



Lisbon School  
of Economics  
& Management  
Universidade de Lisboa

# **MESTRADO**

## **GESTÃO DE SISTEMAS DE INFORMAÇÃO**

### **TRABALHO FINAL DE MESTRADO**

#### **RELATÓRIO DE ESTÁGIO**

**METODOLOGIAS AGILE E DEVOPS NO  
DESENVOLVIMENTO DE SOFTWARE: UM CASO DE  
ESTUDO**

**TOMÁS MIGUEL GONÇALVES NINY DOS SANTOS**

**OUTUBRO - 2023**

**MASTER**  
GESTÃO DE SISTEMAS DE INFORMAÇÃO

**MASTER'S FINAL WORK**  
INTERNSHIP REPORT

METODOLOGIAS AGILE E DEVOPS NO  
DESENVOLVIMENTO DE SOFTWARE: UM CASO DE  
ESTUDO

TOMÁS MIGUEL GONÇALVES NINY DOS SANTOS

**SUPERVISION:**  
PROF. DOUTOR MÁRIO ROMÃO

OUTUBRO - 2023

## LISTA DE ABREVIATURAS

API - Application Programming Interface

CD - Continuous Delivery

CI - Continuous Integration

CTO - Chief Technology Officer

ERP - Enterprise Resource Planning

GPS - Global Positioning System

TFM - Trabalho Final de Mestrado

UI - User Interface

UML - Unified Modeling Language

WIP - Work in Progress

## RESUMO

Com o objetivo de consolidar os conhecimentos e práticas acadêmicas obtidas durante o mestrado em Gestão de Sistemas de Informação, selecionei como caracterização do TFM, a realização de um estágio. Fui apresentado com a oportunidade de estagiar na empresa Escrita Digital, uma empresa de desenvolvimento de software com foco na gestão de frotas.

O presente relatório descreve as atividades realizadas durante o estágio, complementado de uma visão sobre a minha contribuição para a organização de processos da empresa. O relatório explora a aplicação prática das metodologias Agile e DevOps dentro da estrutura organizacional, destacando como estas contribuíram para uma gestão eficiente de projetos e entregáveis.

A adoção de metodologias Agile em empresas de desenvolvimento de software tem atualmente uma grande importância, para assegurar que estas conseguem eficazmente cumprir as suas obrigações com os clientes e manter uma estrutura organizada para possibilitar o seu crescimento. Apesar de ser uma área vastamente estudada, nos casos de estudo de empresas pequenas de desenvolvimento de software, ainda existe uma oportunidade de enriquecer este tema.

A base teórica do relatório abrange as metodologias Agile e a sua abordagem iterativa e colaborativa para o desenvolvimento de software, e fornece percepções sobre as metodologias DevOps, concentrando-se na integração entre desenvolvimento e operações, para aprimorar a velocidade, qualidade e segurança da entrega de software.

São evidenciados os desafios mais relevantes enfrentados pela empresa durante os seus processos e as estratégias implementadas. As melhorias sugeridas e posteriormente implementadas pela empresa, juntamente com a prévia aplicação de princípios Agile e DevOps, possibilitaram otimizar o ciclo de vida de desenvolvimento, a integração e entrega contínuas, fomentando a adaptabilidade e a capacidade de resposta das equipas.

## ABSTRACT

With the purpose of consolidating the knowledge and academic practices gained during the Master's in Management Information Systems, I selected an internship as the characterization of the Final Master's Work. I was presented with the opportunity to intern at Escrita Digital, a software company focused on fleet management.

This report describes the variety of activities undertaken during the internship, providing insight into my contribution to the organization's processes. The report explores the practical application of Agile and DevOps methodologies within the organizational framework, highlighting how the methodologies contributed to efficient project and deliverable management, providing an understanding of the practical implementation of these methodologies in a real software development environment.

There is a significant importance in adopting Agile methodologies in software development companies to ensure effective fulfilment of obligations to clients and maintaining an organized structure for growth. Despite being a widely studied area, there is still an opportunity to enrich this theme in the scope of small software development companies.

The theoretical foundation of the report entails Agile methodologies and their iterative and collaborative approach to software development and provides insights into DevOps methodologies, focusing on the integration between development and operations to enhance the velocity, quality and security of software delivery.

The report highlights the most relevant challenges faced by the company during its processes and the strategies implemented. The improvements suggested and later implemented by the company, in conjunction with the previous application of Agile principles and integration of DevOps practices, made it possible for the company to optimize the development life cycle, enable continuous integration and delivery, fostering adaptability and enhancing the teams' responsiveness.

## AGRADECIMENTOS

A realização deste Trabalho evidencia o final de mais uma experiência académica, com pontos altos e baixos, desafios, mas sobretudo de aprendizagem e conhecimento sobre uma área que sempre desejei seguir.

Começo por agradecer ao meu orientador, o Professor Doutor Mário Romão por, acima de tudo, o sacrifício, disposição e compreensão perante este trabalho e os conselhos que me deu, e também pelo seu método de ensino no âmbito da unidade curricular Gestão de Projetos Informáticos.

Gostaria também de agradecer ao meu orientador na empresa, Tomé Gil, e toda a equipa com quem tive a oportunidade de trabalhar, que me deram muito apoio, orientação e espírito de equipa, e que proporcionaram que o meu estágio fosse gratificante e uma primeira experiência de trabalho memorável.

Por fim, agradeço à minha família, em especial à minha mãe e ao meu pai, por todos os esforços, por sempre proporcionar-me o melhor, pela força transmitida e pelo amor incondicional.

# ÍNDICE

Lista de abreviaturas .....	i
Resumo.....	ii
Abstract .....	iii
Agradecimentos .....	iv
Índice.....	v
Índice de Figuras .....	vii
1. Introdução .....	1
2. Enquadramento Metodológico.....	2
2. Revisão de literatura.....	4
3. Caso de estudo .....	9
3.1. Apresentação da empresa .....	9
3.2. Estágio curricular .....	11
3.2.1. Principais atividades desenvolvidas .....	13
3.3. Identificação de obstáculos ao crescimento .....	18
3.3.1 Projetos falhados.....	18
3.3.2 Falta de estrutura organizada e documentação.....	20
3.3.3 Uso de aplicações não adequadas para a gestão do projeto .....	21
3.4. Mudança e soluções adotadas .....	22
3.4.1 Agile e DevOps .....	22
3.4.2 Levantamento de requisitos e User Stories .....	24
4. Reflexão das alterações implementadas e relação entre o estágio e formação académica .....	27

5. Conclusão.....	30
5.1. Experiência de Estágio .....	30
5.2. Limitações .....	30
Bibliografia .....	31
Anexos .....	33

## ÍNDICE DE FIGURAS

Figura 1 – Diagrama sobre o faseamento dos trabalhos .....	3
Figura 2 – Processo e fases da framework AGILE Scrum .....	5
Figura 3 – Controlo de WIP com frameworks Kanban .....	6
Figura 4 – Fases da cultura DevOps .....	8
Figura 5 - Cronograma de atividades proposto no Plano de Estágio.....	12
Figura 6 – Principais causas de falha de projetos .....	19
Figura 7 – Fases do modelo Waterfall .....	20
Figura 8 – Agile process Work Items .....	23
Figura 9 – Análise da evolução de conclusão de Work Items .....	33

# 1. INTRODUÇÃO

O presente relatório de estágio, redigido no âmbito do TFM do curso de Gestão de Sistemas de Informação, é realizado após o estágio curricular na empresa Escrita Digital, S.A. Adicionalmente, pretende contribuir para um tema presente na área de sistemas de informação, as metodologias *Agile*, que durante a realização do estágio evidenciaram a sua importância, utilização e efeitos numa organização. Permitiu assim ter um contacto com o tema para além do contacto académico previamente obtido, sendo esta a oportunidade para possivelmente contribuir para o mesmo, de um ponto de vista mais prático e baseado em situações reais. A realização do estágio curricular foi possível sobre a orientação do Professor Doutor Mário Romão, responsável pela cadeira de Gestão de Projetos Informáticos, disciplina que me suscitou interesse em metodologias *Agile* e o seu impacto em organizações, e em planeamento de projetos, e sobre a orientação do coordenador de estágio na empresa, o Diretor-geral Tomé Gil.

Neste caso de estudo, será explorado como uma pequena-média empresa de desenvolvimento de software implementou práticas *Agile* e DevOps para superar os problemas que impossibilitavam o seu funcionamento ideal. Será feita uma análise dos mesmos e que práticas ou métodos foram utilizados para os superar.

O objetivo deste trabalho é, ao compreender a experiência desta organização, ganhar uma perspetiva de como estas metodologias e culturas podem ser eficazes, em situações comuns na área de desenvolvimento de software, e otimizar os processos de uma organização.

Será feita inicialmente uma revisão da literatura sobre estas culturas, *Agile* e DevOps, e uma comparação entre ambas. Será apresentada a empresa, a sua área de operações e uma breve descrição dos produtos mais relevantes, de forma a melhor compreender o contexto da empresa onde foi realizado o estágio. Após esta introdução, serão descritas as atividades realizadas durante o estágio e, de seguida, os obstáculos encontrados na organização da empresa, e como as práticas

evidenciadas ajudaram na otimização e resolução dos obstáculos encontrados. Será também explicitada a articulação entre a formação académica e a aplicação dos conhecimentos nas atividades realizadas no estágio. O capítulo final será constituído por conclusões, limitações e futuras sugestões.

## 2. ENQUADRAMENTO METODOLÓGICO

No seguinte capítulo é descrita a metodologia seguida para a realização do presente relatório. Sendo um estágio curricular, inicialmente foi realizado o levantamento dos problemas mais significativos na organização. Este levantamento foi feito através de contacto com colegas de forma a melhor compreender a história da empresa e a sua organização prévia. Após a identificação destes problemas, foi feito um levantamento das soluções implementadas, ou em implementação, e após uma contextualização e pesquisa sobre as mesmas, foram identificadas as áreas que poderiam usufruir de algum reforço na sua otimização. Foi feita posteriormente a descrição destas soluções, com os processos implementados e softwares utilizados.

Com esta descrição detalhada, comecei a elaborar soluções para as áreas que identifiquei como potenciais áreas de melhoria, nomeadamente o levantamento de requisitos e *User Stories*. Destas soluções, seleccionei as mais adequadas e efetuei uma apresentação ao *CTO*, que mencionou qual a solução mais viável e indicou alterações e melhorias que poderia efetuar à proposta. Com a inclusão destas alterações, efetuei uma nova proposta, mais completa, junto do *CTO* e Diretor-Geral, que aprovaram a mesma, podendo então elaborar o documento para realização das alterações necessárias.

Finalizado todo este processo, prossegui com a apresentação das alterações à organização dos processos a todos os colaboradores da empresa, com esclarecimento de potenciais dúvidas e, de forma a cumprir com a nova

metodologia, efetuei um documento final a explicitar as alterações efetuadas e esclarecimento às dúvidas mais pertinentes levantadas durante a apresentação, de forma a possuir uma documentação concreta destas mudanças.

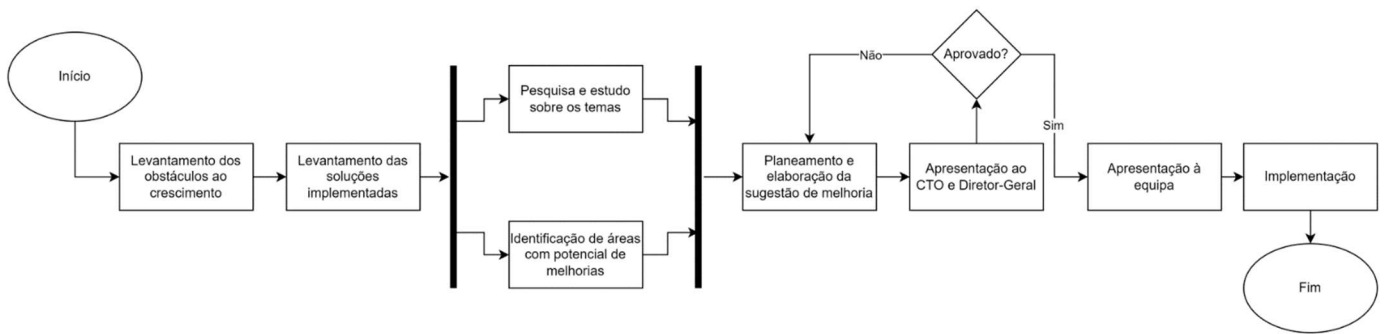


Figura 1 - Diagrama sobre o Faseamento dos Trabalhos

## 2. REVISÃO DE LITERATURA

*Agile* é um termo amplamente mencionado na última década, que descreve a abordagem ao desenvolvimento de software. Promove a entrega incremental, a colaboração de equipas e uma mentalidade de planeamento e aprendizagem contínua, de forma a produzir software de maneira eficaz e de acordo com as necessidades dos stakeholders. Estes métodos ágeis começaram a surgir devido ao facto de os modelos existentes de gestão de projeto, não conseguirem lidar com os cenários de incessantes mudanças rápidas. (Fitzgerald, Rowe, Lyonnet, & Hemon, 2020)

Não é possível efetuar um planeamento perfeito no desenvolvimento de software, devido às ideias que vão surgindo durante o desenvolvimento do mesmo e devido à dificuldade de estimar corretamente o esforço necessário. (Cohn, 2004)

*Agile*, sendo uma mentalidade e cultura que uma organização adota, não possui uma abordagem universal que possa ser utilizada independente da situação organizacional. Cada organização tem as suas necessidades, objetivos e limitações, pelo que surgiram assim as *frameworks* ágeis. Estas descrevem práticas e técnicas que as equipas podem utilizar para implementar uma metodologia *Agile* na sua organização, assim como os princípios importantes. Alguns exemplos de *frameworks* com mais notoriedade são o *Scrum*, *Extreme Programming (XP)* e *Kanban*. Devido à grande variedade de *frameworks*, serão destacadas as duas mais relevantes para este trabalho, o *Scrum* e o *Kanban*.

“O *Scrum* é uma *framework* leve que ajuda pessoas, equipas e organizações a gerar valor através de soluções adotivas para problemas complexos” (Sutherland & Schwaber, 2020).

Funciona com o conceito de sprints – períodos curtos, onde a equipa completa uma certa carga de trabalho definida previamente. É importante destacar que cada sprint é iniciado com o planeamento do mesmo e termina com a sua reflexão, de modo a possibilitar uma melhoria contínua deste processo. Esta organização é

efetuada pelo responsável, o *Scrum Master*, que deve liderar a equipa e assegurar que todos os sprints são corretamente terminados. Também é importante destacar a função de *Product Owner*, que é responsável por definir a visão do produto e priorização do trabalho.

Todo este fluxo, representado de forma simplificada na Figura 2, permite-nos concluir que *Scrum* é uma framework para ajudar as equipas a entregar produtos de alta qualidade, rapidamente e eficientemente, ao incentivar colaboração, transparência e melhorias contínuas nas equipas das organizações.

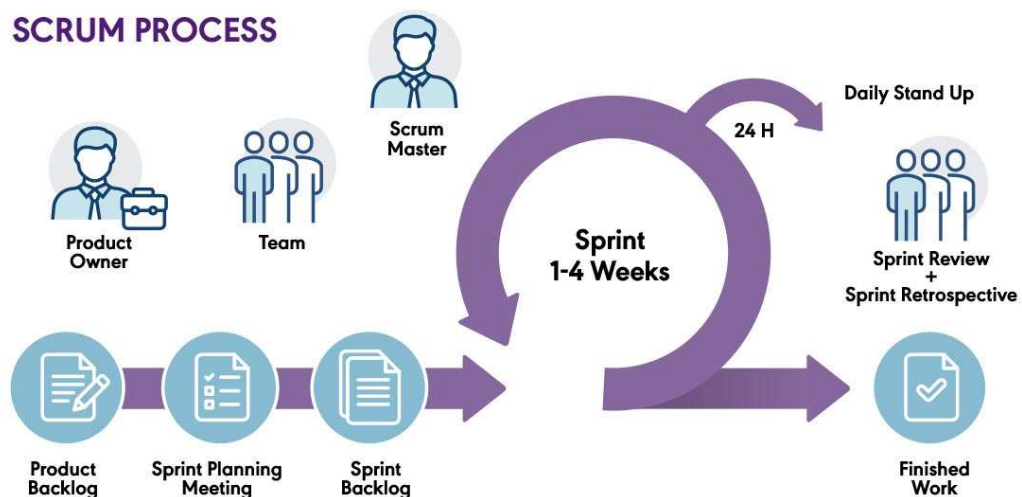


Figura 2 – Processo e fases da framework *AGILE* Scrum. Fonte: Artigo publicado em PM-Partners.com

*Kanban* provém de uma palavra japonesa, que se traduz em *billboard*, e é um mecanismo para ajudar no controlo de fluxo, primeiro utilizado pela Toyota e, apesar de ter sido criado com o intuito de melhorar a eficiência de fabricação, o desenvolvimento de software partilha muitas características com esta área, permitindo assim a utilização desta metodologia. (Ahmad, Markkula, & Oivo, 2013)

Um dos princípios mais relevantes do *Kanban* é a visualização do trabalho com a utilização de *Boards*, e organização do mesmo com a aplicação de prioridades, datas de entrega, e outros instrumentos. Desta forma, as equipas podem visualizar o seu trabalho e tarefas durante todo o processo, facilitando a identificação de obstáculos, restrições e o acompanhamento do progresso. O *Kanban* não define um processo específico ou funções como o *Scrum*, mantendo o foco em ajudar as equipas a melhorar o seu fluxo de trabalho e otimizar os recursos existentes. Também beneficia de um conceito de WIP (Work-In-Progress), aumentando a produtividade da equipa, ao limitar o número máximo de tarefas que a equipa pode realizar simultaneamente, eliminando o excesso de trabalho. Podemos, na Figura 3, visualizar um exemplo de como utilizar um *Kanban Board* simplificado para acompanhar o WIP. Kanban é assim uma framework flexível, adaptável e que ajuda as equipas a melhorar a sua eficiência e rapidez de entrega, otimizando o fluxo de trabalho.

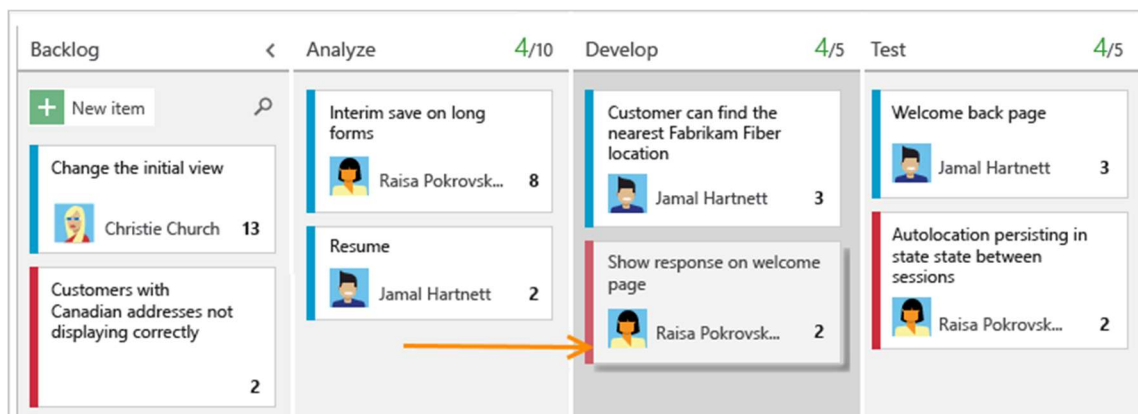


Figura 3 – Controlo de WIP com frameworks Kanban. Fonte: Microsoft Documentation, 2023

Podemos evidenciar que estas metodologias *Agile* são diferentes. O *Scrum* foca-se em sprints fixos, enquanto o *Kanban* é um modelo de fluxos contínuos; um utiliza como métrica a velocidade, enquanto outro utiliza ciclos de tempos. Apesar destas diferenças, estes modelos não são incompatíveis, pelo que muitas

organizações utilizam uma fusão destes dois conceitos, utilizando aspetos de cada um, conforme mais eficaz para a equipa. (Ladas, 2008)

DevOps é a combinação dos termos Desenvolvimento e Operações. Apesar do crescimento de metodologias ágeis, ainda existiam algumas questões que impediam uma maior redução nos ciclos dos produtos, nomeadamente na interação entre as fases de desenvolvimento e de produção, e das suas respetivas equipas. As equipas de desenvolvimento focavam-se nas entregas dentro dos prazos e orçamentos definidos, frequentemente sacrificando estabilidade e manutenção, enquanto as equipas de operações, que são avaliadas nestes dois pontos, ficavam desconfortáveis com cada lançamento, atrasando o processo devido à redução de colaboração entre as duas (Dörnenburg, 2018).

Esta cultura tem assim como objetivo reduzir o lapso entre o Desenvolvimento e Operações, ao otimizar a qualidade, velocidade e confiança das entregas de software. Para atingir o seu objetivo tem como foco automatizar e agilizar o processo de construção, testagem e implementação, através da criação de equipas multifuncionais, permitindo responder melhor às necessidades do cliente e manter uma mentalidade de contínua entrega e desenvolvimento, assegurando a qualidade dos produtos (Fig. 4). Para atingir este objetivo, utiliza combinações de processos, práticas e ferramentas como:

- *Continuous integration (CI)* – A equipa de desenvolvimento integra o seu código num repositório partilhado com frequência, permitindo a mesma detetar e corrigir erros mais rapidamente;
- *Continuous Delivery (CD)* – As equipas automatizam os processos para poderem fazer releases a qualquer momento;

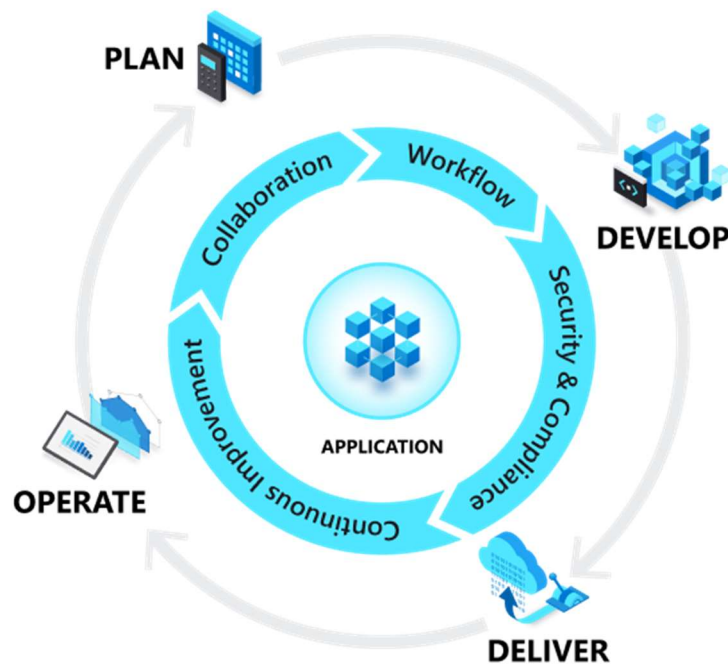


Figura 4 – Fases da cultura DevOps. Fonte: Microsoft Documentation, 2022

Com base nestes conceitos podemos concluir que apesar das semelhanças entre estas duas culturas, DevOps e *Agile* têm diferentes objetivos e áreas de foco. O foco de *Agile* é o desenvolvimento de software, e realça a importância de iterações curtas e rápidas e a satisfação do cliente. DevOps tem como foco a entrega de software, do desenvolvimento à entrega e operação, realçando a importância da velocidade, qualidade e segurança do software.

No entanto, estes dois conceitos são utilizados em conjunto para melhorar todo o processo de desenvolvimento e entrega de projetos, podendo assim serem consideradas metodologias complementares em muitos casos, se usadas corretamente. É comum em DevOps, utilizar as frameworks *Scrum* e *Kanban*, para gerir o desenvolvimento de software e poder visualizar e otimizar o workflow de trabalho para as fases de desenvolvimento, teste e entrega. (Mohammad, 2017)

### 3. CASO DE ESTUDO

#### *3.1. Apresentação da empresa*

A empresa que será utilizada como caso de estudo e onde foi efetuado o estágio para a realização deste relatório, denominada de Escrita Digital, é uma empresa de desenvolvimento de produtos de software para as organizações. Fundada em 2001 com este objetivo, durante os seus primeiros anos focou-se no desenvolvimento de aplicações e software á medida – desenvolvimento personalizado para um determinado cliente com necessidades muito específicas. Em 2005, com o conhecimento adquirido nos anos precedentes, criou o seu primeiro produto, o **Ed-Portal**, um gestor de conteúdos para sites web. Porém, decidiu que este não era o caminho que queria continuar devido á volatilidade do mercado e assim, com toda a experiência adquirida com estes anos no mercado, criou a família de produtos sobre a marca **XRP**, que ainda hoje constitui o centro de atividade da empresa, tendo ao longo dos anos sofrido bastantes alterações e melhorias.

Na sua criação, as aplicações **XRP** foram concebidas com a ideia de serem um conjunto de ferramentas de software vocacionadas para a gestão dos processos de negócios comuns a muitas organizações. Este conceito tem como principal finalidade permitir que os colaboradores possam, em regime de self-service, efetuar diversas tarefas que lhe são requeridas, evitando, assim tarefas de valor acrescentado e minimizando a burocracia nas organizações. São assim ferramentas que permitem aumentar a produtividade, reduzir custos e aumentar a coesão organizacional. Uma grande vantagem das aplicações **XRP** que merece ser destacada é o facto de permitirem a possibilidade de licenciamento por módulos. Isto significa que é possível adquirir o produto, e separadamente adquirir módulos disponíveis que trazem valor acrescentado às aplicações. Outra vantagem é o facto de a empresa permitir que as aplicações possam ser utilizadas em cloud ou a possibilidade de instalação nos servidores dos clientes.

Atualmente, na família **XRP** existem duas aplicações que se destacam, a aplicação de gestão de frotas, que assegura todos os procedimentos associados à gestão de viaturas e parques de máquinas - **XRP Frotas** – e a aplicação de gestão de processos de negócio ligado à gestão de recursos humanos – **XRP RH**. Sendo uma família de aplicações, existe uma série de características e funcionalidades comuns, sendo de destacar: Interface em browser; Solução Multilinguística; Ecrãs intuitivos; Programa de notificações e envio de e-mails automáticos; Fácil integração com outras aplicações de ERP.

Devido ao foco atual da empresa na aplicação de gestão de frotas, avaliada como a aplicação mais promissora e cujas funcionalidades se destacam em relação aos competidores, esta será brevemente apresentada.

### *XRP Frotas*

A aplicação tem como módulo principal a gestão de frotas e máquinas, cujo objetivo é possibilitar aos clientes a gestão da frota automóvel e parque de máquinas da sua organização, os respetivos contratos, a manutenção preventiva e corretiva, os custos associados e as obrigações legais. Disponibiliza a cada utilizador o acesso à informação relevante dos bens que lhe estão delegados, bem como executar ações nos momentos adequados.

É importante destacar também os principais benefícios e funcionalidades da aplicação:

- Centralizar numa única base de dados toda a informação necessária para uma gestão correta de frotas;
- Assegurar contabilização automática da faturação com integração nos principais ERP's;
- Assegurar o cumprimento das obrigações legais, como por exemplo, as inspeções obrigatórios, impostos e seguros e documentação;
- Controlo de utilização e registos da utilização dos veículos;

- Dashboards configuráveis para representações visuais da gestão de frotas, para uma melhor compreensão da situação atual da frota da empresa;
- Integração com sistemas de GPS;
- Integração com ligações ao veículo para receção de eventos (Ex: travagens bruscas);
- Integração com outros equipamentos como sensores de pressão de pneus e boias de depósito autónomas.

### *Módulos adicionais*

Como anteriormente referido, umas das vantagens da aplicação é a possibilidade de integração com os módulos opcionais, sendo de destacar o módulo de gestão partilhada (Pool) e módulo de gestão de oficinas internas.

O módulo **XRP Pool**, é um módulo que tem atraído muita adesão e interesse por parte dos clientes e é considerado um módulo promissor pela empresa. É um módulo que permite a utilização partilhada de viaturas e sistemas de rent-a-car, cabendo à aplicação aferir todos o processo de requisição e afetação de viaturas, permitindo também a gestão de boleias. É um processo de requisição simples e intuitivo, cuja utilização aumenta significativamente a taxa de utilização dos veículos, permitindo executar serviços com menos veículos e reduzir custos.

## *3.2. Estágio curricular*

Efetuada a introdução da empresa, descreverei as atividades que realizei durante o estágio, que tinha como objetivo a preparação e formação para exercer funções de *Product Manager* e *Product Owner*, com ênfase no levantamento de requisitos, testes e documentação.

Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Formação inicial														
Produtos xrp.pt	X													
DevOps	X													
XRP Produtos	X													
Levantamento de requisitos														
Discussão interna	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ouvir clientes				X	X	X	X	X	X	X	X	X	X	X
Documentação				X	X	X	X	X	X	X				
Esboços de ecrãs				X	X	X	X	X						
Testes e documentação														
Testes	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Documentação	X	X	X	X	X	X	X		X					
Produção														
Testes												X	X	X
Comunicação a clientes												X	X	X
Documentação												X	X	X

Figura 5 - Cronograma de atividades proposto no Plano de Estágio.

O estágio teve a duração de 3 meses, iniciando-se a 26 de setembro de 2022 e terminando a 30 de dezembro de 2022, tendo o planeamento proposto na figura 5.

Durante as primeiras semanas de estágio, todas as atividades realizadas tiveram como principal foco a formação, compreensão dos procedimentos da empresa, aprofundamento do conhecimento das funcionalidades das aplicações, integração na empresa e incentivo ao trabalho em equipa. Nesta primeira fase todo o trabalho foi realizado presencialmente nas instalações físicas da empresa, nas Amoreiras, de forma a facilitar este processo de formação e integração, e de forma a fornecer os materiais necessários como o computador portátil e toda a sua preparação, incluindo a instalação dos programas necessários.

Este processo de formação inicial, começou pelo entendimento dos softwares utilizados no dia-a-dia na empresa, nomeadamente os produtos **XRP** utilizados pelos clientes, os serviços Azure DevOps utilizados pela equipa para o registo e

organização do trabalho de desenvolvimento e a ferramenta interna **XRP Produtos**, para o registo de pedidos de clientes, gestão de contratos e faturação. Uma atividade de grande importância que começou nesta fase inicial e continuou ao longo do estágio, foi a realização de testes aos desenvolvimentos efetuados e a documentação dos mesmos.

Após esta fase inicial de adaptação e formação, comecei a efetuar trabalhos mais complexos como o levantamento de requisitos, idealização e apresentação de novas ideias, esboços de ecrãs na aplicação, reuniões e comunicação com os clientes. Durante esta fase, também foi evidenciado que a empresa tinha recentemente sofrido alterações na sua forma de operar, pelo que ainda estava a solidificar os seus métodos de operação. Iniciei uma análise ao funcionamento anterior da empresa e as suas limitações, as soluções adotadas para a resolução dos mesmos e planeamento de abordagens para otimizar os processos que ainda poderiam sofrer de melhorias, como a área de levantamento de requisitos e documentação.

### *3.2.1. Principais atividades desenvolvidas*

#### *3.2.1.1 Testes e documentação*

A primeira atividade realizada durante o estágio, consistiu em efetuar testes a funcionalidades. O objetivo inicial desta atividade era o aprofundamento do conhecimento sobre a aplicação e as suas funcionalidades, de forma a proporcionar uma visão abrangente da aplicação, que seria fundamental para desempenhar efetivamente as atividades futuras de levantamento de requisitos, contacto com o cliente e desenvolvimento de novas funcionalidades. Os testes foram conduzidos em diferentes estágios dos projetos, desde a fase de desenvolvimento até à fase final de validação.

Ao longo dos testes, sempre que possível e necessário, recorria à documentação interna da empresa, e caso a mesma não existisse ou se encontrasse carente, era

encarregado de elaborar documentos detalhados das funcionalidades, incluindo manuais para apoio ao cliente. Deste modo, não só era aprofundado o conhecimento, como contribuía para um registo das funcionalidades atualizado, uma parte muito importante no desenvolvimento de software.

Nesta fase foram realizados os seguintes tipos de testes:

- i. **Testes de familiarização:** Exploração de todas as funcionalidades relevantes existentes da aplicação, permitindo obter uma visão geral da mesma e entender como as diferentes partes se interconectavam. Nas áreas-chave da aplicação, foram efetuados testes com mais frequência, de forma a compreender a lógica das mesmas e identificar possíveis pontos de melhoria.
- ii. **Testes de funcionalidade:** Execução de testes para verificar se as novas funcionalidades desenvolvidas respondiam aos requisitos definidos. Estes testes eram efetuados à medida que as funcionalidades eram desenvolvidas e entravam na fase de testes. Estes testes não se limitavam aos testes de UI, mas incluíam também testes a API's com recurso a ferramentas como o software *Postman*.
- iii. **Testes de usabilidade:** Além dos testes de funcionalidade, também era necessário avaliar a usabilidade das funcionalidades. O facto de não possuir um conhecimento aprofundado da aplicação e familiaridade com a mesma, comparativamente com os restantes *Testers* da equipa, era uma vantagem pois permitia visualizar a aplicação do ponto de vista do utilizador. Também eram efetuados testes com utilizadores reais da aplicação, recebendo *feedback* sobre a interface e navegabilidade da mesma, o que potenciou identificar possíveis melhorias e aprimorar a experiência do utilizador.

- iv. **Testes de desempenho:** Em adição aos testes de UI acima mencionados, foram delineados testes de carga e stress que permitiam avaliar como a aplicação se comportava sobre condições de uso intenso e identificar possíveis problemas de desempenho.

### *3.2.1.2 Levantamento de requisitos*

Durante o estágio, uma das responsabilidades que me foi atribuída foi o levantamento de requisitos para diversos projetos. Estas atividades consistiam em compreender as necessidades e os objetivos dos clientes, a fim de desenvolver soluções eficazes. Para o desempenho deste trabalho, geralmente realizava as seguintes etapas:

#### *3.2.1.2.1 Identificação dos stakeholders*

Fase inicial de identificação dos principais stakeholders envolvidos no projeto. Efetuei reuniões com os clientes e com os membros da equipa para compreender as expectativas e necessidades específicas. Por vezes, as necessidades partiam de melhorias que identificava ao realizar as atividades de testes.

#### *3.2.1.2.2 Análise e documentação dos requisitos*

Após a identificação das expectativas e necessidades, procedia com a análise detalhada dos requisitos dos projetos. Por norma, esta análise era partia do levantamento das necessidades dos stakeholders, seguida de reuniões com membros da equipa para discussão destes requerimentos. Como parte do levantamento de requisitos, elaborava uma documentação clara e concisa, descrevendo todos os requisitos identificados, que servia como guia para o desenvolvimento posterior da solução.

### 3.2.1.2.3 Idealização e esboço de ecrãs

Feito o levantamento de requisitos e a sua documentação, a fase seguinte seria a idealização de soluções dentro das aplicações da empresa. Idealizava e anotava soluções possíveis, e após este período, conciliava as minhas ideias com as ideias restantes da equipa, de forma a elaborar a melhor solução possível, de acordo com todas as necessidades registadas. Nesta fase também existia um grande foco em elaborar soluções que não fossem específicas para o cliente em questão, ou seja, que pudessem ser úteis para os restantes utilizadores das aplicações, e que fossem desenvolvidas de forma a permitir futuras alterações e melhorias.

### 3.2.1.3 *Contacto com clientes*

Durante o estágio, tive a oportunidade de interagir diretamente com clientes, fornecendo suporte e atendendo às suas necessidades. Esta experiência permitiu-me desenvolver habilidades de comunicação e aprender a lidar com situações de atendimento ao cliente. As principais atividades realizadas destacam-se em duas vertentes.

A primeira consistiu no que se pode denominar de suporte técnico. Com o conhecimento abrangente da aplicação, pude fornecer suporte aos clientes. Isto envolveu responder a eventuais dúvidas, solucionar problemas e auxiliá-los na utilização eficaz da aplicação. A minha compreensão detalhada das funcionalidades permitiu oferecer soluções rápidas e eficientes para as questões que surgiam, incluindo a entrega de manuais para esclarecimento.

A segunda vertente de contacto com os clientes, provém da demonstração de funcionalidades, com o acompanhamento do coordenador da empresa. Ao interagir com os clientes, utilizei o meu conhecimento das funcionalidades da aplicação para realizar demonstrações e apresentar as vantagens e benefícios do produto. Estas demonstrações ajudaram a fortalecer o relacionamento com os clientes, fornecendo

uma visão clara de como a aplicação poderia atender às suas necessidades específicas. Realizei sessões de treino e testes na aplicação, produzi documentação e forneci assistência personalizada para garantir que os clientes tivessem o máximo aproveitamento da aplicação. Estas demonstrações foram efetuadas tanto a potenciais clientes da aplicação, como a clientes atuais, que ainda não usufruíam de certas funcionalidades.

Durante o estágio também coorganizei uma apresentação de um seminário, na qual fiz uma demonstração das novas funcionalidades do módulo **XRP Pool** desenvolvidas durante o período do estágio, transmitida por Teams.

### *3.3. Identificação de obstáculos ao crescimento*

A empresa ao longo da sua atividade conseguiu adquirir clientes importantes que possibilitaram a continuação das suas operações e aplicações, e também a criação de novas funcionalidades e melhorias, assim como a comercialização da família **XRP**. No entanto neste período, apesar do crescimento positivo, foi-se evidenciado a existência de alguns obstáculos que limitavam o crescimento da empresa e que impossibilitavam a mesma de atingir o potencial que todos reconheciam existir. Existia uma combinação de fatores que não permitia a empresa atingir o potencial projetado, e cuja solução não seria linear e de rápida implementação, sendo assim adiado todo o processo de resolução destes problemas.

De forma a melhor perceber a situação da empresa, foi efetuada uma análise qualitativa, através de entrevistas semiestruturadas, com um carácter informal, de forma a contextualizar a situação da empresa mais realisticamente. Durante as entrevistas foram realçados vários problemas, que podem ser agrupados em três áreas: Projetos falhados; falta de estrutura organizada; utilização de aplicações não adequadas nos processos.

#### *3.3.1 Projetos falhados*

Um dos principais indicadores da existência de fragilidades nos métodos da empresa, e que provinha da combinação de vários fatores, era o facto de existir uma percentagem notável de falha de projetos. Um projeto falhado neste caso, não é um projeto que era antecipada e erradamente terminado, mas no qual algumas instâncias poderiam ser consideradas como falha de projeto. Na figura 6, encontram-se as principais causas de falha de projetos.



Figura 6 – Principais causas de falha de projetos. Fonte: Kashyap, 2022

Um dos problemas que gerava esta falha nos projetos era o incumprimento de prazos de entregas, devido a um conjunto de fatores.

Por vezes o produto não se encontrava de acordo com o pedido do cliente, devido a falhas no planeamento e documentação, o que impunha um esforço adicional à equipa e uma extensão do prazo de entrega. Outro cenário consistia num projeto passar à fase de testes, e quando era feita a avaliação final, este não cumpria todos os requisitos do mesmo, sendo necessárias alterações. Frequentemente, a causa era uma falha na fase inicial de planeamento do projeto e levantamento de requisitos, mas por vezes também devido à falta de comunicação e documentação.

Uma gestão de tempo incorreta, de definição de esforço necessário e prazos de entrega também pode contribuir para uma taxa elevada de falhas nos projetos. Quando não é considerado o esforço que realmente será necessário, o esforço utilizado em projetos paralelos e quando não são feitas estimativas com *slack* – tempo que uma tarefa pode ser adiada até começar a impedir o desenvolvimento de outras tarefas, pode-se aumentar a taxa de falhas. Na empresa em estudo, as estimativas de projeto eram rígidas, por vezes sem ter em conta a possibilidade de atrasos, refletindo uma gestão e planeamento em falta.

### 3.3.2 Falta de estrutura organizada e documentação

A empresa regia-se por um modelo de desenvolvimento de software que carecia de alguma flexibilidade, e que originava obstáculos. Apesar de não existir uma formalização do modelo utilizado, o mais indicado seria comparar a sua metodologia com variantes do modelo *Waterfall* (Fig. 7). Este modelo, que na sua versão mais simples é constituído por 5 fases: Análise, Conceção, Implementação, Testes e Produção, com a possibilidade de retroceder a uma fase anterior, não seria o mais indicado para a empresa em questão, pois não refletia a realidade dos seus processos, impossibilitando o crescimento da mesma.

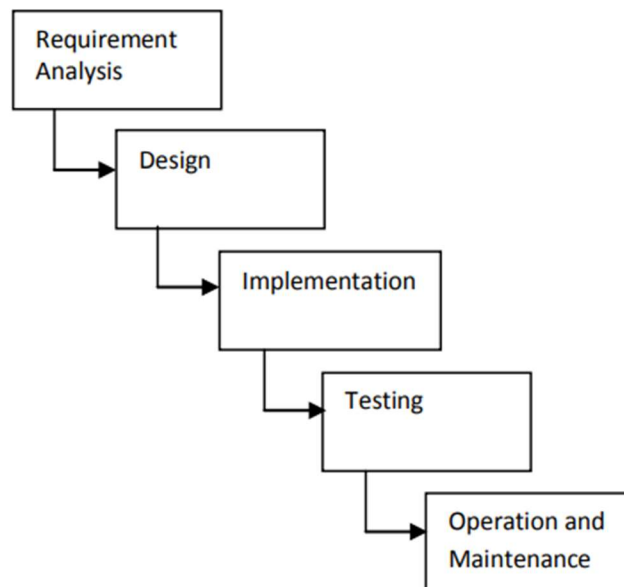


Figura 7 – Fases do modelo *Waterfall*. Fonte: Adenowo & Adenowo, 2013

Existia uma dificuldade durante o processo de criação, de retroceder e efetuar alterações devido ao facto de se planear o sistema na sua totalidade, o que obrigava a que todos os utilizadores conhecessem todas as necessidades do projeto.

Devido também à falta de métodos na equipa e de uma estrutura organizada, existiam também algumas falhas na análise de requisitos e planeamento do projeto e uma conseqüente falta de documentação. Isto originava problemas como a perda

de conhecimento, no sentido em que as funcionalidades criadas por um conjunto de membros da equipa, apenas eram entendidas por esses elementos. Futuramente, e caso fosse necessário fazer alguma alteração ou melhoria nessas funcionalidades, se as pessoas envolvidas originalmente não participassem nestas novas alterações, existiria um grande esforço de forma a entender os procedimentos que foram utilizados e como foi executada a funcionalidade.

Devido a esta falta de estrutura e procedimentos, também existia uma dificuldade em realizar testes contínuos e otimizados, sendo frequentemente realizados perto dos prazos de entrega, o que exercia pressão adicional sobre toda a equipa.

### *3.3.3 Uso de aplicações não adequadas para a gestão do projeto*

Um dos obstáculos que gerava grande desconforto às equipas de projeto, devia-se a toda a organização do mesmo, desde a sua criação, à descrição das suas funcionalidades, gestão de prazos e até mesmo testes, serem feitos em Excel. Apesar de ser uma ótima ferramenta para análise, não é a aplicação ideal para ajudar na gestão de desenvolvimento de software.

Existiam números ficheiros Excel para as várias aplicações, não estando os processos centralizados e onde não era possível um grande nível de detalhe e organização. Mesmo em fase de testes, era uma aplicação rudimentar e que a equipa constantemente detetava as suas falhas e falta de funcionalidades, o que atrasava os projetos e gerava alguma falta de rigor e motivação. Em termos de colaboração, carecia de funcionalidades que permitissem à equipa de colaborar e coordenar os seus esforços e, em termos de segurança, a corrupção de ficheiros era uma preocupação constante.

### 3.4. Mudança e soluções adotadas

Após a identificação destes obstáculos, foi feito um planeamento de mudanças e melhorias, de forma a otimizar os processos da organização. Inicialmente foram adotados métodos *Agile* e DevOps na equipa, com a formação das equipas de forma adequada e de seguida efetuada a implementação de ferramentas como o Azure DevOps.

#### 3.4.1 Agile e DevOps

Com o foco principal das metodologias *Agile* em consideração, a utilização correta dos recursos, eliminação de desperdícios, redução de custos e diminuição da complexidade das tarefas, foi feito um planeamento para adotar uma combinação de duas *frameworks* na organização, *Scrum* e *Kanban*.

##### 3.4.1.1 SCRUM

Com esta implementação, a gestão dos projetos começou a ser realizada em sprints – períodos de 4 semanas, onde a equipa completa uma carga de trabalho previamente definida. É realizada uma *Sprint meeting* onde é definido o trabalho a realizar no sprint seguinte, e é organizado o plano para concretizar o objetivo do mesmo. Para assegurar o bom funcionamento, são efetuadas reuniões diárias (conhecidas como *daily Scrum*), onde cada membro da equipa reporta o seu progresso de trabalho. Estas reuniões são importantes para prevenir qualquer bloqueio desafiador de atrasar o sprint definido, e incentiva uma maior colaboração entre os membros. Após a conclusão do sprint, a equipa reúne para refletir o trabalho realizado, e as melhorias que poderiam ser implementadas futuramente. Com base no *Framework Scrum* foi introduzido o conceito de *Work Items*, lista de tarefas e atividades a serem realizadas para a conclusão de um projeto, sendo estes divididos em *Epics*, *Features* e *User Stories*. (Fig. 8)

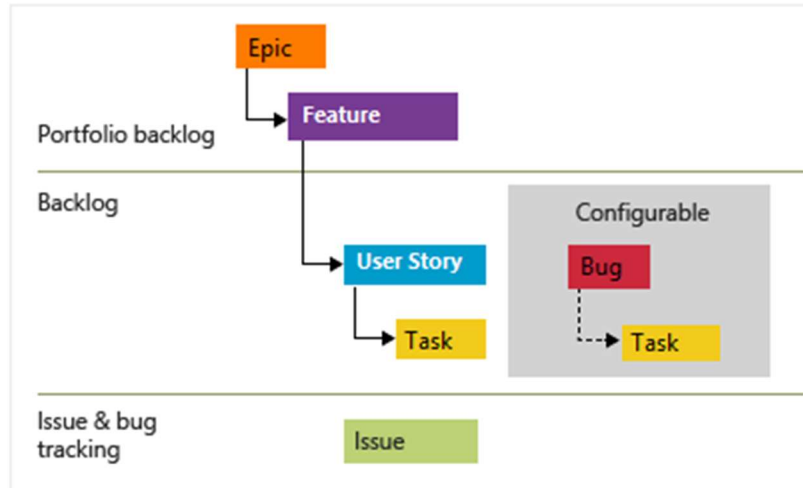


Figura 8 – Agile process Work Items. Fonte: Microsoft Documentation, 2023

*Epics* são utilizados para o agrupamento de todo o trabalho de um certo projeto, podendo ser considerados como o projeto em si e que não são concluídos num Sprint; As *Features* são o trabalho a ser realizado dentro das *Epics*, ou seja, são entregáveis baseados nas necessidades dos clientes e stakeholders, que ao serem concluídos adicionam valor à *Epic* em que estão inseridas; As *User Stories* são criadas através da decomposição das *Features*, de forma a poderem ser completadas num sprint e que descrevem os processos necessários para o desenvolvimento.

Geralmente as *User Stories* são sequenciais, mas existe a possibilidade de serem paralelas, podendo assim serem desenvolvidas ao mesmo tempo por *developers* diferentes. É assim possível agilizar o processo de gestão do produto e testes, ao permitir que estes sejam efetuados ao longo da criação do produto, devido à granularidade das tarefas.

#### 3.4.1.2 Azure DevOps

Face à implementação das culturas mencionadas, foi necessário a substituição do Excel por uma ferramenta que suportasse todo este processo da organização. O Azure DevOps é uma coleção de ferramentas e serviços da Microsoft que permite suportar esta nova cultura e métodos da empresa, nomeadamente a colaboração,

gestão e visualização do trabalho, integração com outras ferramentas e sistemas e os conceitos de CI/CD.

Esta aplicação fornece um conjunto de ferramentas e serviços para efetuar a gestão de projetos, através dos seus serviços principais:

- *Dashboard* – *Dashboards* configuráveis sobre os projetos e tarefas;
- *Wiki* – Ferramenta de publicação de texto online;
- *Boards* – ferramenta de suporte ao planeamento e gestão de trabalho;
- *Pipelines* – Ferramenta para automatização de tarefas;
- *Test plans* – Ferramenta para criação e gestão de testes;

O serviço mais importante na empresa é o Azure DevOps Boards, a ferramenta principal para a gestão dos projetos de software, que suporta a cultura de processos *Agile*, *Scrum* e *Kanban*. Com o *Kanban Boards*, é possível fazer a monitorização do trabalho e das tarefas de cada projeto, tudo centralizado numa só aplicação. É possível assim todas as equipas comunicarem de forma rápida, ao possuir um método visual que permite analisar o estado de cada sprint. Com o uso de *Work Items* e a possibilidade de efetuar discussões dentro dos mesmos, o processo de desenvolvimento e teste de funcionalidades é otimizado, e com o uso de gráficos customizáveis, é efetuada a monitorização do progresso de cada projeto de forma visual e intuitiva. (Microsoft Documentation, 2023)

Como a aplicação permite fazer a gestão do trabalho individual, com o uso de filtros e *Queries* que potenciam a gestão personalizada, possibilita o utilizador efetuar toda a gestão do processo de desenvolvimento de software.

### 3.4.2 Levantamento de requisitos e User Stories

Após a implementação de culturas *Agile* e de DevOps e das respetivas ferramentas para otimizar esta integração, foi necessária uma delineação do

levantamento de requisitos e *User Stories*. A empresa encontra-se atualmente com foco neste tópico sendo o objetivo melhorar como os requisitos de cada cliente são descritos, decompostos e transmitidos aos *Developers*. Após a coleção de requisitos, é necessário o planeamento de criação de funcionalidades que satisfarão todos os mesmos, sendo posteriormente discutido com a equipa de projeto o caminho mais adequado. Esta fase é atualmente a mais duradoura, devido à variedade de implementações possíveis na aplicação, pelo que é normalmente realizada com maior antecedência.

Após esta deliberação, é criada a *Feature* que traduz o trabalho que será desenvolvido e conseqüentemente decomposta em *User Stories*, que detalham as tarefas necessárias, sendo atribuídas ao *Developer* mais adequado. Todo este trabalho tem como propósito adicional a documentação das funcionalidades, não só para a organização interna, como para facilitar a criação de documentação de apoio para os clientes. Uma das responsabilidades atribuídas durante o estágio, era garantir que esta organização e fluxo eram mantidos, de forma constante, para toda a equipa ter um plano claro da gestão do produto.

Durante o estágio sugeri um conjunto de melhorias, algumas que se encontram presentemente implementadas na empresa. A primeira sugestão consiste na criação de um *Work Item* prévio à criação de *Features*, que tem como objetivo servir de recipiente de ideias. Deste modo, todas as ideias que surjam são criadas previamente e apenas quando são corretamente detalhas, idealizadas e com ecrãs esboçados, é que são avaliadas e transformadas como *Features*, dentro de um Sprint. Desta forma é possível distinguir as ideias concebidas na sua totalidade, das ideias que ainda requerem alguma documentação. Isto possibilita uma gestão interna mais prática, ao apenas apresentar à equipa de desenvolvimento de Software as *Features* detalhas, enquanto a equipa de produto pode manter um registo atualizado de todas as melhorias futuras para a aplicação. A implementação desta sugestão permitiu assim efetuar um levantamento de requisitos mais

organizado, assim como um processo de documentação definido e que não permite que funcionalidades desenvolvidas não possuam qualquer registro.

A segunda sugestão que apresentei brevemente foi relativamente ao automatismo de testes, usando o Azure DevOps e o software *Postman*. Este processo de automatização poderia otimizar alguns processos na empresa, porém não seria viável, face à recente mudança na empresa e preocupação em implementar corretamente todas as alterações à organização dos projetos, sendo considerado que o seu enquadramento ficaria num futuro um pouco mais distante.

#### 4. REFLEXÃO DAS ALTERAÇÕES IMPLEMENTADAS E RELAÇÃO ENTRE O ESTÁGIO E FORMAÇÃO ACADÊMICA

O estágio curricular descrito neste relatório, permitiu efetuar uma reflexão do confronto entre a formação académica obtida durante o curso de mestrado e a sua aplicação no trabalho na empresa. Como pontos que merecem destaque, posso realçar a aprendizagem e conhecimentos obtidos nas unidades curriculares de Gestão de Projetos Informáticos, Análise de Sistemas de Informação, Gestão de Dados e Bases de Dados e Programação e Ciência de dados.

Com o conhecimento obtido através da unidade curricular de Gestão de Projetos Informáticos, foi possível uma compreensão e análise da estrutura atual da empresa e dos seus projetos. Também me proporcionou ferramentas e métodos de como iniciar, planear e acompanhar projetos e a sua execução.

Esta consolidação de conhecimentos, juntamente com os conhecimentos de outras unidades curriculares, preparou-me para as minhas funções, com ênfase no levantamento de requisitos e idealização de soluções, podendo destacar a aprendizagem de Criação de User Stories, esquemas relacionais e UML, e bases de desenvolvimento de software e manipulação de dados.

Durante o processo de mudança da organização, foi possível observar uma baixa resistência durante as implementações de novas metodologias, ferramentas e processos. Embora a resistência à mudança possa ser um fenómeno em transições organizacionais, a limitada oposição observada é indicativa da necessidade que existia de mudança, e de uma cultura de trabalho adaptável.

Esta receptividade pode ser atribuída a vários fatores. Primeiro, ao facto de a empresa ser de pequena dimensão, possibilitando facilmente uma comunicação eficaz e transparência ao longo da transição, e objetivos claros entre todos os membros das equipas. Além disso, os esforços proativos em envolver os membros das equipas no processo de decisão capacitou os funcionários, fomentando um sentido de *ownership* nas mudanças em andamento. Foram também realizadas

sessões de formação para familiarizar a equipa com as novas metodologias, abordando quaisquer preocupações ou incertezas que pudessem surgir.

O compromisso da liderança da organização em fornecer os recursos e o suporte necessário para a transição, desempenhou um papel crucial na minimização da resistência. Ao investir em programas de formação, atualizar a infraestrutura e oferecer orientação contínua, a empresa demonstrou um compromisso com o sucesso da implementação de Agile e DevOps.

Com a alta receptividade às alterações descritas no capítulo anterior, a empresa utilizada como caso de estudo, conseguiu ultrapassar obstáculos que impossibilitavam a mesma de atingir o seu potencial. Não só houve uma maior eficácia nos projetos e uma maior satisfação e ambição na equipa, como um aumento da satisfação do cliente devido às entregas frequentes, atempadas e com qualidade.

Uma das métricas possíveis de utilizar para evidenciar esta evolução, foi a contagem de *User Stories* e *Features* planeadas, e adicionalmente a comparação com as concluídas, durante o trimestre anterior à implementação, face ao trimestre após implementação (Anexo I).

Como refletido no gráfico, existiu um aumento do número de *User Stories* e *Features* planeadas no quarto trimestre relativamente ao trimestre anterior, assim como, um aumento da percentagem de conclusão destes mesmos *Work Items* face ao planeamento inicial. Esta evolução evidencia uma melhor organização e processos definidos, desde o levantamento de requisitos à fase final, e reforça a produtividade geral das equipas ao refletir que, após a implementação, existiu não só um crescimento das tarefas planeadas como da sua percentagem de conclusão dentro do prazo estabelecido.

Também foi possível para a organização evidenciar esta evolução positiva, devido aos resultados de questionários de avaliação de fornecedores, que alguns clientes enviam semestralmente à organização, visto esta ser fornecedora de um

software. Estes questionários evidenciavam, que face aos resultados obtidos anteriormente, os clientes se encontravam mais satisfeitos com a aplicação e os resultados e frequência das entregas.

Esta mentalidade de melhoria e otimização deverá ser preservada e incentivada, de forma a garantir o crescimento da empresa e a retenção e captação de clientes. Atualmente, como referido anteriormente, a equipa encontra-se a otimizar o levantamento de requisitos e organização do *Board Kanban*, com base nas melhorias sugeridas e expostas no capítulo anterior. Apesar de se observar um progresso positivo, sendo um trabalho contínuo, é necessário manter o rigor e métodos obtidos com estas alterações. Isto permitirá uma uniformização, e um melhor funcionamento nesta fase de desenvolvimento de Software.

Foi possível identificar como área com grande potencial a automatização de muitos aspetos do processo de desenvolvimento de Software e da empresa, e que libertará muito esforço individual da equipa e minimizar erros humanos. Este será o foco futuro, pois a adoção de metodologias *Agile* parte do pressuposto de ser um processo contínuo, pelo que incentiva o constante desenvolvimento de conhecimento, aprendizagem e otimização. Este será um tópico interessante para abordar como trabalho futuro, de forma a perceber como a automatização de tarefas pode aumentar a produtividade da equipa e a segurança dos seus produtos.

## 5. CONCLUSÃO

### 5.1. *Experiência de Estágio*

O estágio realizado na empresa Escrita Digital, foi além das minhas expectativas, e garantiu que não fiquei com qualquer apreensão de entrar no mercado de trabalho. Fui recebido por uma equipa muito acolhedora e que tem uma política de trabalho que espero manter durante a minha experiência profissional. Permitiu-me ganhar conhecimento sobre diferentes aspetos no desenvolvimento de software, algo que como primeira experiência profissional, considero fundamental. Permitiu-me também ganhar algum esclarecimento, que previamente não possuía, sobre o início de carreira profissional que pretendo, que neste caso será o cargo de *Product Owner*. Recebi um grande apoio e motivação por parte da equipa, de forma a tornar este o próximo objetivo da minha experiência profissional.

### 5.2. *Limitações*

Como limitações encontradas durante o trabalho, devido à confidencialidade e regimes de proteção de dados de clientes, não foi possível apresentar dados quantitativos relativos à evolução e crescimento financeiro da empresa, e aumento da retenção de clientes, como também não foi possível descrever detalhadamente as atividades que realizei para os clientes da empresa e algumas funcionalidades para as quais contribuí.

No entanto, é possível assegurar que com a implementação das culturas e ferramentas referidas, face aos obstáculos identificados, a empresa ultrapassou o empasse em que se encontrava e conseguiu um crescimento significativo e também uma grande motivação por parte da equipa, que se traduz num ambiente de trabalho mais produtivo.

## BIBLIOGRAFIA

- Adenowo, A., & Adenowo, B. A. (2013). Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach. *International Journal of Scientific & Engineering Research*, 4(7), 427-434.
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, 9-16.
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc.
- Dörnenburg, E. (2018). The Path to DevOps. *IEEE Software*, 35(5), 71-75.
- Fitzgerald, B., Rowe, F., Lyonnet, B., & Hemon, A. (2020). From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*, 927-945.
- Kashyap, V. (12 de July de 2022). *Top 10 Reasons Why Projects Fail (And How to Solve Them)*. Obtido de Proofhub: <https://www.proofhub.com/articles/reasons-why-projects-fail>
- Ladas, C. (2008). *Scrumban essays on Kanban Systems for Lean Software Development*.
- Microsoft Documentation. (Novembro de 2022). *Plan efficient workloads*. Obtido de Microsoft Azure DevOps Documentation: <https://learn.microsoft.com/en-us/devops/plan/planning-efficient-workloads-with-devops>
- Microsoft Documentation. (Fevereiro de 2023). *Boards & Kanban*. Obtido de Microsoft Azure DevOps Documentation: <https://learn.microsoft.com/en-us/azure/devops/boards/boards/kanban-quickstart?view=azure-devops>
- Microsoft Documentation. (Junho de 2023). *What is Azure Boards?* Obtido de Microsoft Azure DevOps: <https://learn.microsoft.com/en-us/azure/devops/boards/boards/kanban-quickstart?view=azure-devops>

us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops

Microsoft Documentation. (Agosto de 2023). *Work items*. Obtido de Microsoft Azure DevOps Documentation: <https://learn.microsoft.com/en-us/azure/devops/boards/work-items/about-work-items?view=azure-devops&tabs=agile-process>

Mohammad, S. M. (2017). DevOps automation and Agile methodology. *International Journal of Creative Research Thoughts*, 5(3), 946-949.

Sutherland, J., & Schwaber, K. (2020). *The Scrum Guide*. [www.scrum.org](http://www.scrum.org).

*The Agile Journey: A Scrum overview*. (2021). Obtido de pn-partners: <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/>

## ANEXOS

## Anexo I

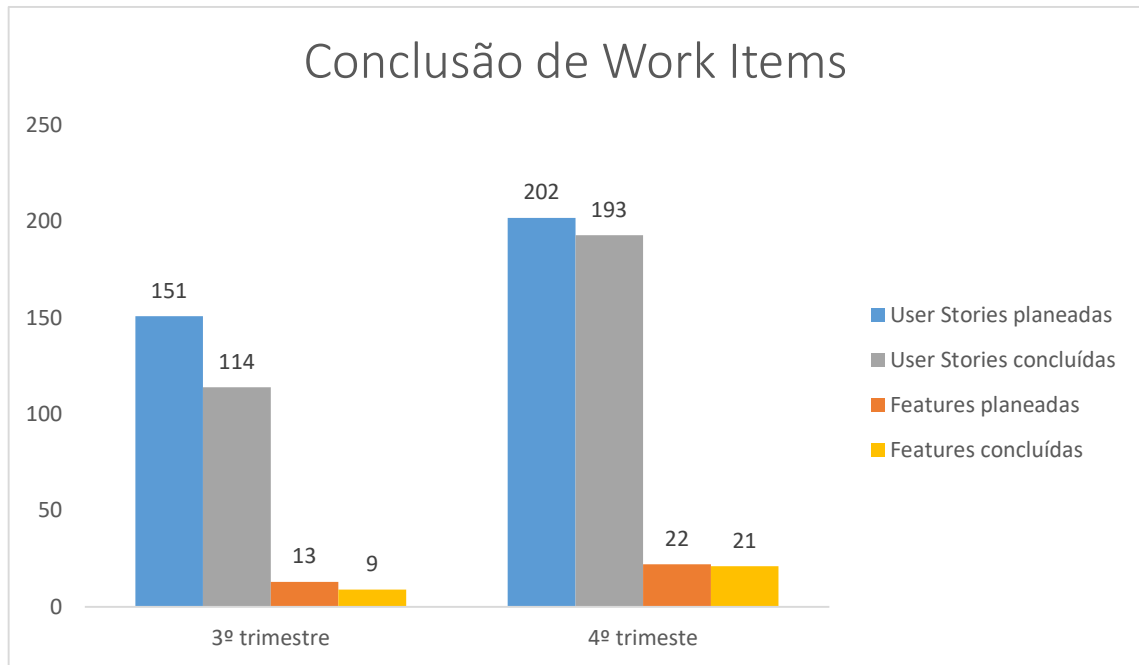


Figura 9 – Análise da evolução de conclusão de Work Items