare for these daily meetings, we met with the development team and explained the purpose of future meetings by addressing problems that the literature points out as common in these meetings. So, problems were addressed and a set of rules such as the last arrival speaks first, do not exceeds 15', in every meeting at least one impediment has to be discuss, will occur every morning at 10 a.m., were defined by the team.

SPRINT MONITORING

According to Wong, K. (2019) monitoring allows everyone to know the status of a project at any given time.

So, to this methodology tasks should be placed in different categories on the task board such as to-do, in progress, to verify, verifying, and done. This functionality already existed in the previous version of the PMS. In the developments of this new version, the option was to make the status of tasks visible in the new view as we can see in the Appendix J.

Another important consideration about the changes we produced to resolve lack of project visibility by other stakeholders was to parameterise the solution was to give visibility awards to other project stakeholders, such as sales and operations teams.

SPRINT REVIEW

According to Zamith, M. (2018) the sprint review meeting serves the purpose of providing a collaborative environment to analyse the value that has been built in the course of the sprint and discuss what are the next tasks to achieve.

For Wong, K. (2019) sprint review meetings are informal events where development team could demonstrate their product and collect feedback from stakeholders.

For the case of TBFiles, it was defined that every 7 days, end of the sprint, the developments must be demonstrated by the developers, approved by the product owner (IT manager), the resources must be removed from the product backlog and the parts stakeholders should provide feedback and suggest new features.

PROJECT RETROSPECTIVE

Wong, K. (2019) defines project retrospective as a team review of the entire project after it is complete. The aim of this event is to examine what went well, what did not, as well as identify how the project could be improved. For Zamith, M. (2018) the team should do this reflection so that the process can be better when applied to future projects.

In TBFiles it was decided that this moment will become part of the PM and development process. Those involved in the different areas must be summoned by the IT Manager. And reasons like the lack of time, the lack of desire to look at the past, and the fact that people do not want to expose themselves, cannot be an excuse for this more formal meeting not to happen.

5.3.1.4. OTHER STRATEGIES AND TOOLS

For Zamith, M (2018) a Test-Driven Development (TDD) is an approach to the coding process that encourages programmers to write tests for their code before writing the code. In this development style, the order program must start to write a single unit test for describing a part of the program, since this will fail once that hit is missing. Then, you should write the code in your most simplified way to make this code and meet this standard.

The same author refers that refactoring is a practice that consists of improving the internal structure of the existing code, preserving its functional behaviour. This concept is currently linked to TDD.

These two concepts are intimately linked as agile once they provide an 'antidote' to the problem with growing complexity of projects.

5.3.2. Process Redesign and Standardisation

For Boehm, B. and Turner, R. (2004), agile and plan-oriented approaches to development have generally been seen as opposing views, and ideas still seem to remain conflicting. However, every successful venture in a changing world requires agility and discipline. This is as true in sport, in art, in business development as it is in SD. Large companies and large software projects have adequate measures for their objectives and environment.

This introduction seems pertinent to us, because our approach at this level may seem too rigid, as we point to a clearly defined process with standardized documents throughout the process.

However, according to Wong, K. (2019), in a natural flow through SD, there are also PM activities in progress, such as the creation of a process, the definition of standards, among others. For this author, in any SD method, an important starting point is to establish a process and set standards to ensure that the team's activities are carried out.

Taking this into consideration we redesign the future process of creating a customised software solution with the team. In the next few paragraphs, we will detail the new process and documents that will ensure the expected improvements.

This process is triggered by the visit of the sales account to the customer's company. The first differences are in the budget phase. The customer will need to create the feature specification together with the sales account using user stories. Thus, when the IT department receives this document, their understanding of what is intended by the customer and the estimation of the development phase will be clearly facilitated.

There was no standard document or place to attach it to the budget request interface. A functionality was then developed that allows the sales account to attach this document as we can see in appendix M.

The assumption in this new document, which we can observe in appendix N, is that it is filled in by the commercial and complemented by IT in order to be the basis of the proposal that will be presented to the client. So, there was a need to develop two new features. One that allows the document workflow and the other to attach that document to the client's file.

The process is thus no longer iterative between the client and TBFiles but is only incremental between the two departments.

All documentation and planning activities take place after the customer's decision to proceed. At this moment there was also a break with the past. The planning and documentation process will be carried out by the project manager with the sales account and a team leader of the operations department. The results of this meeting will necessarily be a draft of the functional dossier containing the scope of the project, accepted features, a macro planning, and an appointment for a kick-off project meeting.

In this kick-off meeting, the final document must be presented and signed by all stakeholders, including off course the client.

As we already mentioned during the development the IT manager is the responsible for the sprints planning. Every 7 days, developments must be demonstrated and approved by the IT Manager and suggestions for customer improvements must be evaluated. After each meeting a minute of the meeting should be attached to the documents of the project.

Simultaneously at the end of each sprint, an independent team must carry out an internal audit with the aim of checking the main deliverables. As a result of this audit process, a non-conformity report (Appendix O) must be prepared and delivered to the Product Owner, so that corrective measures can be created. These measures must be transformed into tasks and included in the product backlog, so that non-conformities are corrected.

As agile development process is iterative and incremental with the different stakeholders, same of the sales team responsibility no longer exists.

Then, in the last meeting where the developments are demonstrated, the dates of the test phase and pilot period must be defined considering the period in the budget.

The test phase already takes place in the production environment. What we proposed was the creation of a test environment before going to production. The code is review, and the uniformity of the nomenclature and the documentation of the code is verified so that there is the least difference in the code of the different programmers. this is a change with the past

delivery method because the code is now reviewed by the IT manager also regarding efficiency and not just in relation to effectiveness.

The process finishes after the pilot with the moment pre-defined to go live with a kick-off meeting.

6 Conclusion and future research

To guarantee its adaptation to the dynamic reality of the businesses in which it operates the main objective of the research was to study the current situation of the company, to identify the root causes of the problems and to develop possible solutions, foreseeing its benefits and difficulties of implementation, in order to guarantee improvements in the software development.

So, the software development and project management processes were selected as targets of this research due to their impact on the company's business performance and its long-term sustainability.

In this last chapter of the dissertation, we will present the conclusions that were drawn throughout the project, considering the research objectives described in the first chapter, and point out possible limitations and future directions of research.

6.1 Main conclusions

Given the new market reality, the main objective of this research was to identify problems in current processes and practices, developing possible solutions to achieve operational excellence in the development and project management of customised software solutions.

The main research objectives defined at the beginning of the project were satisfactorily achieved through a continuous focus on answering the research questions.

It was found that TBFiles, regardless of basing its strategy on an ambitious organic growth since its creation, opted along its growth for strategic acquisitions to increase its presence in the national market.

It was found that the business expansion has been achieved through the increase of its production and sales, focusing on a differentiation strategy centred on the customer, on the quality of the service provided, and on the continuous development of products and services that solve the problems related to dematerialization and digital transformation of its customers.

Therefore, customised software solutions have gained increasing importance in the TBFiles business, meaning that the processes related to the development and management of these solutions have been chosen as a target for our project.

The first conclusion that we get in this work after informal conversations while observing the performance of the tasks is that although the context has changed the modus operandi and the structure has remained practically the same.

A qualitative approach to research, informal workshops, and data analysis allowed the identification of relevant metrics in the analysis of the company's software development and project management processes, thus managing to answer our first research question (RQ1).

Although the studied literature points to indicators related to the time, quality and cost of the projects, the two defined and analysed are more related to time, such as project delivery time and on-time delivery, and with quality, rework. This decision is due to the expectation that the impact of the solutions to reduce in 20% the lead time, reducing rework by about 57%, and the increase of on time deliveries to 75% will have positive repercussions on the cost of the projects.

So, when we looked to the numbers of this indicators before the project, we perceived an average of 150 working days required since the customer requests the visit of the sales account

until the solution is in a production environment and available for full use. It was possible to recognize the weight of each phase has on the lead time of the different projects, implementation showed a weight of 66%, while acceptance and budgeting presented very similar percentages, 16% and 17% respectively. Others important conclusions were made through this analysis, we realized that 80% of the projects did not finish in the estimated time and 70% of the projects needed rework.

When we set out to identify problems that affect processes, opportunities for improvement, and the root causes of problems, thus responding to RQ2 and RQ3, we faced with a great challenge since the beginning, the procedures are mainly ad hoc.

However, following a qualitative research approach, direct observation of the team's tasks, document analysis and informal workshops allowed the identification of 28 causes or opportunities that were affecting the failure to achieve the expected results in each phase of the process.

After the causes have been identified to understand and analyse the root causes, we chose the diagram of Ishikawa also known cause-effect diagram. We can then see that the main problems with the project management and software development processes are related to the lack of visibility of the team load, lack of standardisation and deficient management processes. These causes result in problems such as IT being unable to plan the work in advance and problems associated with quality that negatively influence the lead time of projects.

Given the duration of the project, after identifying the opportunities for improvement, the problems, and understanding their root causes, the next phase of the project focused on formulating and prioritizing possible solutions to correct or mitigate the problems encountered by conducting an effort-impact analysis and additional workshops with process participants, responding to RQ4.

The defined solutions demonstrate ways in which current project management and software development processes can be improved and, specifically, how these changes can be implemented in the context of the company, overcoming existing problems.

So, we believe that with a more agile approach to SD, specifically using some of the techniques and events of Scrum we managed to improve the current state of the situation in relation to the team management.

Regarding inefficient processes and lack of standardisation we opted for the solution of redesigning the entire process of creating customised solutions by TBFiles.

As successful process redesign projects often involve major efforts in information systems, we have also implemented several changes to the current project management software to adapt it to new changes.

These two approaches, one more related to agility and other to discipline, are often seen in most literature as contradictory. However different authors refer that any company requires discipline and agility to succeed. Discipline is the basis for the success of any work, and that provides strength and comfort in complicated situations, creates organization and experience, while agility is characterized by the ability to react to changes and adapt to new environments outside the comfort zone. While advocates of agile methodology criticize traditional processes, on the other hand, they respond with accusations of lack of quality and rigour, not to mention an environment that is too relaxed to encourage professionalism. The conclusion that made us

move towards this solution is that there are different circumstances in which each one can improve the project. Therefore, there is no reason to justify the fact that they cannot be combined in a hybrid process that tries to collect the best characteristics of each approach, through a series of factors. In fact, we realised that the implementation of solutions related to a more agile methodology will positively affect the implementation of more structured processes.

All changes were accepted and prepared to be implemented, however not all were initiated given the time of the project.

Despite the significant involvement of all participants throughout the course of the project, which is crucial for the generation of improvement actions and implementation of the selected solutions, organizational logistics was a major challenge and time consuming to the project.

The involvement of all participants throughout the project was significant and critical for the generation of actions to improve and implement the selected solutions.

Thus, this dissertation should serve as a starting point for future change projects to be carried out in the company, following a continuous improvement approach.

6.2 Future research directions

This dissertation followed a mainly qualitative research approach due to the lack of data, which limited the quantification of causes and metrics. Therefore, future research guidelines include the development of a project with the objective of enriching and complementing the analysis carried out.

Given the average duration of TBFiles development projects, an average of 150 working days and the team's capacity of around 9 projects per year, we immediately set aside the possibility of assessing the solution's impact on the defined indicators, time, quality and cost. In this sense, future research directions can focus on analysing the results achieved and, if necessary, starting new process improvement projects.

According to the literature, quality is a problem that cannot be solved in a timely manner. The thinking of each team member must be oriented towards the process and quality. To this extent, the best way to perform a task must be found and established. We thus point out the possibility of developing future concepts such as improvement cycle, PDCA (Plan / Execute / Verify / Act), or SDCA standardisation (Standardize / Execute / Verify / Act), or even a Stage Gate Model.

Given the short project time and the company's priority is to improve the development of customised solutions and the management of projects associated with this solution, it would be interesting to see if this development model would be applicable to the company's other solutions and other business contexts because the literature is inconclusive and controversial.

Simultaneously, although we realize that by improving the processes of developing this kind of solutions, it directly impacts processes common to other solutions, another work that we advise in the future would be to focus each of these processes in isolation.

We found that some of the analysed processes are directly related to other departments within the company, such as the budgeting process. Thus, it would be interesting to analyze the impact of improvements in the budgeting process on the percentage of conversion from prospect to a client (hit rate).

As a proposal for future studies in this area, we also recommend focusing on the impact of a change in the organizational structure. TBFiles has a very vertical model of organizational structure, which, as the literature points out, can hinder an improvement process. Thus, future research can focus on assessing the impact of changing the current model to a more horizontal structure model.

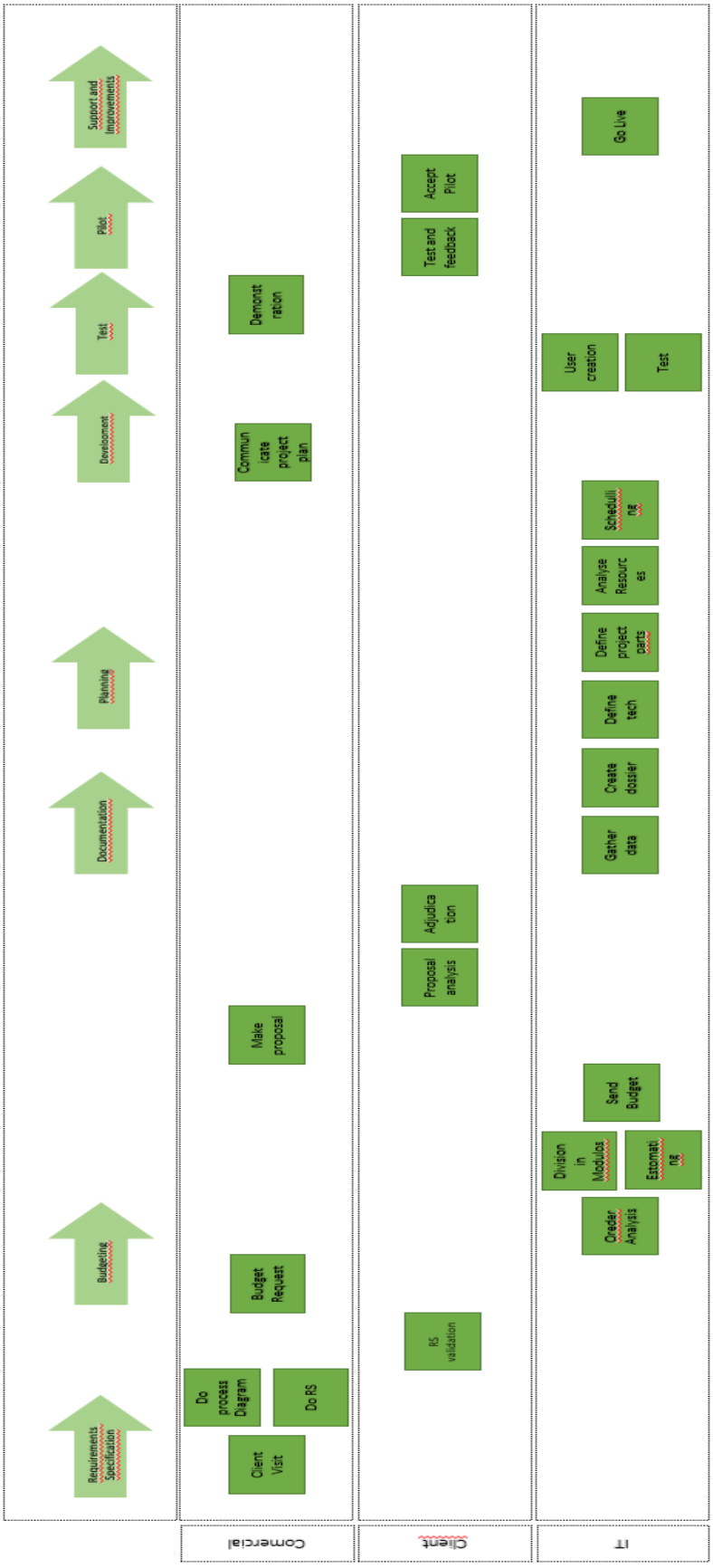
Finally, in addition to the limitations mentioned throughout this chapter, we also point out the fact that the context of our project is the specific context of TBFiles. Thus, future investigations may choose to evaluate the applicability of our solutions in different contexts.

Refrences

- Agarwal, N. and Rathod, U. (2005), "Defining success for software projects: An exploratory revelation", International Journal of Project Management, Vol. 24 No. 4, pp. 258-370.
- Arlow, Jim and Neustadt, Ila (2002) UML and the Unified Process. Edited by Pearson Eductaion.
- Bhuiyan, N. and Baghel, A. (2005), "An overview of continuous improvement: from the past to the present", Management Decision, Vol. 43 No. 5, pp. 761-771
- Buttle, F. and Maklan, S. (2015) Customer Relationship Management: Concepts and Technologies, 3rd ed., Routledge
- Cadle, J. and Yeates, D. (2008), Project Management for Information Systems, 5th ed., Pearson Education Limited, England.
- Cervone, H. (2011), "Understanding agile project management methods using Scrum", OCLC Systems & Services: International digital library perspectives, Vol. 27 No. 1, pp. 18-22.
- Coimbra, A. (2016), Kaizen: Uma estratégia de melhoria, crescimento e rentabilidade. Edited by McGraw-Hill.
- Downs, L. and Nunes, P. F. (2013), “Big Bang Disruption,” Harvard Business Review (91:3), pp. 44-56
- Fitzgerald, M., Kruschwitz, N., Bonnet, D. and Welch, M. (2013) Embracing Digital Technology, MIT Sloan Management Review.
- Guedes, A. (2019). SOML Notes. FEUP-UP. Department of Industrial Engineering.
- Hall, E. (1998): Managing Risk: Methods for Software System Development, Addison-Wesley, New York
- IEEE (1993). EEE Standards Collection: Software Engineering. In IEEE Standard 610, pp.12-1990.
- Hamza, S. (2008) Design process improvement through the DMAIC Six Sigma approach: a case study from the Middle East. International Journal of Six Sigma and Competitive Advantage, pp. 35-47.
- Heldman, Kim, Patti M. Jansen, and Claudia M. Baca. 2005. PMP: Project Management Professional Study Guide. Edited by Wiley.
- Horlach, B., P. Drews, and I. Schirmer, (2016). "Bimodal IT: Business-IT Alignment in the Age of Digital Transformation", Proceedings on the Multikonferenz Wirtschaftsinformatik, pp.1417–1428.
- Johnston, R, Clark G. and Shulver, M. (2012) Service Operations Management: Improving Service Delivery, 4th ed., Pearson Education Limited, England
- Keil, M., Lee H., and Deng T. (2012) Understanding the most critical skills for managing IT projects: A Delphi study of IT project managers. Information & Management, Vol. 50 No. 4, pp. 398-414.

- Kerzner, Harold R. (2013). Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Edited by John Wiley & Sons. 11th ed.
- Lakein A. (1973), How to Get Control of Your Time and Your Life. Edited by Signet. 11th ed.
- Paulk, M C, Curtis, B and Chrissis, M. B. (1991) Capability Maturity Model for Software, Software Engineering Institute
- Pereira, C. (2008) Desenvolvimento do processo de gestão de projectos no INEGI. Master Thesis. FEUP-UP
- Pinto, P (2008). Software Improvement Process at CERN CO-AP. Master Thesis. FEUP-UP.
- PMI (2013). Guia PMBOOK. 5th ed
- Sebastian I. et al. (2017). How Big Old Companies Navigate Digital Transformation. MIS Quarterly Executive, 16, pp.197–213.
- Sharp, A. and McDermott, P. (2009). Workflow Modeling: Tools for Process Improvement and Applications Development: Artech House.
- Shenhar, A. J. (2001). One size does not fit all projects: Exploring classical contingency domains. Journal of the Institute for Operations Research and the Management Science, 47(3), 394–414.
- Sheetal Sharma (2012). Agile Processes and Methodologies: A Conceptual Study. International Journal on Computer Science and Engineering (IJCSE).
- Spundak, M. (2014). Mixed agile/traditional project management methodology. Reality or illusion. Social and Behavioral Sciences 119, 939-948.
- Urbach, N., Ahlemann, F., Böhmann, T., (2017). The Impact of Digitalization on the IT Department. Bus Inf Syst Eng 61, 123–131
- Wong, K. (2019). SPMS Notes. University of Alberta. Department of Computing Science
- Zamith, M (2019). Desenvolvimento de Software para dispositivos méidocs: uma abordagem ágil. Master Thesis. FEUP-UP.
- Zamudio, L., Aguilar, J. A., Tripp, C., & Misra, S. (2017). *A Requirements Engineering Techniques Review in Agile Software Development Methods.* Paper presented at the International Conference on Computational Science and Its Applications.

APPENDIX A:Process Mapping



APPENDIX B: Projects' data

Project	First Visit	Budget Request	Budget Send	Final Budget	Proposal Final Version	Acceptance	Estimated days	Estimated delivery	On Time Delivery	Delivery	Rework	Final Delivery
1 (1575)	18 January 2019	23 January 2019	26 February 2020	6 March 2019	7 March 2019	18 March 2019	14	5 April 2019	No	23 October 2019	Yes	5 February 2020
2 (1165)	7 January 2019	22 January 2019	23 January 2019		24 January 2019	26 April 2019	50	15 July 2919	No	15 August 2019	Yes	1 October 2019
3 (1578)	18 January 2019	22 January 2019	6 March 2019	17 April 2019	17 April 2019	15 May 2019	12	15 June 2019	Yes		Yes	18 July 2019
4 (1802)	7 February 2019	4 April 2019	12 April 2019		12 April 2019	11 May 2019	40	16 August 2019	No	20 February 2020	No	
5 (0478)	10 February 2019	17 February 2019	5 March 2019		9 March 2019	26 March 2019	36	Not defined	Yes	1 June 2019	No	
6 (0075)	26 March 2019	26 March 2019	12 April 2019		13 April 2019	4 May 2019	16	5 June 2019	No	2 July 2019	Yes	20 August 2019
7 (0001)	5 May 2019	6 May 2019	7 June 2019		8 June 2019	15 July 2019	70	31 October 2019	No	15 November 2019	Yes	15 February 2020
8 (1578)	11 September 2019	15 September 2019	3 October 2019	17 October 2019	17 October 2019	27 October 2019	33	27 November 2019	No	13 December 2019	No	
9 (0001)	26 April 2019	26 April 2019	2 May 2019	12 May 2019	13 May 2019	4 June 2019	21	5 July 2019	No	2 August 2019	Yes	20 September 2019
10 (0478)	15 October 2019	17 October 2019	5 November 2019		9 November 2019	29 December 2019	24	15 February 2020	No	15 March 2020	Yes	25 March 2020

APPENDIX C: The rationale of problems and root causes of the budgeting phase

Problem ID	*Problem*	*Root Cause*	*Rationale*
P1	Missing information from the customer/sales account	Lack of standardisation of budget request	Because the customer does not have everything fully defined. Because? Because we don't ask the right questions
P2	Difficult interpretation in requirement specification	Lack of standardisation of budget request	As no one from IT participates in the first meetings with the customer, neither the commercials have IT training/knowledge, they require badly defined, poorly defined scope, etc. Which leads to the need for several communications to understand the customer's needs.
P3	Deficient IT / commercial communication	Lack of standardisation of budget request	Absence or deficient communication between departments leads to not understanding all the requirements, so the budgets and even the final solutions are presented not meeting expectations, which leads to rework that leads to increased lead time.
P4	Subjective criteria or lack of sensitivity to estimate hours	Lack of standardisation of budget request	As it is not clear what the client needs, the estimated hours are given with safety margins, which subsequently leads to development in the estimated period and not in the actual development period.
P5	Lack of detail in the budget	Lack of standardisation of budget request	As there is no response standard, the IT department responds to the commercial that has difficulty creating a proposal for the client to support the budget, which will lead to the need for extra communications between departments.
P6	Difficult interpretation of the budget given	Lack of standardisation in the response to the budget	As there is no response standard, IT responds to the commercial that has difficulty understanding the details of the budget, resulting in extra communications between departments in order to formulate the budget.
P7	Constant hourly renegotiation	Lack of standardisation of budget request	Because IT only sends the ideal solution and not the different solutions so that the commercial can negotiate with the customer.

Problem ID	*Problem*	*Root Cause*	*Rationale*
P8	Lack of visibility about the acceptance from the client	Lack of processes after customer acceptance	There is no clearly defined process and responsible after the acceptance of the proposal by the client, which leads the IT to delay the start of the development process.
P9	IT does not give deadlines before acceptance	Lack of visibility of the team's capacity load	The IT manager is unaware of the team's capacity load
P10	Lack of technical specifications	Lack of the process of creating documentation	Because the development process does not "oblige to document what is developed, so whenever the customer requests technical specifications the document will have to be created.

APPENDIX D:The rationale of problems and root causes of the documentation phase

Problem ID	*Problem*	*Root Cause*	*Rationale*
P11	Little detail	Lack of standardisation of the functional dossier	The lack of certain requirements in the functional dossier which makes the scope too wide.
P12	There is no acceptance by the client	Lack of acceptance of the functional dossier by the client	Lack of acceptance of the functional dossier by the client means that in the future the requirements are not met and requires rework to meet expectations.
P13	Functnal dossier is not carried out	Lack of the process of creating the functional dossier with stakeholders	The requirements are not met and require rework to meet expectations which will affect the lead time.
P14	There are several versions of the same document	Lack of the process of creating the functional dossier with stakeholders	There are many versions of the same document because it is sent to many people, who are involved in its creation. What hinders the progress of this phase.
P15	Document executed simultaneously with the development	Lack of the process of creating the functional dossier with stakeholders	The requirements are not met and requires rework to meet expectations which will affect the lead time.

APPENDIX E: The rationale of problems and root causes of the planning phase

Problem ID	*Problem*	*Root Cause*	*Rationale*
P16	There is no time set aside for unforeseen events	Lack of processes of planning	Unforeseen circumstances will lead to more work.
P17	Planning is not carried out in advance	Lack of visibility of the team's capacity load	The lack of visibility of the team's load means that it is not possible to plan in advance, which may result in the work not starting on the ideal date to finish within the estimated time, thus increasing the lead time.
P18	IT does not always participate in the roadmap	Lack of visibility of the IT department's capacity load	IT does not always immediately start working on a project or order because it has work in progress. The other teams do not take this into account and think that the lead time they were able to start immediately.

APPENDIX F: The rationale of problems and root causes of the development phase

Problem ID	*Problem*	*Root Cause*	*Rationale*
P19	There are no priorities for tasks	Lack of standardisation of task documentation	As the tasks are not detailed, they do not "exist" for the team to be able to make them worse, which may lead to "bad" developments and consequently rework.
P20	Unclear tasks	Lack of standardisation of task documentation	If, on the one hand, the development of solutions that do not satisfy the customer directly impacts the lead time, this problem still gives rise to wrong priorities and inability to perceive the team's load and consequently bottlenecks.
P21	"Support" or other situations corrupt the fulfillment of the planning	Lack of documentation all developments	The support that should oversee one of the developers does not solve and assigns the task to another colleague who, given the priority of the request, may lead to interrupting the planning.
P22	There is no definition of dependencies between tasks	Lack of the process of planning tasks and dependencies	Stopping and starting can lead to errors that will lead to increased lead time
P23	Tests when done are only performed at the end	Lack of test processes	If in the end the solution does not work it can lead to rework
P24	Code review is centralized in one person	Lack of team organisation	The code review gets stuck in one person, which slows down the process (bottleneck). Why? Because there is no versatility within the team or in the testing environment.

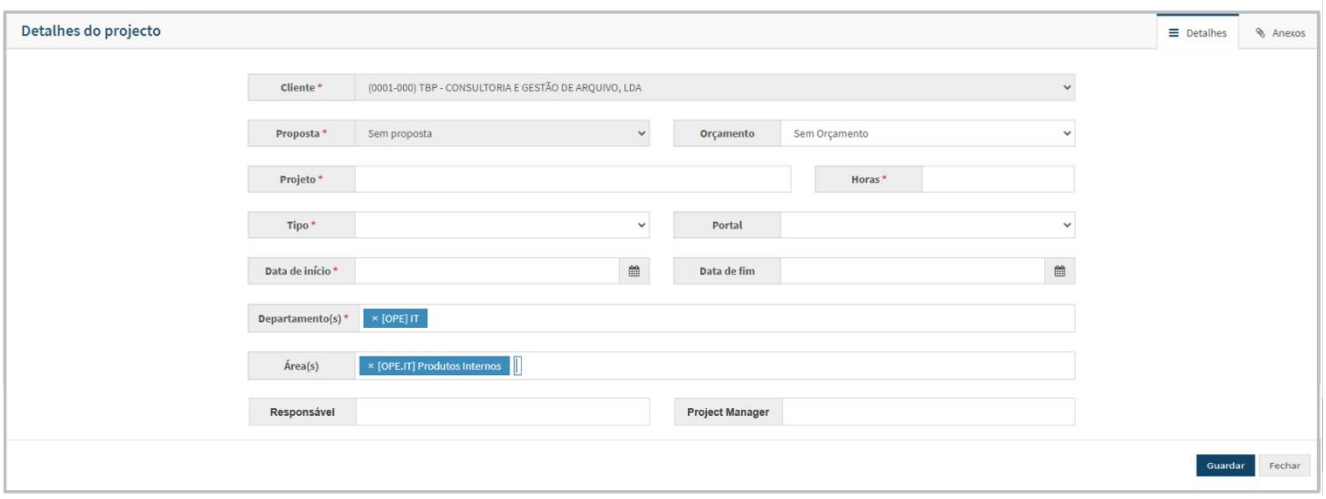
APPENDIX G:The rationale of problems and root causes of the testing phase

Problem ID	*Problem*	*Root Cause*	*Rationale*
P25	Features do not match customer requirements	Lack of the process of creating the functional dossier with stakeholders	If in the end the solution does not work it can lead to rework
P26	Tests fail due to unforeseen operational needs	Lack of the process of creating the project with internal departments	If in the end the solution does not work it can lead to rework.

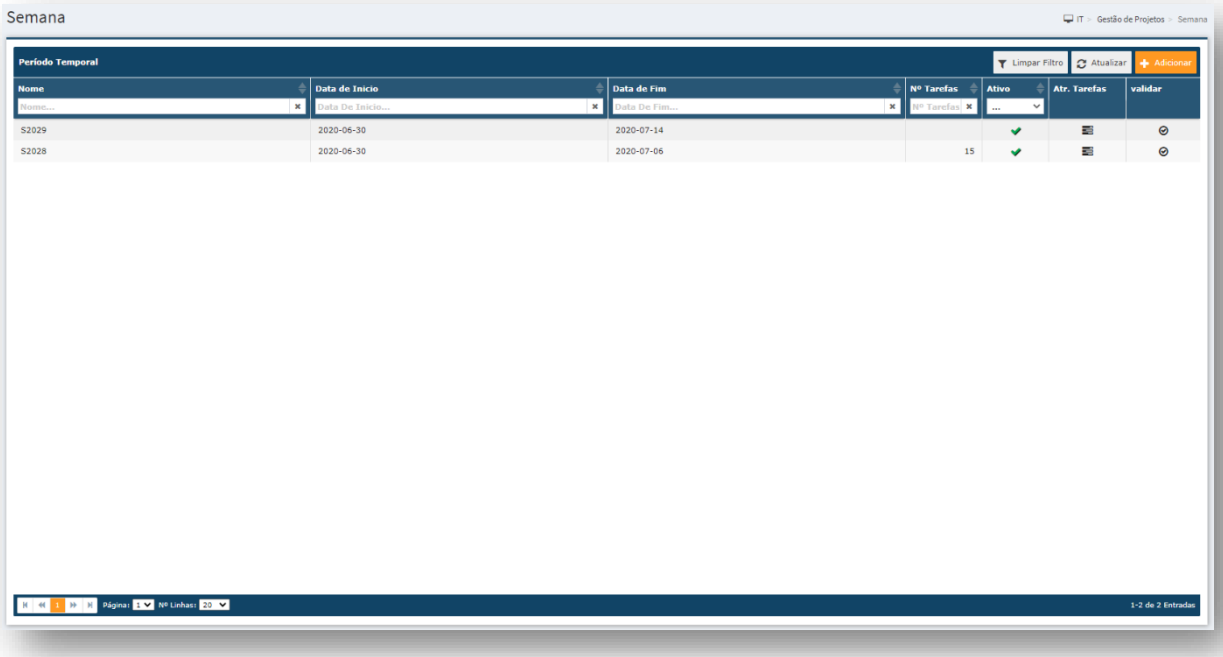
APPENDIX H:The rationale of problems and root causes of the development phase

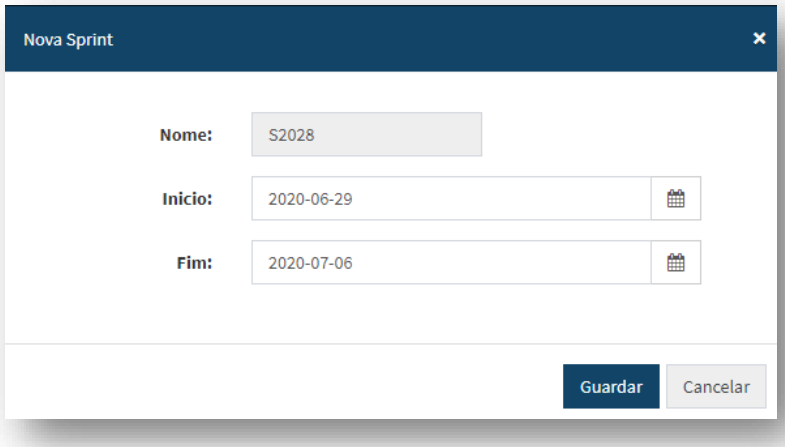
Problem ID	*Problem*	*Root Cause*	*Rationale*
P27	Most times there is no closed period for pilot	Lack of standardisation of budget request	As the pilot period is not defined, it extends beyond the delivery period
P28	Presentation meeting does not effectively empower the client	Lack of customer training processes	The customer does not understand that was delivered and therefore it takes time to test and notice improvements. Makes late suggestions, interrupts IT a lot with questions.

APPENDIX I: Modification in PMS – New fields in the Project Creation View

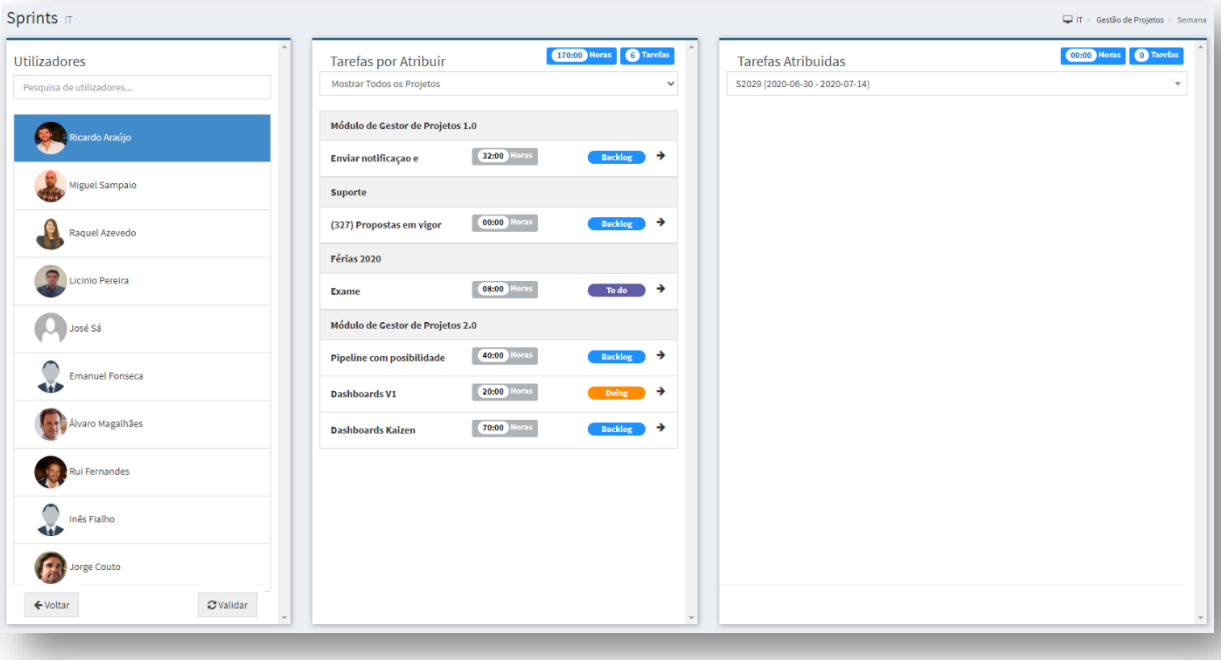


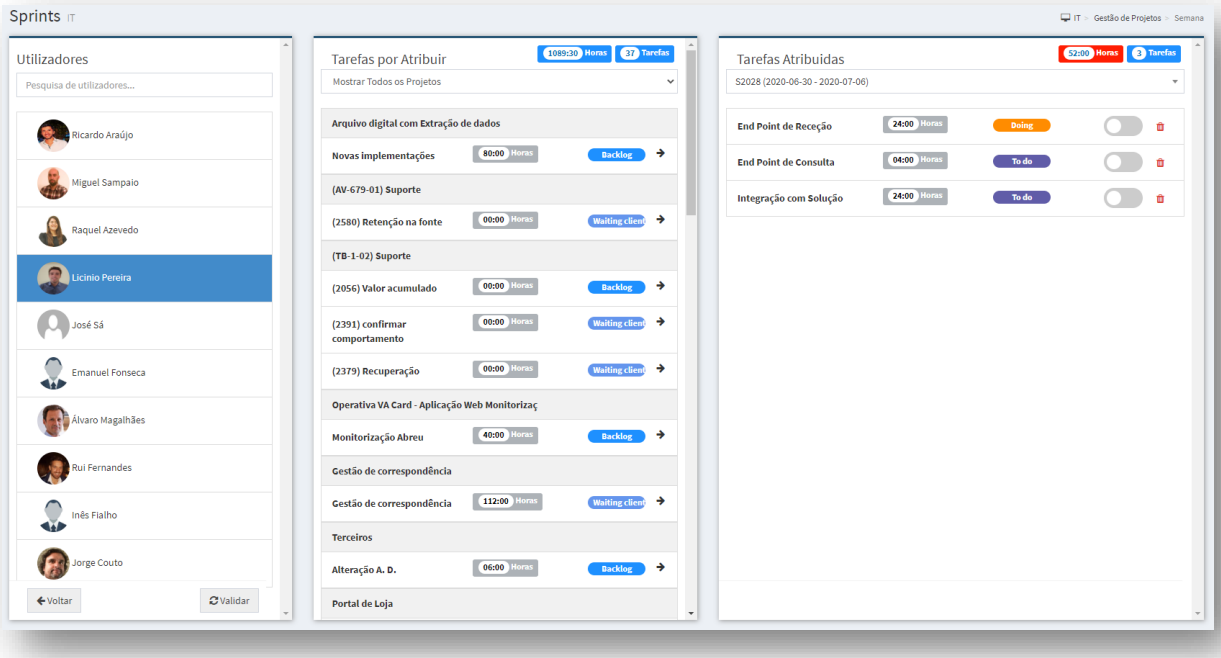
APPENDIX J: New view and features in PMS – Sprint duration





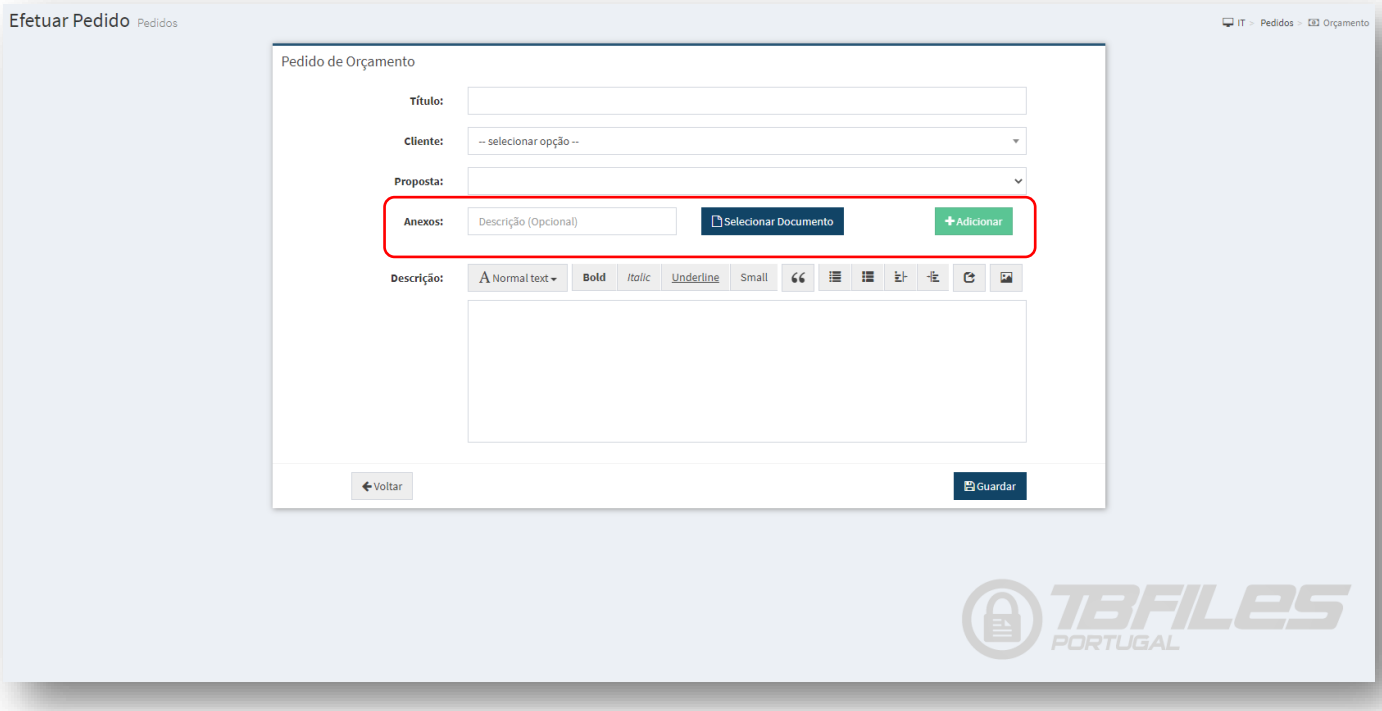
APPENDIX K:New Features in PMS





APPENDIX L: Team Table

APPENDIX M: New functionalities in Budget Request



APPENDIX N: Budget Request Document

	Orçamento	Nº.:	Data:
TBFiles Improving Together	**Cliente**		
	Principais problemas		

	Departamento Comercial	Departamento Tecnologias Informação
Realizado por		

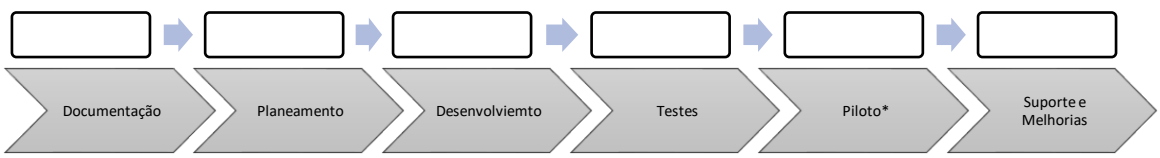
#	*User Stories*			Estimativa de Esforço*
	As a **<papel>**	***I need*** **<requisito>**	***So that*** **<objetivo>**	

Observações:

***a preencher pelo IT**

Descrição Macro da Solução:

Necessidades Emergentes:



*a fase de piloto não deve impactar o orçamento do cliente apenas os dias do projeto

APPENDIX O:Tests Report Example

	**RELATÓRIO DE TESTES**		**Portal Services**
	Projeto		
	Data		
	Módulo a testar		
https://services.tbfiles.com	**Local**		

Testes Realizados por	

Serve o presente para descrever todos os testes efetuados no módulo de Gestão no https://services.tbfiles.com/.

Gestor de Projetos | Semana	
Cenário do teste	
Caso de teste	
Pré-condições	
Etapas do teste	
Dados de teste	
Resultado esperado	
Data do teste	
Analise	
Passou/Falhou	**Etapa 1:** Passou **Etapa 2:** Passou **Etapa 3:** Falhou **Etapa 4:** **Etapa 5:**
Resultado:	**Reprovou.**
Dados de teste:	
Resultado esperado	
Data do teste	
Analise	
Passou/Falhou	**Etapa 1:** Passou **Etapa 2:** Passou **Etapa 3:** Passou **Etapa 4:** Passou **Etapa 5:** Passou
Resultado:	**Passou.**