

UNIVERSIDADE ABERTA



Dissertação de Mestrado

**O *SCRUM* COMO FERRAMENTA DE APOIO PARA O
DESENVOLVIMENTO DE SOFTWARE COM TIMES
DISTRIBUÍDOS**

Mariana Souza (nº2302309)

Mestrado em Gestão

Orientador: Prof. Doutor Pedro Isaías

2026

UNIVERSIDADE ABERTA



Dissertação de Mestrado

**O *SCRUM* COMO FERRAMENTA DE APOIO PARA O
DESENVOLVIMENTO DE SOFTWARE COM TIMES
DISTRIBUÍDOS**

Mariana Souza (nº2302309)

Mestrado em Gestão

Orientador: **Prof. Doutor Pedro Isaías**

2026

RESUMO

Mudanças culturais no cotidiano das empresas e dos trabalhadores vem aumentando gradativamente. Empresas buscam profissionais independentemente do local físico em que se encontram; por sua vez os profissionais buscam empresas que tenham mobilidade e um espaço criativo para que possam realizar o seu trabalho da melhor forma possível. Nesse sentido, este trabalho teve como objetivo descrever os desafios de se trabalhar com times distribuídos, e mostrar como o método ágil *Scrum* facilita a comunicação, neste cenário, garantindo entregas com qualidade. Para isso, foi feito um levantamento bibliográfico sobre o método ágil *Scrum* e as características dos times distribuídos; foi realizado também, um estudo de caso que relata os desafios de desenvolver um software com um time geograficamente distribuído, e como o método *Scrum* com seus processos e valores teve um papel fundamental para o sucesso na entrega do projeto. Após análise do estudo de caso, conclui-se que o método *Scrum* através do seu foco em pessoas se torna uma excelente alternativa para trabalhar com times distribuídos.

PALAVRAS-CHAVE:

Método Ágil *Scrum*, Times distribuídos, Comunicação, Mudança cultural, Gestão de projetos de software

Esta dissertação está redigida em Português do Brasil

ABSTRACT

Cultural changes in the daily life of companies and workers have been increasing gradually. Companies seek professionals regardless of their physical location; in turn the professionals look for companies that have mobility and a creative space so that they can carry out their work in the best possible way. In this sense, this paper aimed to describe the challenges of working with distributed teams, and show how the agile Scrum methodology facilitates communication, in this scenario, ensuring quality deliveries. For this, a bibliographical survey was made on the agile Scrum methodology and the characteristics of the distributed teams; a case study was presented that discusses the challenges of developing software with a geographically distributed team, and how the Scrum methodology with its processes and values played a key role in the successful delivery of the project. After analyzing the case study, we conclude that the Scrum methodology through its focus on people becomes an excellent alternative to work with distributed teams.

KEYWORDS:

Agile Scrum Method, Distributed Teams, Communication, Cultural Change, Software Project Management

SUMÁRIO

1. INTRODUÇÃO	8
1.1 Motivações	9
1.2 Questão e objetivo de investigação	9
1.3 Justificativa	10
1.4 Metodologia do Estudo	10
1.5 Planejamento do Projeto	11
1.6 Estrutura de trabalho	12
2. REVISÃO BIBLIOGRÁFICA	13
2.1 Métodos Ágeis	13
2.2 Scrum	17
2.2.2 Características de um time Scrum	23
2.3 Times Distribuídos	25
2.3.1 História dos Times Distribuídos	25
2.3.2 Desafios na utilização de Times Distribuídos	27
2.3.3 Fatores de sucesso para um DDS	29
2.3.4 Utilização do Scrum em um ambiente DDS	31
2.4 Considerações do Capítulo	35
3. DESENVOLVIMENTO DE SOFTWARE COM SCRUM EM UM TIME DISTRIBUÍDO: UM ESTUDO DE CASO	36
3.1. História da empresa Lean Software	36
3.2 O desenvolvimento de software na empresa Lean Software	37
3.3 Projeto Lista de Pagamentos	39
3.4 Adaptações do Scrum para times distribuídos	41
4. ANÁLISE DE RESULTADOS	45
4.1 Análise do Time Distribuído	45
4.1.1 Características do Time Distribuído	46
4.1.2 Desafios e fatores de sucesso	46
4.2 Análise das Entrevistas	47
4.2.1 Avaliação da eficácia do Scrum	47
4.2.2 Experiência com equipes distribuídas	50
4.2.3 Implementação do Scrum	53
4.2.4 Síntese de Adaptações e Eficácia das Cerimônias Scrum	54

4.2.5 Comunicação, colaboração e cultura.....	56
4.2.6 Perfil e percepções dos participantes.....	61
4.2.7 Síntese interpretativa	65
4.3 Síntese dos Resultados	65
5 CONSIDERAÇÕES FINAIS	69
5.1 Conclusão Geral do Estudo.....	69
5.1.1 Considerações Específicas sobre o Caso da Lean Software	71
5.1.2 Considerações Finais Ampliadas.....	72
5.2 Limitações do Estudo.....	73
5.3 Sugestões para Estudos Futuros	74
REFERÊNCIAS	76
ANEXO A – GUIÃO DA ENTREVISTA	79

Índice de Tabelas

Tabela 1 - Análise do planejamento do projeto	12
Tabela 2 - Fundamentos dos Métodos Ágeis	16
Tabela 3 - Fases do <i>Scrum</i>	19
Tabela 4 - Papéis no <i>Scrum</i>	22
Tabela 5 - Valores do <i>Scrum</i>	23
Tabela 6 - Desafios no desenvolvimento distribuído	27
Tabela 7 - Detalhes dos fatores de sucesso	29
Tabela 8 - Desafios de um projeto DDS	33
Tabela 9 - Fases do <i>Scrum</i> no projeto Lista de Pagamentos	42
Tabela 10 - Característica da equipe	46
Tabela 11 - Desafios e estratégia	46
Tabela 12.1 - Síntese de Experiência em DDS (Participantes A a E)	52
Tabela 12.2 - Síntese de Experiência em DDS (Participantes F a J)	52
Tabela 12.3 - Síntese de Experiência em DDS (Participantes K a O)	53
Tabela 13 - Adaptações do <i>Scrum</i> ao projeto	53
Tabela 14.1 - Adaptação de Cerimônias (Entrevistas A-E)	54
Tabela 14.2 - Adaptação de Cerimônias e Ferramentas (Entrevistas F-J)	55
Tabela 14.3 - Adaptação de Cerimônias e Ferramentas (Entrevistas K-O)	55
Tabela 15.1 - Perspectivas sobre Colaboração e Cultura (Participantes A a E)	60
Tabela 15.2 - Perspectivas sobre Colaboração e Cultura (Participantes F a J)	60
Tabela 15.3 - Perspectivas sobre Colaboração e Cultura (Participantes K a O)	61
Tabela 16 - Síntese dos Pilares do Sucesso do <i>Scrum</i> em DDS na <i>Lean Software</i>	68

Índice de Figura

Figura 1. Gráfico da análise do planejamento	12
Figura 2. Ciclo do <i>Scrum</i>	21
Figura 3. Fatores de sucesso para o DDS	29
Figura 4. <i>Go To Meeting</i> telepresença com todo o time	41
Figura 5. <i>Board</i> de Retrospectiva gostamos e a melhorar	43
Figura 6. Board de Retrospectiva aprendemos e expectativa	44

Lista de abreviaturas

DDS - Desenvolvimento Distribuído de Software
FDD - *Feature Driven Development*
LeSS - *Large-Scale Scrum*
SAFe - *Scaled Agile Framework*
PO – *Product Owner*
ISO - *International Organization for Standardization*
CMMI - *Capability Maturity Model Integration*
SCM - Gerenciamento da Supply Chain
UX – *User Experience*
QA – *Quality Assurance*
DevOps - Desenvolvimento (Dev) e Operações (Ops)
CSM - *Certified ScrumMaste*
PSM - Professional Scrum Master
SM - *Scrum Master*
Dev - *Developer*
XP – *Extreme Programming*

1. INTRODUÇÃO

Este capítulo apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

1.1 Motivações

Nos dias atuais as mudanças culturais no cotidiano das empresas e dos trabalhadores vêm aumentando gradativamente. Empresas buscam por bons profissionais independentemente do local físico em que os mesmos se encontram.

Profissionais buscam empresas que tenham mobilidade e um espaço criativo para que possa exercer o seu trabalho da melhor forma possível.

Ambos os lados, empresas e profissionais têm algumas coisas em comum dentre elas, sendo que a boa comunicação entre times e a fluidez do desenvolvimento de um projeto, está dentre as mais procuradas em termos de ambiente de trabalho e desempenho profissional (Prikladnicki et al, 2002).

Diante desse cenário, torna-se relevante refletir sobre a demanda das empresas por profissionais qualificados distribuídos em diferentes localidades, seja em cidades, estados ou países distintos, e sobre o papel central de uma comunicação eficiente na execução de projetos complexos, assegurando a qualidade e a efetividade das entregas.

1.2 Questão e objetivo de investigação

O Objetivo desta pesquisa é descrever os desafios, de se trabalhar com times distribuídos, e mostrar como é possível realizar entregas de qualidade e ter uma boa comunicação através do uso do *Scrum* como método de desenvolvimento e gestão de projetos.

Com isso, esse trabalho irá relatar um caso real de como pode ser utilizado o método ágil *Scrum* como base para a implementação de um projeto em que o time se encontra de forma distribuída. Será analisado um estudo de caso da empresa *Lean Software* onde todos os seus projetos são realizados com times distribuídos e como, através do método, é possível passar pelos desafios ao se trabalhar nesse cenário.

Neste trabalho será analisada a seguinte questão de investigação: **Como o método ágil *Scrum* pode dar suporte para que times de desenvolvimento consigam entregar projetos de alta complexidade enfrentando os desafios de times distribuídos garantindo boa comunicação e fluidez nas entregas?**

1.3 Justificativa

Segundo Cibotto et al. (2009) devido a problemas na execução, 44% das empresas já tiveram seus projetos cancelados. Uns dos problemas pelos quais os projetos não conseguem obter o crescimento necessário para a sua finalização, é o desenvolvimento com equipes geograficamente distribuídas.

Borges Junior et al. (2011), relata que a utilização de métodos ágeis como o *Scrum* é uma tendência a ser utilizada por organizações para solucionar os problemas enfrentados por times distribuídos, dado que os métodos ágeis têm um foco maior na comunicação entre os membros da equipe e com o próprio cliente.

Executar um projeto com times distribuídos sempre será desafiador, sendo que realizar entregas com alta complexidade com pessoas que não estão no mesmo espaço físico faz com que o projeto se torne ainda mais complexo, levando em consideração a intercomunicação cultural e linguística que pode ser um dos grandes impedimentos na comunicação conforme afirma Cohn (2000).

A crescente necessidade do mercado por sistemas mais complexos, desenvolvidos em prazos curtos e com requisitos em constante mudança, impulsionou as organizações a adotarem a combinação de desenvolvimento ágil e distribuído. Além disso, a presença de recursos localizados em diferentes regiões, que demandam capacitações específicas para cada projeto, reforçou essa escolha. Essa abordagem busca minimizar o tempo e o custo de entrega, trazendo benefícios, mas também gerando uma série de desafios, conforme apontado por Júnior et al.(2011).

1.4 Metodologia do Estudo

Este trabalho segue uma abordagem qualitativa e exploratória, utilizando como estratégia principal o estudo de caso. A pesquisa é direcionada para compreender as características e os desafios de times distribuídos no contexto do Desenvolvimento Distribuído de Software (DDS), com foco na aplicação do método ágil *Scrum* na empresa *Lean Software*.

Características da Investigação:

Natureza do Estudo: O estudo é qualitativo, voltado para interpretar fenômenos em um ambiente real, especificamente a dinâmica dos times distribuídos da empresa analisada.

Estratégia de Pesquisa: O estudo de caso foi escolhido por permitir uma análise aprofundada de um fenômeno específico (o uso do *Scrum* em DDS).

Base Teórica: Os dados foram analisados com base em referenciais teóricos como Borges (2012) e Prikladnicki e Audy (2002), permitindo uma triangulação teórica entre prática e teoria.

Estrutura da Metodologia: A pesquisa utiliza uma abordagem qualitativa e exploratória, estruturada como um estudo de caso na empresa *Lean Software*. Os métodos de coleta incluem análise documental, entrevistas semiestruturadas e observação indireta, com os dados sendo comparados a referenciais teóricos de Borges (2012) e Prikladnicki e Audy (2002). A análise dos dados adota técnicas de triangulação teórica e organização em tabelas para facilitar a comparação e interpretação, garantindo consistência e validade aos resultados. Limitada a uma única organização e método ágil (*Scrum*), a metodologia oferece uma visão profunda, embora específica, do tema.

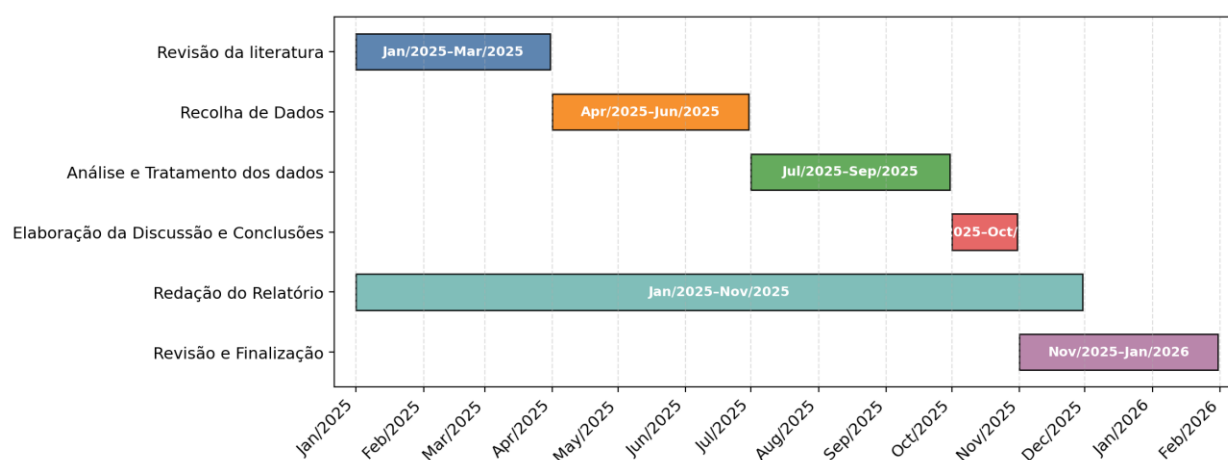
1.5 Planejamento do Projeto

Nesta seção, são descritas as principais etapas do projeto, contemplando os momentos de investigação teórica, elaboração de instrumentos de coleta de dados, realização das entrevistas, análise qualitativa e redação dos resultados. Cada fase foi organizada de forma lógica e sequencial, com prazos e objetivos específicos, respeitando os critérios metodológicos de uma pesquisa qualitativa aplicada a um estudo de caso.

Tabela 1 – Análise do planejamento do projeto

Etapa	Duração	Período
Revisão da literatura	3 meses	Janeiro – Março 2025
Recolha de Dados	3 meses	Abril – Junho 2025
Análise e Tratamento dos dados	3 meses	Julho – Setembro 2025
Elaboração da Discussão e Conclusões	1 mês	Outubro 2025
Redação do Relatório	Ao longo de todo o período	Janeiro – Novembro 2025
Revisão e Finalização	3 mês	Novembro 2025 – Fevereiro 2026

Figura 1. Gráfico da análise do planejamento



1.6 Estrutura de trabalho

Este trabalho está organizado em cinco capítulos.

O Capítulo 1 INTRODUÇÃO apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

O Capítulo 2 REVISÃO BIBLIOGRÁFICA, uma visão teórica sobre os métodos ágeis aprofundando mais no *Scrum* em seu funcionamento e implementação e, além disso, uma visão teórica sobre times distribuídos, sua história e suas características.

O Capítulo 3 DESENVOLVIMENTO DE SOFTWARE COM SCRUM EM UM TIME DISTRIBUÍDO: UM ESTUDO DE CASO apresenta um estudo de caso, ou seja, um caso real sobre a implementação do método *Scrum* em um time distribuído, em como a empresa *Lean Software* atua em situações adversas com projetos de desenvolvimento de *software* de alta complexidade.

O Capítulo 4 ANÁLISE DOS RESULTADOS descreve uma análise comparativa sobre a parte teórica relatada no capítulo 2 e o estudo de caso apresentado no capítulo 3, uma análise de como o método ágil *Scrum* auxiliou de forma efetiva em um time onde os integrantes se encontram de forma distribuída.

O Capítulo 5 CONSIDERAÇÕES FINAIS descreve uma conclusão relatando os pontos em destaques sobre a implementação do método *Scrum* em um cenário complexo de desenvolvimento de software com um time distribuído, relatando pontos positivos e negativos da implementação do método.

Referências listam todas as referências utilizadas para a construção deste trabalho.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado uma descrição teórica sobre os assuntos relacionados com o tema deste trabalho. São apresentados o conceito do método ágil *Scrum* assim como a estrutura do funcionamento do *Scrum*, ciclo de vida, características de um time *Scrum* e, além disso, conceitos sobre Times Distribuídos tais como: história dos times distribuídos, desafios na utilização de times distribuídos, fatores de sucesso para um DDS e utilização do *Scrum* em um ambiente DDS.

2.1 Métodos Ágeis

Um método de desenvolvimento de software é um grupo de atividades que auxiliam a produção de software. O resultado dessas atividades é um produto que reflete a forma como todo processo foi conduzido. Os problemas com os projetos e o desagrado com os métodos de desenvolvimento tradicionais de software como: Cascata, Espiral e Incremental, levaram desenvolvedores de software na década de 1990 a propor novos métodos, dando início ao manifesto ágil segundo Sommerville(2007).

O início das Metodologias Ágeis aconteceu após o manifesto e tornou-se popular quando dezassete especialistas em processos de desenvolvimento de software, representando os métodos *Scrum*, *Extreme Programming*, *Feature Driven Development* (FDD), *Crystal*, entre outros, estabeleceram princípios comuns compartilhados por todos esses métodos. A consequência foi à formação da Aliança Ágil (*Agile Alliance*) e o estabelecimento do “Manifesto Ágil” (*Agile Manifesto*), no ano de 2004. As metodologias ágeis divergem em termos de práticas e ênfases, e compartilham de algumas características como: desenvolvimento iterativo e incremental, comunicação e redução de produtos intermediários (Koscianski & Soares, 2006).

Para Sommerville (2007), com a utilização da metodologia ágil, o *stakeholder* precisa estar sempre acessível. Os desafios na utilização dos métodos ágeis aumentam, principalmente quando se tem muitos *stakeholders* com prioridades distintas. Neste contexto, segundo Sommerville (2007), algumas das motivações para a aplicação dos métodos ágeis e iterativos são as seguintes:

- Clientes ou usuários não terem certeza do que querem;
- Clientes ou usuários terem dificuldade de dizer o que querem e o que sabem;
- Detalhes do que clientes ou usuários precisam vão ser revelados durante o desenvolvimento;
- Os detalhes são complexos para os usuários;
- Forças externas (como um competidor, produto ou serviço) podem levar a modificações.

Koscianski & Soares (2006) descrevem alguns conceitos considerados chaves do “Manifesto Ágil” tais como:

- Indivíduos e interações ao invés de processos e ferramentas;
- Software executável ao invés de documentação;
- Colaboração do cliente ao invés de negociação de contratos;
- Respostas rápidas a mudanças ao invés de seguir planos.

Para Koscianski & Soares (2006) o “Manifesto Ágil” não é contra os processos, documentação, ferramentas e planejamento, porém apenas mostra que têm uma importância secundária quando comparados com:

- Indivíduos;
- Interações;
- Software estar executável;
- Colaboração do cliente;
- Respostas rápidas às mudanças e alterações.

Segundo Pressman (2006) a palavra "Agilidade" tornou-se uma referência quando se descreve um processo atual de software, onde tudo deve ser considerado ágil. Para o autor, uma equipe pode ser considerada ágil quando está preparada para responder de forma adequada às mudanças.

Pode-se considerar mudanças para Pressman (2006), tudo aquilo que o desenvolvimento de software tem como foco principal, como por exemplo: alterações no software em desenvolvimento; alterações no time de desenvolvimento; alterações em consequência a novas tecnologias; todas as alterações que podem ter impacto no produto que é construído.

O autor destaca que o apoio para modificações deve ser absorvido em tudo que é feito em software de forma simples e coesa. Um *time* que se define como ágil consegue reconhecer que o software é criado por pessoas trabalhando em equipes e que as especialidades desses indivíduos e a capacidade de colaboração estão no coração do sucesso do projeto.

Segundo Sommerville (2007) as metodologias ágeis são respostas eficientes às modificações constantes de um produto de software. Elas dispõem de estruturas

que facilitam a comunicação dos membros da equipe, como os engenheiros de software e, além disso, também dos seus respectivos gerentes. Para o autor, a metodologia ágil tem como prioridade a entrega do produto e considera os clientes como integrante da equipe.

Além disso, estudos mais recentes reforçam essa base teórica dos métodos ágeis. Por exemplo, a revisão sistemática de Wonohardjo et al. (2021) demonstrou que o *Scrum*, embora originalmente concebido para equipes co-localizadas, está presente na literatura como abordagem ágil dominante em distintos contextos de desenvolvimento de software. Nesse estudo, destaca-se que a adoção de *Scrum* implica não só práticas formais, mas adaptações culturais, organizacionais e tecnológicas ao contexto real da equipe.

Sommerville (2007) conclui que os métodos ágeis mesmo sendo baseados no desenvolvimento e entregas incrementais, também se socorrem de processos diferentes para conseguir esse objetivo. Contudo, dividem alguns princípios em comum. Esses princípios são listados na tabela 2.

Tabela 2 - Fundamentos dos Métodos Ágeis

Princípio	Descrição
Envolvimento do Cliente	Clientes devem ser profundamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar as iterações do sistema.
Entrega incremental	O software é desenvolvido em incrementos e o cliente especifica os requisitos a serem incluídos em cada incremento.

Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Os membros da equipe devem desenvolver suas próprias formas de trabalhar sem processos prescritivos.
Aceitação das mudanças	Tendo em mente que os requisitos do sistema irão mudar, o sistema deve ser projetado para acomodar essas mudanças.
Manter a simplicidade	Concentração na simplicidade do software que está sendo desenvolvido e do processo de desenvolvimento. Sempre que possível, deve-se trabalhar ativamente para eliminar a complexidade do sistema.

Fonte: Sommerville (2007)

2.2 Scrum

O método ágil *Scrum*, conforme formalizado por Schwaber e Beedle (2002), é um *framework* que auxilia na gestão de times que prioriza as pessoas e reconhece que os requisitos podem mudar rapidamente. Ele organiza o trabalho em ciclos curtos chamados *Sprints*, com duração aproximada de um mês, nos quais a equipe entrega incrementos funcionais de software, buscando maximizar a qualidade em cada iteração. Essa abordagem permite que as equipes respondam rapidamente às mudanças e ajustem continuamente seus planos.

A simplicidade do *Scrum* é intencional: ele define papéis (*Product Owner*, *Scrum Master* e *Developers*), eventos (*Daily Scrum*, *Sprint Planning*, *Sprint Review* e *Retrospective*) e artefatos (*Product Backlog*, *Sprint Backlog* e Incremento), mas não prescreve técnicas específicas de desenvolvimento de software. Essa leveza estrutural facilita a adoção, mas exige maturidade da equipe para gerar valor real. A transparência é fundamental, garantindo que todos os membros da equipe e *stakeholders* tenham

visibilidade sobre o progresso, impedimentos e objetivos, permitindo a inspeção e adaptação constantes.

No *Scrum*, dois processos centrais sustentam seu funcionamento: inspeção e adaptação. Durante o *Sprint*, a equipe realiza inspeções regulares, identifica bloqueios e ajusta seu trabalho. A *Sprint Review* permite avaliar o incremento entregue e coletar *feedback* dos *stakeholders*, enquanto a Retrospective possibilita a reflexão sobre o processo interno, fomentando a melhoria contínua (Shwaber & Beedle, 2002).

Estudos recentes evidenciam que a maturidade da equipe é um fator crítico para a eficácia do *Scrum*. Verwijs & Russo (2021) desenvolveram um modelo teórico empírico mostrando que cinco fatores influenciam significativamente a performance de equipes *Scrum*: *responsividade, preocupação com stakeholders, melhoria contínua, autonomia da equipe e suporte da gestão* (Verwijs & Russo, 2021). Kalluri (2024) demonstrou que variáveis psicológicas, como confiança e segurança psicológica, são determinantes para que a equipe desenvolva autonomia, reduzindo a dependência do *Scrum Master* (Kalluri, 2024).

O papel do *Scrum Master* também evolui com a maturidade da equipe. Em equipes iniciais, o *Scrum Master* centraliza funções de liderança, enquanto em equipes maduras ocorre a transferência gradual de papéis, processo facilitado pela criação deliberada de uma “lacuna de liderança”, permitindo que os membros assumam responsabilidades (Spiegler, Heinecke, & Wagner, 2018; Spiegler et al., 2020). Esse movimento promove auto-organização, confiança e autonomia, elementos centrais para o sucesso do *Scrum*.

Práticas estruturadas contribuem para o aumento da maturidade da equipe. O estudo de Camargo e Galeale (2024) propôs um *roadmap* de práticas ágeis que inclui refinamento contínuo do *backlog*, retrospectivas regulares, uso de métricas de performance e *coaching* ativo do *Scrum Master*. Essas práticas demonstraram ser eficazes para elevar a maturidade técnica, psicológica e organizacional das equipes (Camargo & Galeale, 2024).

Além disso, a literatura aponta desafios para a implementação efetiva do *Scrum*. A adoção “cerimonial”, na qual equipes seguem rituais sem internalizar os valores fundamentais do *Scrum* (compromisso, coragem, abertura, respeito e foco) limita a entrega de valor e a aprendizagem contínua. A escalabilidade também apresenta desafios: equipes grandes ou múltiplas equipes simultâneas podem necessitar de *frameworks* complementares, como LeSS ou SAFe, sem perder a essência ágil (Schwaber & Beedle, 2002; Verwijs & Russo, 2021).

Em síntese, o *Scrum* combina simplicidade estrutural com profundidade psicológica. A verdadeira eficácia do *framework* acontece quando equipes internalizam seus valores, desenvolvem autonomia, compartilham liderança e aprimoram continuamente seus processos, apoiadas por um ambiente organizacional que promove maturidade, confiança e colaboração (Schwaber & Beedle, 2002; Spiegler, Heinecke, & Wagner, 2018; Camargo & Galeale, 2024; Verwijs & Russo, 2021; Kalluri, 2024).

2.2.1 Estrutura do funcionamento do *Scrum*

O *Scrum* enfatiza o uso de um conjunto de “Padrões de processo de software” que se mostraram efetivos para projetos com prazos apertados, requisitos mutantes e criticidade de negócio. Cada um desses padrões de processo define um conjunto de atividades de desenvolvimento, de acordo com Pressman (2006).

A Tabela 3 mostra que a estrutura de funcionamento do *Scrum* é separada em três fases: *Pre-Game*, *Game* e *Post-Game*.

Tabela 3 - Fases do *Scrum*

Fases	Descrição
-------	-----------

Pré-planejamento (<i>Pre-game</i>)	Os requisitos são descritos em um documento chamado <i>backlog</i> . Posteriormente eles são priorizados e são feitas estimativas de esforço para o desenvolvimento de cada requisito. O planejamento inclui a definição da equipe de desenvolvimento, as ferramentas a serem usadas, os possíveis riscos do projeto e as necessidades de treinamento.
Desenvolvimento (<i>Game</i>)	Nesta fase o software é desenvolvido em ciclos (<i>sprints</i>) em que novas funcionalidades são adicionadas. Cada um desses ciclos é desenvolvido de forma tradicional, ou seja, primeiramente faz-se a análise, em seguida o projeto, a implementação e os testes. Cada um desses ciclos é planejado para durar de uma semana a um mês.
Pós-planejamento (<i>Post-Game</i>)	Após a fase de desenvolvimento são feitas reuniões para analisar o progresso do projeto e demonstrar o software atual para os clientes. Nesta fase são feitas as etapas de integração, os testes finais e a documentação.

Fonte adaptada: Koscianski & Soares (2006)

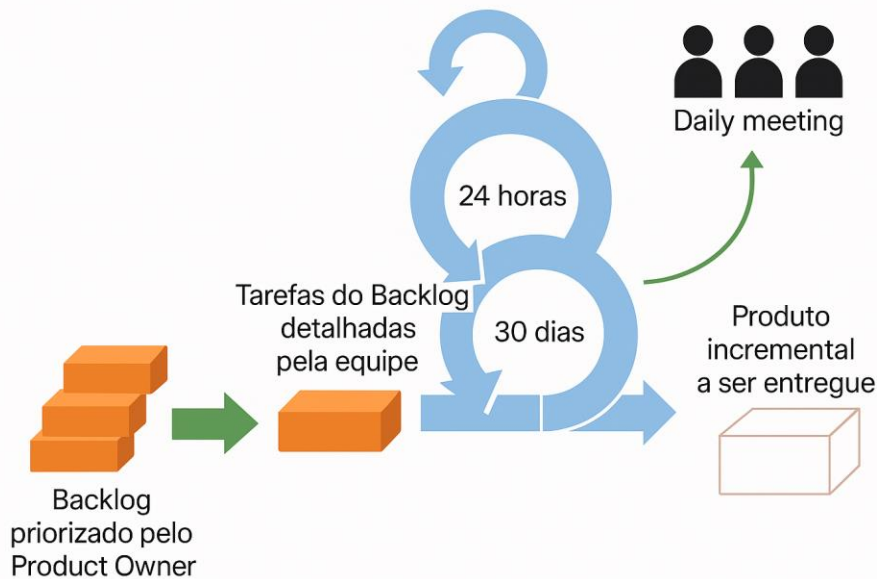
Segundo Koscianski & Soares (2006), em relação à metodologia *Scrum*, existem vários casos de sucesso relatados. Para usar *Scrum* em projetos que necessitem de equipes maiores, deve-se dividir as pessoas em equipes de até 10 integrantes. Isto é necessário para melhorar a comunicação entre os membros das equipes.

Para Kniberg (2007) o melhor e o pior da *Scrum* é que em cada projeto é necessário adaptar o processo a situação específica.

Além disso, segundo Schwaber & Beedle (2002), o *Scrum* pode ser usado em trabalhos complexos nos quais não é possível prever tudo o que irá ocorrer. Ele oferece uma metodologia e um conjunto de práticas que torna

tudo visível. Com isso, é possível para os integrantes do projeto ter o conhecimento do que exatamente está acontecendo em todas as fases e fazer os ajustes necessários para manter o projeto se alterando no decorrer do tempo, a fim de alcançar os seus objetivos.

Figura 2. Ciclo do *Scrum*



Fonte: Pereira et al (2007)

A Figura 2, mostra o ciclo de desenvolvimento do *Scrum*. O *Scrum* tem a evolução de seus ciclos com base em algumas iterações definidas; essas iterações podem levar de 2 a 4 semanas e são chamadas de *Sprint*. A reunião de planejamento é realizada antes do início de cada *sprint*. Nesse momento a equipe de desenvolvimento interage com o cliente que é chamado de *Product Owner* (PO), com o objetivo de priorizar as atividades que serão desenvolvidas e realizar as estimativas das tarefas que o time de desenvolvimento deverá realizar dentro de uma *Sprint*, de acordo com Pereira et al (2007).

Após o planejamento anterior, o *time* segue para a fase de realização da *Sprint*. Nesse momento o *time* tem o controle de sua evolução através das reuniões diárias,

sendo que o objetivo dessa reunião é que cada integrante do time responda a 3 perguntas importantes sobre o desenvolvimento de suas tarefas, sendo elas: O que você fez ontem? O que você fará hoje? Há algum impedimento no seu caminho? A duração da reunião é de aproximadamente 15 minutos e além disso, é exibido o gráfico de andamento do *Sprint* chamando *Burndown*, afirma Pereira et al (2007).

Ao final de cada *Sprint*, realiza-se uma breve revisão do produto a ser entregue, com o objetivo de verificar se tudo o que foi definido no início foi efetivamente implementado. Essa reunião é denominada *Sprint Review* e inclui a demonstração do trabalho realizado pelo time em conjunto com o *Product Owner*. Em seguida, ocorre a reunião de retrospectiva, cujo objetivo é analisar as lições aprendidas a partir dos pontos que o time considera não terem sido bem executados na *Sprint* anterior, visando à melhoria do processo e/ou do produto nas próximas *Sprints*, conforme Pereira et al. (2007).

Segundo Schwaber & Beedle (2002) existem seis papéis distinguíveis em um time *Scrum* que tem tarefas e finalidades diversas no decorrer do processo e de suas práticas e são apresentadas na Tabela 4:

Tabela 4 - Papéis no Scrum

Papel	Descrição
<i>Scrum Master</i>	É um novo gerente introduzido ao <i>Scrum</i> . É responsável por garantir que o projeto seja realizado de acordo com as práticas, os valores e as regras do <i>Scrum</i> e que avança conforme o planejado.
<i>Product Owner</i>	É oficialmente responsável pelo projeto, pelo gerenciamento, controle e por tornar visível a lista de <i>Product Backlog</i> . Ele é selecionado pelo <i>Scrum Master</i> , cliente e gerente.
<i>Scrum Team</i>	É o time do projeto que tem a autoridade para decidir as ações necessárias e organizá-las em ordem para atingir os objetivos de cada <i>Sprint</i> .

Cliente	Participa nas tarefas relacionadas aos itens do <i>Product Backlog</i> para o sistema ser desenvolvido ou aprimorado.
Usuário	É responsável por utilizar o produto quando este estiver em produção.
Gerente	É responsável pela decisão final, juntamente com as normas e convenções a serem seguidas no projeto. O gerente também participa da definição das metas e dos requisitos.

Fonte adaptada: Schwaber & Beedle (2002)

No decorrer da *Sprint*, de forma organizada, o *time* tem a capacidade de controlar as tarefas que serão executadas, não sendo permitido ocorrer mediação externa. Um dos principais papéis do *Scrum Master* é justamente não deixar que interferências externas prejudiquem o objetivo que foi traçado pelo time na execução da *Sprint*, de acordo com Pereira et al (2007).

2.2.2 Características de um *time Scrum*

Para um time que deseja implementar a metodologia *Scrum* como *framework* para o desenvolvimento de seu projeto e obter sucesso com um software que atende as definições de negócio, segundo Sabbagh (2016), é preciso seguir alguns valores que o *Scrum* tem como base para a sua implementação. Eles abrangem as regras descritas em: papéis, artefatos e eventos.

Na Tabela 5, são descritos os cinco valores do *Scrum* segundo Sabbagh (2016).

Tabela 5 - Valores do *Scrum*

Valor	Descrição
-------	-----------

Foco	- Times atuando apenas em um projeto evitando multitarefas. - Times trabalhando com foco em metas de negócios claras e alcançáveis. - <i>Product Owner</i> definindo as necessidades de negócio mais urgentes de acordo com cada momento do projeto. - <i>Scrum Master</i> auxiliando o time a manter o foco no trabalho, facilitando e removendo impedimentos.
Coragem	- <i>Time</i> tem de ter a coragem para aceitar as mudanças como parte atual do processo. - <i>Product Owner</i> tem de ter coragem para confiar no time e deixá-lo livre para realizar necessário para atingir metas. - O time tem de ter coragem para falhar e criar visibilidade sobre os problemas. - <i>Time</i> tem de aprender com as falhas e os problemas o mais cedo possível. - <i>Time</i> tem de ter coragem para mostrar e entregar com frequência as partes criadas do produto. - <i>Scrum Master</i> tem de ter coragem para remover impedimentos e realizar as mudanças necessárias para ajudar a equipe a se tornar mais produtiva.
Franqueza	- Franqueza em realizar inspeção e adaptação melhorando a forma de trabalho do time através de feedbacks. - O <i>time</i> buscar validar os resultados de seu trabalho a partir do feedback frequente sobre o que produzem, o que cria visibilidade sobre as mudanças necessárias. - O <i>time</i> manter visível o progresso de seu trabalho, para poder tomar medidas de correção necessárias. - O <i>time</i> sentir-se seguro para estabelecer e informar suas estimativas de quanto trabalho acredita que pode ser realizado.
Compromisso	- O <i>time</i> determinar como seu trabalho será realizado, monitorar seu progresso e realizar as correções de rumo que achar necessárias. - Em cada ciclo de desenvolvimento, o time se compromete em alcançar as metas de negócio acordada com o <i>Product Owner</i>. - <i>Product Owner</i> se compromete a estabelecer as prioridades de trabalho de forma a satisfazer as necessidades de negócios do projeto.

Respeito	- Os membros do time trabalham juntos, compartilhando responsabilidades. - Todos que trabalham no projeto respeitam as opiniões uns dos outros, ouvem e buscam entender os diferentes pontos de vista. - O <i>Product Owner</i> respeita as decisões técnicas dos membros do time, que respeitam suas decisões de negócios.
----------	---

Fonte: Sabbagh (2016)

Trabalhar em um projeto utilizando a metodologia *Scrum* pode ser um desafio para muitas empresas pois, a quebra de paradigma de desenvolvimento de software, é realmente muito grande levando em consideração a mudança de cultura e hábitos, que precisam ser adaptadas ao iniciar um projeto com uma metodologia ágil *Scrum*. Porém, esse desafio pode-se potencializar quando o *time* de desenvolvimento se encontrar geograficamente em lugares distintos, sendo esses chamados de times distribuídos.

Estudos confirmam que equipes com maior maturidade, autonomia e clareza de papéis tendem a alcançar melhores resultados com *Scrum*. Por exemplo, a investigação de Noll, Razzak & Beecham (2020) sobre motivação e autonomia em equipes globais mostrou que autonomia sozinha não era suficiente para motivar equipes distribuídas, era necessário também competência e sentido de pertencimento para que a metodologia ágil pudesse ter impacto positivo.

Na próxima seção serão detalhadas as características sobre times distribuídos, e seus desafios.

2.3 Times Distribuídos

2.3.1 História dos Times Distribuídos

O Desenvolvimento Distribuído de *Software* (DDS) surgiu como uma alternativa para a resolução da diminuição de mão de obra especializada das empresas. As demandas de profissionais nas empresas têm um alto crescimento e a oferta de

profissionais capacitados não é razoável para preencher a carência de tantos projetos, afirmam Prikladnicki et al. (2002).

O DDS tem como objetivo agilizar o desenvolvimento e, além disso, diminuir o tempo de entrega, e com isso, melhorar a satisfação do cliente. Segundo Prikladnicki & Audy (2002) a diminuição do tempo de desenvolvimento é possível por meio de times trabalhando em paralelo. É possível para as empresas o início de parcerias com o objetivo de estudar mercados maiores, além da clareza em selecionar profissionais com competências específicas em qualquer espaço geográfico (Prikladnicki et al. 2002).

Para que um ambiente DDS seja produtivo, é preciso um grande empenho na gestão do projeto com o objetivo de sincronizar a atuação dos times, sendo que, as tecnologias de comunicação e informação são usadas a fim de auxiliar e gerenciar as dependências, de acordo com Prikladnicki & Audy (2002).

Outros desafios dos times distribuídos são a utilização de ferramentas de controle de versão e o uso de ferramentas compartilhadas de integração dos resultados. Além disso, a intercomunicação cultural e linguística são grandes causas dos problemas de comunicação dos times conforme explicam Prikladnicki & Audy (2002).

Para Prikladnicki & Audy (2002) existem alguns fatores que demonstram o motivo pelo qual o DDS está virando grande tendência, que são relacionados com o ambiente de negócio. Esses são: globalização, aumento da importância dos sistemas de informação nas empresas e processos de terceirização que formam um ambiente adequado a esse cenário de desenvolvimento.

Conforme Borges (2012), o Desenvolvimento Distribuído de Software (DDS) apresenta características que o diferenciam do desenvolvimento co-localizado, realizado no mesmo espaço físico, destacando-se principalmente:

- Distância;
- Diferença em fuso horário;
- Diferença culturais;
- Distância geográfica dos clientes e desenvolvedores;
- Idioma;
- Costumes;

- Comportamentos;

Mesmo com todos os desafios encontrados no desenvolvimento de software distribuído, para Pehls (2016), o DDS tem atraído muitas empresas no mundo, por ser uma forma de baixar o custo em decorrência de incentivos fiscais ou de concentração de massa crítica em algumas áreas, dentro ou fora do país. De acordo com Pehls (2016) o custo de um engenheiro de software nos Estados Unidos é em média US\$ 60 por hora, sendo que, por outro lado, o custo de um engenheiro de software na Índia custa em média US\$ 6 por hora. Nesse contexto, unindo a carência de uma organização com a complexidade em encontrar mão de obra qualificada o DDS torna-se uma grande opção para as empresas que têm como foco a diminuição de custos mantendo a qualidade do desenvolvimento Pehls (2016).

Mais recentemente, a evolução das práticas de trabalho remoto e colaborativo reforçou ainda mais a relevância dos times distribuídos. Por exemplo, o estudo de Stray & Moe (2020) mostrou que equipes que trabalham globalmente passam em média 7h45 por semana em reuniões agendadas e 8h54 em reuniões não-agendadas, e que a falta de disponibilização de pessoas chave e de apoio organizacional para reuniões espontâneas se configuravam como barreiras para a coordenação eficaz entre sites.

2.3.2 Desafios na utilização de Times Distribuídos

Apresentam-se alguns desafios que dificultam a implementação do DDS como a distância física, temporal e além disso diferenças culturais. Com isso algumas tecnologias são utilizadas para dar suporte ao DDS. A tabela 6 explana alguns desafios no processo de desenvolvimento (Prikladnicki et al., 2002):

Tabela 6 - Desafios no desenvolvimento distribuído

Processo	Desafios

Arquitetura do Software	Uma arquitetura adequada é um dos fatores utilizados para a redução do esforço entre as equipes para o DDS e deve-se basear no princípio da modularidade, pois permite alocar tarefas complexas de forma distribuída. Com isso, há uma redução na complexidade e é permitido um desenvolvimento em paralelo simplificado.
Engenharia de Requisitos	A engenharia de requisitos contém diversas tarefas que necessitam de um alto nível de comunicação e coordenação. Assim os problemas apresentados são mais complexos em um contexto de DDS.
Gerência de Configuração	O gerenciamento de configuração é importante para controlar múltiplas peças no DDS. Porém, controlar modificações nos artefatos em cada uma das localidades que participam do processo de desenvolvimento de um produto pode ser complicado. Apesar da utilização de práticas de Gerência de Configuração auxiliarem no controle da documentação e do software, a gerência de modificações simultâneas a partir de locais diferentes é um grande desafio.
Processo de Desenvolvimento	Em projetos DDS, o uso de uma metodologia que auxilia a sincronização e a padronização das atividades é essencial. Isto permite que todos os membros utilizem uma nomenclatura comum em suas atividades.
Testes e Garantia da Qualidade	As práticas de teste e de garantia da qualidade do processo de desenvolvimento têm apresentado grande importância em projetos de software. A globalização do processo de desenvolvimento de software tem destacado a importância da adoção de modelos de qualidade internacionalmente aceitos, tais como ISO 9001 e CMMI visando reduzir os problemas existentes na área de Engenharia de Software.

Fonte: Borges (2012)

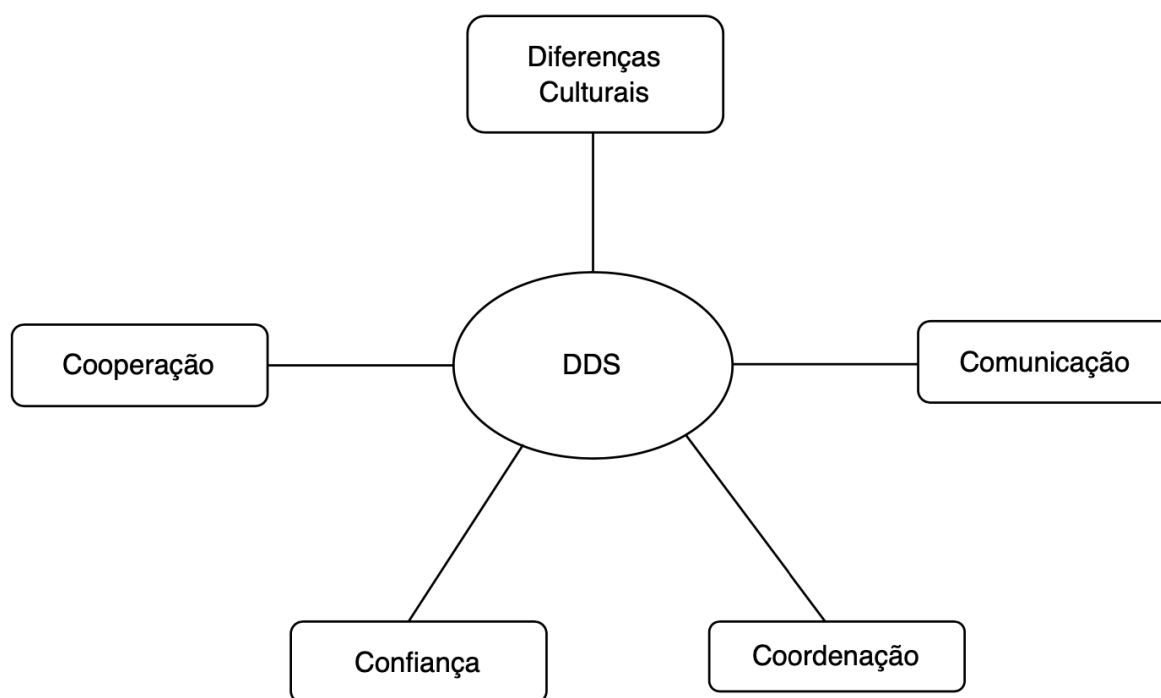
Além dos itens clássicos, pesquisas contemporâneas reforçam que a comunicação eficaz é crítica. Por exemplo, segundo Kostin & Strode (2022) é proposto um modelo composto por três dimensões: transparência da comunicação, qualidade da

comunicação e disciplina da comunicação. Que resultam no alinhamento da compreensão da equipe. Isso destaca que, em times distribuídos que usam *Scrum*, a simples existência de reuniões não é suficiente: a forma, qualidade e disciplina da comunicação são determinantes para o sucesso.

2.3.3 Fatores de sucesso para um DDS

Segundo Prikladinick & Audy (2002) um conjunto de fatores são definidos para determinar o DDS, sendo que estes são considerados cruciais para obtenção do sucesso em seus processos, na figura 3 podem ser identificados os seguintes fatores:

Figura 3. Fatores de sucesso para o DDS



Fonte: adaptado de Prikladinick & Audy (2002)

Tabela 7 - Detalhes dos fatores de sucesso

Fatores	Detalhes
---------	----------

Comunicação	A comunicação em um DDS é a chave para o bom andamento e boa execução de um projeto. Já a falta de comunicação faz com que as equipes fisicamente distantes não saibam de informações básicas sobre o projeto, sobre a própria equipe de projeto, entre outros. Por isso, deve existir um fluxo de informações ágil e eficaz entre os membros da equipe.
Confiança	Confiança em um DDS é muito importante para o fluxo de informações entre as equipes distribuídas. Ter confiança é ter segurança e firmeza no trabalho da equipe como um todo. Por isso, torna-se essencial a procura de mecanismos que propiciem o estabelecimento de um clima de confiança nas relações e ações entre os atores envolvidos.
Cooperação	Cooperação em um DDS significa colaboração da equipe para um objetivo comum, ou seja, possibilita ajudar e pensar como equipe. Por isso, é fundamental a cooperação entre as equipes fisicamente distantes, pois de nada adianta uma boa coordenação e uma ótima comunicação se não existir cooperação entre os membros das equipes.
Coordenação	A coordenação em um DDS é um fator que visa dispor as atividades de forma ordenada baseando-se em processos e regras previamente definidos. Por isso, deve existir um eficiente suporte para a coordenação das atividades de desenvolvimento.
Diferenças Culturais	Quando se desenvolve software em um cenário no qual exista o DDS, uma das primeiras atividades deve ser verificar qual o nível de diferenças culturais existentes entre as equipes fisicamente distantes, pois às vezes determinadas ações podem ser mal interpretadas pelo simples fato de ser parte da cultura de uma equipe e não da outra. As diferenças devem, pois, ser capturadas e adequadas entre os participantes do processo.

Fonte: Prikladinick & Audy (2002)

Os itens da Tabela 7 mostram os fatores considerados essenciais para a realização de um projeto com times distribuídos de sucesso. Assim, a implementação de um processo, adaptado às necessidades de um time que não atua em um mesmo espaço geográfico, é essencial para a obtenção do sucesso, sendo que dessa maneira, uma

empresa que deseja iniciar um desenvolvimento distribuído precisa criar um processo novo para o cenário ou realizar adaptações para o processo atual.

Segundo Pikladinick & Audy (2008) a ausência de um processo que seja adaptado à realidade da empresa pode afetar a falta de sucesso em seus projetos, sendo que um dos maiores riscos no DDS é a criação de um processo que não atenda às reais necessidades da empresa principalmente se tratando de times de desenvolvimento distribuído.

Adicionalmente, o *framework* proposto por Shahzad et al. (2023) consolidou 12 desafios principais enfrentados em equipes *Scrum* distribuídas e apresentou estratégias de mitigação que abrangem comunicação, coordenação, colaboração e uso de ferramentas de forma alinhada. O estudo revelou que 91% dos artigos selecionados discutiam comunicação, coordenação e colaboração como barreiras críticas no contexto de *Scrum* distribuído. Isso reforça que, para alcançar sucesso em DDS, não basta adotar *Scrum* “como está”, mas sim adotar práticas customizadas para o contexto de distribuição geográfica.

Por fim, o processo pode ser conduzido por meio de atividades desnecessárias ou até mesmo incluir tarefas que nunca serão efetivamente realizadas, como planilhas sem instruções claras ou cálculos sem relação prática com o cotidiano da organização. Dessa forma, é fundamental que o gerenciamento de riscos seja implementado por meio de atividades adaptadas à realidade da empresa, garantindo que o processo seja efetivamente aplicado, conforme destacam Pikladnick & Audy (2008).

2.3.4 Utilização do *Scrum* em um ambiente DDS

As metodologias ágeis, assim como o *Scrum* se destacam pela interação dos times por meio de uma comunicação constante e presencial. Porém o desenvolvimento distribuído de software está crescendo gradativamente desde a última década conforme abordam Pikladinick & Audy (2008).

No entanto, o uso de ferramentas e técnicas de desenvolvimento do *Scrum* tem se tornado um grande aliado para a comunicação entre os times que são distribuídos, auxiliando na realização das atividades. Para Boehm (2003) existem algumas formas de

ampliar a satisfação do cliente como: métodos, práticas e técnicas que podem se encontrar em uma metodologia de apoio como o *Scrum*.

Realizar o DDS junto com o *Scrum* está em crescimento no mercado segundo Soares et al. (2009) que realizou um trabalho relatando o DDS em um projeto *Open Source*, alcançando o sucesso por meio desta abordagem. Assim, Soares et al. (2009) afirma que a junção de *Scrum* com DDS pode ser considerado um processo que é focado nas necessidades do cliente de forma simples e objetiva, ao invés da utilização de documentações excessivas.

Para Prikladnicki & Audy (2002) a metodologia *Scrum* tem sido uma boa alternativa para o DDS, sendo que um dos objetivos do desenvolvimento distribuído é agilizar o desenvolvimento e diminuir o tempo de entrega, e com isso, gerando uma grande satisfação para o cliente. Com a diminuição do tempo de desenvolvimento é possível dispor de times trabalhando de forma paralela, como por exemplo, times trabalhando além do fuso-horário. Isto permite que as organizações tenham maior flexibilidade para ter profissionais com conhecimentos específicos independentemente do espaço geográfico.

Segundo Prikladnicki e Audy (2002) as grandes preocupações no gerenciamento de projetos de software distribuído são: comunicação, coordenação e cooperação. Por isso, o *Scrum* tem sido um grande aliado para o DDS, pois os seus conceitos têm como meta otimizar as entregas por meio de incrementos funcionais do produto para os clientes. Além disso, exige que o time tenha uma boa comunicação podendo contribuir sempre em todas as etapas do projeto, apoiando o gerenciamento e suas entregas.

As principais características do *Scrum* para esses projetos são:

- Organização dos recursos em equipes pequenas e multidisciplinares;
- Iterações dos incrementos normalmente mais curtas;
- Envolvimento do cliente durante todo o processo de desenvolvimento.

Com isso, segundo Schwaber & Beedle (2002), o *Scrum* tem sido a solução de muitas empresas para ultrapassar os problemas que são encontrados nos times DDS em relação à comunicação e ao gerenciamento. Assim e neste contexto, para alcançar

um ambiente produtivo, é preciso o acompanhamento do time enfatizando a comunicação e a interação dos membros.

Entretanto, é importante salientar que a adoção do *Scrum* em contexto distribuído não elimina automaticamente os riscos e não garante sucesso. Conforme evidenciado pelo estudo de Prasetya et al. (2021) As equipes distribuídas que adotaram *Scrum* precisaram adaptar fortemente seu processo, ajustar fusos horários, padronizar ferramentas e treinar equipes para obter resultados superiores ao modelo Cascata. Isso reforça que o *Scrum* deve ser adaptado ao ambiente distribuído, e não simplesmente “transportado” da equipe co-localizada para a distribuída sem mudança.

Na tabela 8, são descritos os principais desafios de um projeto DDS e como são solucionados através das práticas do *Scrum*, porém com algumas adaptações para que a metodologia se adeque ao ambiente distribuído de acordo com Prado & Ferrari (2013).

Tabela 8 - Desafios de um projeto DDS

Desafios	Descrição
Gerenciar Equipes Distribuídas	Aumento de reuniões presenciais para validação e avaliação conjunta do desenvolvimento e com isso o time sempre estaria em contato evitando falhas de implementação causadas pela distância física e também a ineficiência das ferramentas de comunicação.
Gestão de Atividades	Para estimativa de tempo das atividades e esforços é recomendado a utilização do <i>Planning Poker</i> em times distribuídos, com apoio de ferramentas web, para realizar estimativas mesmo sem a necessidade de encontro presencial.
Falta de Domínio da Plataforma	Distribuir as tarefas de acordo com o perfil e experiência de cada pessoa do time. Realização de treinamento antes do início do projeto com o objetivo de realizar o nivelamento do conhecimento entre os membros do time.

Ferramentas de Comunicação	A videoconferência é o meio de comunicação mais utilizado no processo de comunicação síncrona, e além disso, o uso de telepresença em todos os espaços físicos com o objetivo que os membros tenham a oportunidade de se verem sempre que necessário.
Reuniões Diárias	As reuniões diárias são de extrema importância para os times distribuídos pois é o momento em que o time pode realizar uma conversa diária sobre as suas atividades. A reunião deve durar 15 minutos, porém é realizada em um formato diferente, sendo que o time utiliza audioconferência, garantido assim que o time tenha uma interação diária.
Integração	A comunicação deve sempre ser tratada como um elemento essencial para garantir a integração dos artefatos com o mínimo de problemas. Por isso, é importante que o time mantenha um bom plano de SCM, revisado periodicamente durante as reuniões de planejamento.

Fonte: Prado & Ferrari (2013)

Para Schwaber & Beedle (2002) a metodologia *Scrum* tem uma série de práticas que possibilitam o gerenciamento e a boa comunicação. Por ser uma metodologia simples, e com papéis definidos se tornou a mais utilizada pelas organizações que utilizam times distribuídos. Sua flexibilidade e adaptação possibilita às organizações na adaptação da metodologia no ambiente distribuído de acordo com cada necessidade do projeto.

Adicionalmente, a pesquisa de Kostin & Strode (2022) mostra que em equipes *Scrum* distribuídas globalmente, a simples adoção de cerimônias ágeis não basta, é necessário garantir disciplina na comunicação, transparência e qualidade da interação para alinhar o “*mental model*” da equipe e evitar falhas de coordenação. Por fim, o estudo de Shahzad et al. (2023) indica que o uso de *frameworks* específicos para “*Distributed Scrum*” com mitigação explícita de riscos de comunicação, coordenação e colaboração

ajuda a tornar o *Scrum* mais efetivo em contextos distribuídos, o que reforça a necessidade de adaptação do *Scrum* ao ambiente DDS.

2.4 Considerações do Capítulo

Os métodos ágeis consistem em desenvolvimento de software através de entregas incrementais. Todos os métodos consistem nos mesmos princípios:

- Envolvimento do Cliente;
- Entrega Incremental;
- Pessoas, não processos;
- Aceite mudanças;
- Mantenha a simplicidade.

O método *Scrum* não é um processo prescritivo, ou seja, não mostra o que fazer em cada situação do projeto, porém é sempre realizada uma inspeção e adaptação em cada incremento com o objetivo da melhoria contínua. O time precisa estar preparado para as mudanças de requisitos pois os mesmos devem e vão mudar. Os valores do *Scrum* focam-se em pessoas e comportamentos.

O DDS deu início, através da necessidade das empresas, na construção de software de qualidade e menor custo. Os desafios do DDS são a comunicação fluida e colaborativa, sendo seus fatores de sucesso acompanhados, a fim de garantir as entregas.

Considerando a necessidade das organizações em entregar projetos de qualidade por meio da abordagem DDS e o foco do *Scrum* em pessoas e comunicação eficaz, a integração dessas práticas pode gerar ganhos significativos em eficiência e qualidade. Contudo, essa combinação exige atenção a desafios como o alinhamento cultural, a capacitação contínua das equipes e a adaptação dos processos aos diferentes contextos organizacionais, fatores que influenciam diretamente seus resultados.

Em resumo, a literatura recente evidencia que, embora o *Scrum* seja um *framework* amplamente adotado, em ambientes com equipes distribuídas geograficamente o seu sucesso depende fortemente da adaptação do processo, da

maturidade da equipe, da qualidade da comunicação, da coordenação eficaz e da cooperação entre sites. Sem tais elementos, a adoção de *Scrum* pode falhar ou trazer apenas benefícios limitados. Logo, o presente trabalho parte do pressuposto de que o *Scrum*, quando adaptado e bem implementado em cenário de software distribuído pode ser uma ferramenta de apoio valiosa para o desenvolvimento de software com times distribuídos.

3. DESENVOLVIMENTO DE SOFTWARE COM SCRUM EM UM TIME DISTRIBUÍDO: UM ESTUDO DE CASO

Neste capítulo é apresentado um cenário real no qual a empresa autorizou a publicação, porém todos os nomes são fictícios, com isso, o objetivo deste estudo de caso é analisar um projeto de alta complexidade técnica e comportamental em como foi possível aplicar o método *Scrum* em um time geograficamente distribuído, realizando uma entrega de qualidade e gerando satisfação para todos os envolvidos.

Esse projeto surgiu por uma necessidade de um grande banco brasileiro que tinha a necessidade de uma entrega imediata. Porém, o time tinha mais desafios do que apenas uma entrega de qualidade, como o time distribuído e pessoas com necessidades especiais. É mostrado, também, como o método ágil teve um papel fundamental para manter uma boa comunicação e fluidez no projeto.

A autora deste estudo de caso participou do projeto a ser descrito com o papel de Arquiteta de Software.

3.1. História da empresa *Lean Software*

A Empresa *Lean Software* é uma multinacional brasileira de soluções digitais. Tem-se tornado parceira de grandes marcas em seus desafios de transformação digital, adotando práticas de *design*, método ágil e filosofia *lean*. É composta por 2.500 funcionários que estão localizados em Portugal, Brasil, Espanha, EUA, Canadá, Inglaterra, Japão e China.

A *Lean Software* foi fundada em 1995 por estudantes de Ciência da Computação de uma Universidade Estadual de São Paulo, a empresa tem como objetivo a transformação digital das empresas parceiras, e com isso, disseminar o seu vasto conhecimento em processos com o intuito de acelerar a mudança nos processos e na cultura das empresas.

A empresa se destaca pela sua diversidade em ter um processo seletivo que abrange profissionais de diferentes países e culturas, além de prover aprendizado em novas tecnologias. Todos os colaboradores da *Lean Software* são engajados e estão sempre aprimorando seus conhecimentos para o melhor desenvolvimento profissional e com isso, ajudando a empresa a estar sempre a frente em tecnologias e inclusão social.

A transformação digital, segundo a empresa *Lean Software* consiste em mudança de cultura, processos, estratégia e estrutura de uma empresa, a fim de ganhar mais produtividade, aumentando cada vez mais o foco em seus clientes e funcionários.

Os clientes da *Lean Software* são empresas que querem passar por uma transformação digital em toda a sua estrutura desde a concepção de um produto, de um time de desenvolvimento até a entrega para o seu cliente final, desse modo a *Lean Software* além de realizar o desenvolvimento de *software* realiza consultoria de como mudar a cultura de seus parceiros.

Os clientes são na grande maioria do setor financeiro, além de empresas de varejo e saúde, fazendo com que sua proposta em transformação digital e a criação de *software* de qualidade em menor tempo de desenvolvimento, seja presente em diversas empresas de Portugal e do mundo.

3.2 O desenvolvimento de *software* na empresa *Lean Software*

A área de desenvolvimento da *Lean Software* opera com times estruturados *Scrum*, adotando práticas consolidadas do ágil, porém com adaptações específicas para o contexto de *Distributed Software Development* (DDS). Conforme a literatura aponta, times distribuídos exigem maior disciplina comunicacional, clareza de papéis e

mecanismos de transparência (Prikladnicki & Audy, 2008; Stray & Moe, 2020). O modelo da empresa é coerente com esses achados.

Cada *squad* da *Lean Software* é composta por profissionais distribuídos entre Lisboa, Campinas, Belo Horizonte e Madrid, refletindo um ecossistema multicultural e geograficamente disperso. Com base nos dados coletados nas entrevistas, a composição típica das funções é a seguinte:

- Lisboa → *Scrum Masters*, *Tech Lead*/Arquiteto de Software, *Frontend Developer*, *Business Analyst*, *Project Manager*
- Campinas → *Product Owners*, *QA Engineer*, *Full Stack Developer*, *UX Designer*
- Belo Horizonte → *Backend Developer*, *DevOps Engineer*
- Madrid → *Senior QA*, *QA Engineer*

Tal distribuição difere do modelo simplificado inicialmente previsto nos documentos internos da empresa (4 Devs, 1 PO, 1 SM, 1 QA e 1 Arquiteto). A prática organizacional revelou equipes mais diversas e alinhadas às necessidades dos projetos complexos em escala.

Os times são formados de modo flexível, conforme demanda do projeto, e cada *squad* é autônoma, porém integrada a uma estratégia global de produto. A *Lean Software* utiliza o termo “*squad*” para designar seus times *Scrum*, permitindo a existência de múltiplos *squads* atuando paralelamente em um mesmo produto, conforme orientação de escalabilidade já discutida por Shahzad et al. (2023).

Antes do início de um novo projeto, realiza-se a reunião de *Inception*, na qual o time discute objetivos, personas, funcionalidades em alto nível e *roadmap* preliminar. Esse alinhamento inicial é particularmente importante para equipes remotas, pois reduz ruídos de comunicação que tendem a se amplificar em ambientes distribuídos (Borges, 2012).

Após o entendimento geral do produto, o *Scrum Master* (SM) e o *Product Owner* (PO) conduzem o processo de priorização do backlog, elaborando a *business narrative*, documento que descreve as funcionalidades do *Sprint*. Em seguida, o *Tech*

Lead/Arquiteto desenvolve a *technical narrative*, antecipando riscos e decisões técnicas. Ambas são apresentadas no *Sprint Planning*, garantindo que todo o time, independentemente da localização, compreenda o escopo.

No dia seguinte ao planejamento, ocorre a *Technical Alignment* reunião conduzida pelo Arquiteto, para assegurar que todos os desenvolvedores possuam entendimento uniforme da solução técnica. Essas práticas dialogam diretamente com recomendações da literatura para mitigação de gaps em equipes distribuídas (Stray & Moe, 2020).

O *Sprint* é conduzido ao longo de duas semanas, com cadências fixas de cerimônias. Ao final de cada ciclo, realiza-se a *Sprint Review* e a *Retrospective*, nas quais são definidos planos de ação estruturados, mecanismo que reforça a melhoria contínua em times distribuídos, conforme descrito por Noll, Razzak & Beecham (2020).

Assim, o modelo adotado pela *Lean Software* combina a estrutura do *Scrum* com adaptações orientadas à realidade distribuída, garantindo alinhamento frequente, previsibilidade e qualidade técnica mesmo com múltiplos fusos horários.

3.3 Projeto Lista de Pagamentos

O projeto Lista de Pagamentos foi desenvolvido para um cliente do setor financeiro, com o objetivo de permitir que usuários realizassem o pagamento de múltiplos boletos em uma única operação no aplicativo móvel do banco. A natureza do projeto exigiu alta confiabilidade técnica, prazos rígidos e validações regulatórias, ampliando os desafios usuais do desenvolvimento distribuído.

A equipe foi organizada em duas *squads Scrum* operando em paralelo, cada uma composta por:

- Desenvolvedores (Frontend, Backend e Full Stack)
- *QA Engineers* (incluindo profissionais sediados em Campinas e Madrid)
- *Product Owners*
- *Scrum Masters*

- *Tech Lead*/Arquiteto (compartilhado entre as *squads*)

Essa formação distribuída envolvia profissionais localizados em Lisboa, Campinas, Belo Horizonte e Madrid, exigindo coordenação intensa e controle rigoroso do fluxo de informação. Estudos como os de Stray & Moe (2020) demonstram que times distribuídos tendem a investir mais tempo em reuniões síncronas e comunicação assíncrona estruturada, fenômeno observado também no projeto analisado.

As cerimônias do *Scrum* foram conduzidas diariamente via *GoToMeeting* ou *Google Hangouts*, com webcams sempre habilitadas, prática recomendada para reduzir a distância psicológica entre membros remotos (Shahzad et al., 2023). Ferramentas como *Jira*, *Confluence*, *Goal* e *Slack* foram fundamentais para manter a transparência do *backlog*, discussões técnicas e documentação.

Devido à criticidade do projeto, que envolvia prazos legais e potenciais multas governamentais ao banco, as *squads* mantiveram um regime de acompanhamento detalhado, revisando métricas de produtividade, qualidade e estabilidade da aplicação ao longo do *Sprint*. A presença constante do *Tech Lead* contribuiu para antecipação de riscos técnicos, prática destacada como essencial para equipes distribuídas por Prikladnicki e Audy (2002).

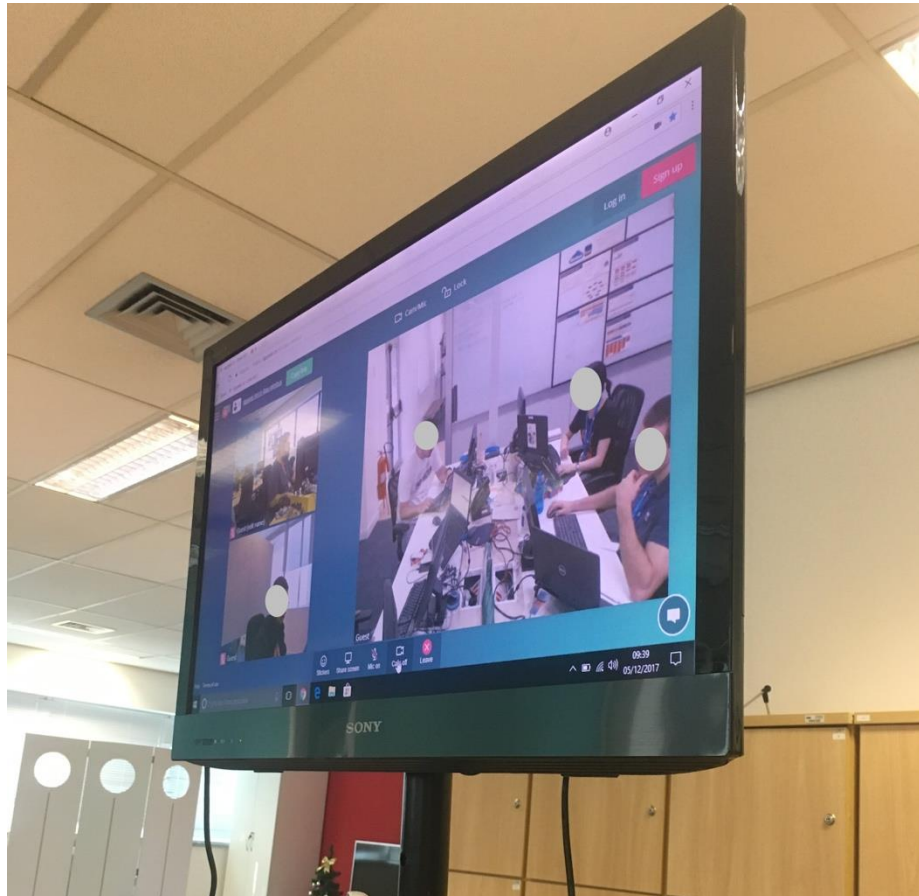
A principal dificuldade relatada foi a coordenação de alinhar decisões entre profissionais em três países, exigindo disciplina, clareza comunicacional e reuniões pré-planejadas para evitar perdas de informação. Ainda assim, o modelo adotado promoveu resultados consistentes e entregas alinhadas ao prazo estabelecido pelo cliente.

Assim como em todos os projetos da *Lean Software* foram utilizadas algumas ferramentas para comunicação remota (*Hang Out*, E-mail, *Go to Meeting* e *Slack*) e para a organização do projeto (*Jira* e *Goal*) são ferramentas de mercado que eram aliadas para o dia a dia dos integrantes do time.

Para todas as reuniões realizadas pelo time como: Reuniões diárias, planejamento, retrospectivas, revisão da *sprint* e demonstração eram realizadas via "*Go To Meeting*" ou "*Hang Out*" com as web câmeras abertas para que todos os membros do

time possam ter contato visual uns com os outros, como pode ser observado na Figura 4;

Figura 4. *Go To Meeting* telepresença com todo o time



Fonte: Elaborada pela Autora

Todos os demais membros do time diariamente faziam uma revisão em sua forma de se expressar para que todos do time tivessem a capacidade de compreender o que estava sendo passado, logo era possível que todos compreendessem o status atual do projeto.

3.4 Adaptações do *Scrum* para times distribuídos

O processo *Scrum* no projeto Lista de Pagamentos precisou ser ajustado para acomodar o trabalho totalmente distribuído. Como evidenciado pela literatura, a

aplicação de métodos ágeis em DDS requer cuidados adicionais, principalmente no que se refere à comunicação, confiança e visibilidade do trabalho (Borges, 2012; Noll et al., 2020)

A Tabela 9 resume as fases do processo adaptado:

Tabela 9 - Fases do *Scrum* no projeto Lista de Pagamentos

Fases	Descrição
Pré-planejamento (<i>Pre-game</i>)	Os requisitos são descritos em um documento chamado <i>backlog</i> em uma ferramenta <i>web</i> . Nesse momento todo o time de desenvolvimento é mobilizado e por meio de uma reunião remota é realizada a estimativa de cada tarefa a ser executada e é utilizada também uma ferramenta <i>web/mobile</i> para apoiar nas estimativas.
Desenvolvimento (<i>game</i>)	Os <i>sprints</i> são de 2 semanas no qual 2 dias da semana o time para, para realizar as reuniões de retrospectiva, análise de causa raiz de bugs e planejamento, tudo feito de forma remota com todo o cuidado para que nenhuma informação seja esquecida.
Pós-planejamento (<i>post-game</i>)	Após a fase de desenvolvimento são feitas reuniões para analisar o progresso do projeto e demonstrar o <i>software</i> atual para os clientes. Nesta fase são feitas as etapas de integração, testes finais e documentação.

Fonte: Koscianski et al. (2008)

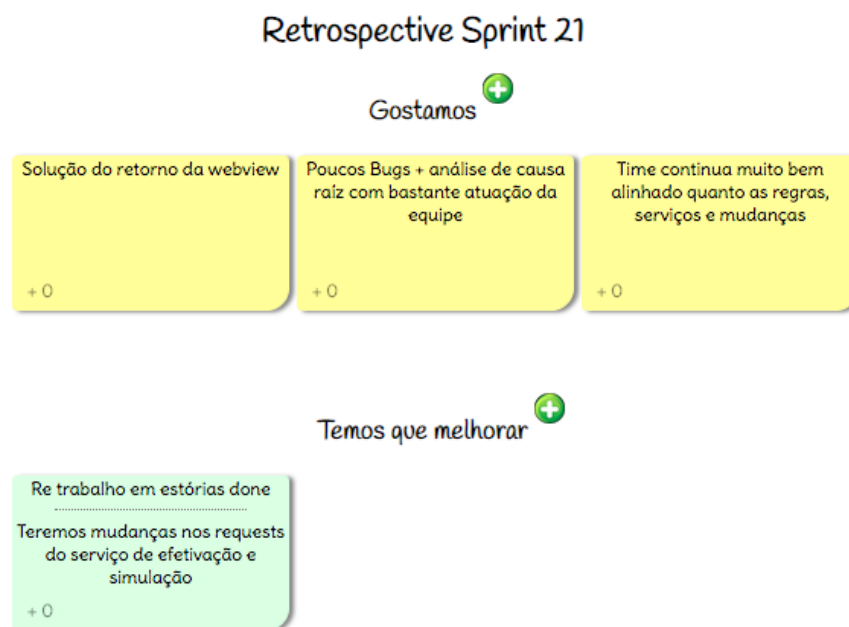
Além das atividades de cada fase do *Scrum*, exibidas na tabela 9, uma reunião chamada de *Sprint Review* é realizada na metade de cada *sprint* com o objetivo de analisar se o *sprint* está no caminho de entrega com sucesso, essa reunião acontece com todos os membros do time juntos de forma remota e a cada *sprint* um membro do time é responsável pelo preparo do material e apresentação da reunião, garantindo com isso, que todos estejam envolvidos 100% no projeto.

Cada característica de um time *Scrum* em um ambiente remoto aumentam drasticamente, cada uma delas tem um peso maior pois o Foco, Coragem, Franqueza,

Compromisso e o Respeito, além de ser algo que se espera naturalmente de um time, nesse cenário ele é cobrado de forma persistente a cada iteração.

As reuniões de retrospectivas têm o objetivo de sempre gerar "plano de ação" para cada item que se considera a melhorar no projeto, esse item é criado do *board* de projeto do time, como pode ser observado na Figura 5 e 6, e um responsável para cada item é eleito pelo time para a sua execução. A cada reunião de *Sprint Review* os itens são revisados garantindo a melhoria contínua do projeto e de seus processos.

Figura 5. *Board* de Retrospectiva gostamos e a melhorar



Fonte: Elaborada pela Autora

Figura 6. *Board* de Retrospectiva aprendemos e expectativa

Retrospective Sprint 21

Aprendemos

Mandar para a Elenilva fazer testes antes é uma boa prática

+0

Chamar de block o que realmente for um block impeditivo

+0

Consultar os times ao lado, pode ajudar nas soluções

+0

Expectativa

Entregar a experiência full no prazo (até o dia 26/10) para que os testes sejam feitos

Ter stories sem changes, ajudaria a homologar mais rapidamente

+0

Pirâmide de testes

+0

Fonte: Elaborada pela Autora

Com isso, pode-se observar que as adaptações realizadas no projeto para que se tornasse melhor a convivência remota do time visando sempre o acompanhamento das pessoas são:

- Reunião de análise de causa raiz dos bugs encontrados do projeto com o objetivo da melhoria contínua do código a ser entregue.
- Reuniões semanais com o objetivo de mostrar para todo o time o momento exato em que o projeto está mostrando métricas de qualidade e de velocidade nas entregas.
- A presença de um arquiteto de *software* com o objetivo de mitigar os riscos técnicos a serem enfrentados pelo time e com isso, adiantar-se com soluções e narrativas técnicas.
- Apresentação da narrativa técnica realizada pelo arquiteto de *software* com o objetivo de mostrar para o time os desafios do *sprint* seguinte.

As adaptações foram realizadas com o objetivo de tornar o time remoto o mais próximo possível do projeto e seus desafios. Com isso, a cada iteração realizando ajustes necessários a fim de realizar uma entrega de qualidade e satisfatória para todos.

4. ANÁLISE DE RESULTADOS

Este capítulo apresenta a análise detalhada da implementação do *Scrum* em equipes distribuídas na empresa *Lean Software*, combinando dados obtidos a partir da análise do time e das entrevistas semiestruturadas realizadas com membros da equipe.

O objetivo é compreender como a metodologia ágil é aplicada, quais adaptações são realizadas e como os desafios de comunicação, colaboração, fusos horários e diversidade cultural são enfrentados. A análise é apoiada na literatura sobre desenvolvimento distribuído de software (DDS) e metodologias ágeis (Borgers, 2012; Prikladnicki & Audy, 2002).

Além disso, a discussão incorpora evidências práticas obtidas nas entrevistas com quinze colaboradores, representando diferentes funções, níveis de experiência e localidades, de modo a refletir uma visão ampla e multidimensional da aplicação do *Scrum* em contexto distribuído.

4.1 Análise do Time Distribuído

A empresa *Lean Software* atua com equipes distribuídas em Lisboa, Madrid, Belo Horizonte, Campinas. Esse modelo de DDS é adotado devido à flexibilidade e à disponibilidade de profissionais qualificados. Apesar dos desafios inerentes à distância, à comunicação e às diferenças culturais, a metodologia *Scrum* fornece suporte estrutural e flexível para o gerenciamento desses projetos, promovendo integração e visibilidade contínua do progresso do trabalho.

Segundo Shahzad et al. (2023), a adoção de *frameworks* ágeis em contextos distribuídos requer ajustes culturais e operacionais constantes, algo observado na *Lean Software*, que adota práticas iterativas para adaptar suas cerimônias e artefatos.

Com o amadurecimento do modelo distribuído, a empresa também passou a integrar colaboradores na Espanha (Madrid), o que adicionou uma nova dimensão cultural e linguística às dinâmicas de trabalho, ampliando a necessidade de práticas ágeis mais adaptativas.

4.1.1 Características do Time Distribuído

O time atende às principais características de equipes distribuídas, conforme Borges (2012):

Tabela 10 - Característica da equipe

Característica	Descrição
Distância	Lisboa, Belo Horizonte, Campinas e Madri
Diferença de fuso horário	Até 4 horas; ajustes em horários de <i>Dailys</i> e <i>Planning</i>
Diferenças culturais	Brasileiros, Portugueses e Espanhóis; variações nos estilos de comunicação e tomada de decisão
Idioma	Português e Inglês
Comportamentos	Comunicação intensa, engajamento, empatia intercultural e melhoria contínua

Essas características evidenciam que, embora exista uma base linguística comum, as sutis variações culturais influenciam o ritmo das entregas, o estilo de *feedback* e a dinâmica de reuniões. A presença de múltiplos fusos horários exige planejamento cuidadoso das cerimônias do *Scrum* para garantir participação equitativa.

4.1.2 Desafios e fatores de sucesso

A implementação do DDS envolve desafios técnicos e humanos, mitigados por práticas estruturadas:

Tabela 11 - Desafios e estratégia

Desafio	Estratégia de mitigação
---------	-------------------------

Arquitetura de software	Documentação atualizada, revisões contínuas e uso de repositórios compartilhados
Engenharia de requisitos	Documentação técnica detalhada com participação do Product Owner e stakeholders
Configuração e controle de versão	Práticas DevOps e ferramentas de versionamento integradas (Git, Bitbucket)
Processo de desenvolvimento	Suporte do <i>Scrum</i> para comunicação e acompanhamento transparente
Testes e QA	Testes automatizados e QA dedicado em cada <i>sprint</i>

Entre os principais fatores de sucesso, destacam-se a comunicação constante, a confiança, a cooperação e a coordenação clara. Além disso, práticas como retrospectivas participativas e uso intensivo de ferramentas de rastreamento (Jira, *Confluence*, *Slack*) contribuem para mitigar diferenças culturais e fortalecer a coesão da equipe. Conforme Gren et al. (2017), tais práticas aumentam a maturidade e a coesão de equipes ágeis distribuídas.

Observou-se também que o comprometimento dos líderes em promover um ambiente psicologicamente seguro foi decisivo para a estabilidade do time, reforçando o valor da confiança em contextos distribuídos.

4.2 Análise das Entrevistas

Foram realizadas 15 entrevistas com membros da equipe em diferentes papéis (*Scrum Master*, *Product Owner*, Desenvolvedores, QA, *UX Designer*, *DevOps*, *Tech Lead*, Analista de Negócios e Gestor de Projeto), com experiência entre 4 e 12 anos e certificações formais (CSM, PSM ou equivalentes). O Anexo A apresenta o guião utilizado.

A análise das entrevistas reforça os achados sobre o funcionamento do *Scrum* em DDS e amplia a compreensão sobre as adaptações necessárias em ambientes multilocais.

4.2.1 Avaliação da eficácia do *Scrum*

O *Scrum* foi considerado eficaz para equipes distribuídas, destacando-se em:

- Transparência do backlog e tarefas;
- Inspeção contínua por meio de reuniões frequentes;
- Agilidade e resposta rápida a mudanças;
- Engajamento de todos, mesmo remotamente.

A maioria dos entrevistados ressaltou que a clareza dos papéis e a cadência das cerimônias foram fatores determinantes para a produtividade.

O Participante G (*Product Owner*, Campinas) afirmou que “a disciplina das reuniões mantém a conexão entre as equipes mesmo com a distância; sem isso, a visibilidade do projeto se perderia rapidamente”.

Sugestões de melhorias incluem automação de testes, integração de ferramentas e ajustes nos rituais para reduzir tempo ocioso em fusos diferentes. Também foi proposto um maior investimento em formação intercultural para novos membros remotos, a fim de acelerar a integração ao time.

Observou-se que o *Scrum*, ao estruturar ciclos curtos de entrega (*Sprints*), reforçou a previsibilidade das entregas e a percepção de progresso contínuo, mesmo quando os membros da equipe estavam em diferentes países. Esse fator foi recorrente nos depoimentos e aparece como elemento chave para a sensação de pertencimento e alinhamento coletivo. O Participante F (*Tech Lead*, Lisboa) destacou que “a *Sprint Review* deixa todo mundo na mesma página. Não tem como alguém se sentir ‘fora’ do projeto se está vendo e discutindo a entrega junto.”

Schwaber & Beedle(2002) afirmam que o *Scrum* cria um ambiente controlado de inspeção e adaptação, no qual erros e desvios são corrigidos rapidamente. A experiência observada na *Lean Software* confirma essa afirmação, uma vez que os ritos de retrospectiva foram consistentemente utilizados como espaço de melhoria contínua. Segundo o Participante D (Desenvolvedor *Frontend*, Lisboa) “a retrospectiva virou um momento de ajuste emocional e técnico. A gente fala do código, mas também fala de gente.”

Além disso, a eficácia do Scrum ficou evidente na forma como a comunicação estruturada reduziu mal-entendidos, aspecto crítico em times distribuídos, conforme apontado por Borges (2012). A literatura destaca o risco de perda de alinhamento quando equipes não compartilham o mesmo espaço físico, enquanto os resultados empíricos desta pesquisa indicam que a cadência das cerimônias substituiu parcialmente o papel das interações presenciais.

Outro aspecto relevante identificado foi o fortalecimento da autonomia individual, acompanhado de um senso de responsabilidade coletiva. O Participante I (*DevOps Engineer*, Belo Horizonte) relatou que “cada um sabe o que está fazendo e por quê. O *Scrum* dá liberdade, mas também deixa tudo exposto. Ninguém ‘desaparece’ no projeto.”

Essa percepção converge com Highsmith (2002), ao afirmar que métodos ágeis reforçam a autonomia orientada por propósito, ao contrário de modelos tradicionais que dependem de controle hierárquico mais rígido.

Entretanto, foram identificados limites práticos na aplicação do *Scrum*. A necessidade de reuniões frequentes, quando alinhadas a fusos diferentes, pode aumentar a carga cognitiva e resultar em fadiga organizacional. O Participante H (QA, Espanha) afirmou que “a metodologia funciona, mas às vezes cansa. Especialmente quando temos entregas críticas e as reuniões se acumulam.”

Esse ponto encontra respaldo em Prikladnicki e Audy (2002), que sustentam que equipes distribuídas demandam equilíbrio entre sincronia e flexibilidade, sob risco de sobrecarga comunicacional.

O Participante N (QA, Madrid) observou que a previsibilidade das entregas aumentou após a padronização dos critérios de aceitação no *Jira* e *Confluence*: “Quando todos falam a mesma língua técnica, o fuso horário deixa de ser um obstáculo e vira apenas um detalhe operacional”.

Já o Participante O (Gestor de Projeto, Lisboa) destacou a importância da *Sprint Review* como mecanismo de alinhamento estratégico, afirmando que “ela evita que times

remotos se transformem em silos isolados, pois dá visibilidade ao progresso para toda a organização”.

Esses achados reforçam o que Schwaber & Beedle(2002) afirmam sobre a transparência como princípio central do *Scrum*.

Outra limitação observada foi a dependência do *Product Owner* na manutenção da clareza do backlog. Quando essa figura está sobrecarregada, o fluxo de trabalho tende a perder precisão, resultando em retrabalho ou interpretação divergente de requisitos. O Participante C (PO, Campinas) confirmou que “se o *backlog* não estiver bem escrito, o time desanda rápido. No remoto, não tem margem para suposição.”

Com base nos resultados, recomenda-se:

- Revisar a duração e frequência das cerimônias, especialmente em semanas de entregas críticas;
- Instituir um programa formal de *onboarding* multicultural, incluindo sensibilização para comunicação entre países;
- Aumentar a automação de testes e integração contínua, reduzindo dependência de retrabalho manual;
- Padronizar critérios de aceitação e *Definition of Done* entre todas as localidades.

Assim, conclui-se que o *Scrum* demonstrou elevada eficácia no contexto analisado, mas depende da maturidade das práticas, da clareza dos papéis e da capacidade de adaptação da equipe confirmando, ao mesmo tempo, os benefícios previstos pela literatura ágil e os desafios descritos nos estudos sobre Desenvolvimento Distribuído de Software.

4.2.2 Experiência com equipes distribuídas

Todos os entrevistados relataram experiências anteriores em DDS, enfrentando fusos horários distintos e barreiras de comunicação mitigadas com videoconferências e ferramentas online.

“No início, tivemos problemas de sobreposição de horários, mas ajustamos a *daily* para horários intermediários, garantindo participação plena.” (Participante B)

Participantes localizados em fusos distintos relataram que o equilíbrio entre atividades síncronas e assíncronas foi crucial para o sucesso. As gravações de reuniões e o uso de canais dedicados de comunicação assíncrona (como o Slack e o Confluence) permitiram que todos se mantivessem alinhados, mesmo em horários diferentes.

A literatura destaca que a sobreposição de horários é fator crítico para DDS (Priklanick & Audy, 2008). Kostin & Strode (2022) complementam que a qualidade e a disciplina na comunicação são determinantes para o desempenho de times distribuídos, o que se reflete na prática da *Lean Software*.

Esta seção sintetiza as percepções dos entrevistados sobre os desafios e adaptações relacionadas à experiência em Desenvolvimento Distribuído de Software (DDS), focando especificamente em fusos horários e comunicação.

- De acordo com a Tabela 12.1, os participantes de A a E destacam desafios como o ritmo das cerimônias e a necessidade de liderança à distância, utilizando ferramentas como Jira e reuniões técnicas para mitigar as distâncias
- Conforme apresentado na Tabela 12.2, os participantes de F a J focam na sincronização de processos técnicos, como testes e design, recorrendo à automação e sessões colaborativas
- Segundo a Tabela 12.3, os participantes de K a O reforçam a importância da clareza documental, revisões de código e o uso do *Scrum* como ponto de convergência para a colaboração multidisciplinar

Tabela 12.1: Síntese de Experiência em DDS (Participantes A a E)

Questão	Participante A (SM, Lisboa)	Participante B (Dev, BH)	Participante C (PO, Campinas)	Participante D (Dev, Lisboa)	Participante E (QA, Campinas)
Experiência prévia em DDS?	Sim, em 3 localidades	Sim	Sim	Sim	Sim
Maior desafio do DDS?	Manter engajamento/ Ritmo das cerimônias	Sobreposição/Comunicação	Coesão e alinhamento	Alinhamento remoto	Liderança à distância
Adaptações para fusos?	Horários intermediários (Dailys); Webcam obrigatória	Equilíbrio Síncrono/Assíncrono (Slack/Zoom)	Documentação técnica detalhada	Uso intensivo de Jira/Confluence	Padrões unificados e reuniões técnicas

Tabela 12.2: Síntese de Experiência em DDS (Participantes F a J)

Questão	Participante F (TL, Lisboa)	Participante G (PO, Campinas)	Participante H (QA, Madri)	Participante I (DevOps, BH)	Participante J (AN, Lisboa)
Experiência prévia em DDS?	Sim	Sim	Sim	Sim	Sim
Maior desafio do DDS?	Sincronizar testes/Diferença de fuso	PO presente em todas as cerimônias	Disciplina/Clareza/Estilo de comunicação	Concepção de Design (Distância)	Reduzir atrito entre entregas

Adaptações para fusos?	Automação de testes	Priorização global (Jira/Miro)	Testes assíncronos/Relatórios automáticos	Sessões colaborativas (Miro/Figma)	CI/CD, Automação de pipelines
------------------------	---------------------	--------------------------------	---	------------------------------------	-------------------------------

Tabela 12.3: Síntese de Experiência em DDS (Participantes K a O)

Questão	Participante K (Dev, Campinas)	Participante L (SM, Lisboa)	Participante M (UX, Campinas)	Participante N (QA, Madrid)	Participante O (PM, Lisboa)
Experiência prévia em DDS?	Sim	Sim	Sim	Sim	Sim
Maior desafio do DDS?	Clareza dos requisitos/ Documentação técnica	Controle de riscos/Previsibilidade	Paciência/Comunicação/Revisões	Documentação arquitetural	Colaboração multidisciplinar
Adaptações para fusos?	Comunicação estruturada/Empática	Cadência Scrum/Dashboards Jira	Revisões de código/Pareamento remoto	Confluence/PlantUML (Registro técnico)	Cerimônias como pontos de convergência

4.2.3 Implementação do Scrum

As cerimônias do *Scrum* foram mantidas, porém adaptadas ao contexto remoto:

Tabela 13 - Adaptações do Scrum ao projeto

Cerimônia	Adaptação
<i>Planning</i>	2h via videoconferência, documentação detalhada das <i>user story</i>
<i>Daily</i>	15 minutos, ajustados ao fuso horário, webcams obrigatórias
<i>Review</i>	Demonstração de funcionalidades com compartilhamento de tela
Retrospective	Ferramentas colaborativas online para votação de ações de melhoria

O *Scrum Master* destacou que a disciplina nas cerimônias é o principal elemento para manter o foco do time, mesmo à distância. O uso de *webcams*, microfones abertos e quadros digitais (Miro, Mural) aumentou a sensação de presença. Além disso, o

Product Owner relatou que a transparência visual do *backlog* no *Jira* fortalece a confiança e a rastreabilidade das decisões.

Ferramentas como *Jira*, *Confluence*, *Slack* e *Zoom* garantem transparência, comunicação e registro de decisões, confirmando a importância do suporte tecnológico em DDS (Borges, 2012). Além disso, os participantes relataram a criação de “momentos informais” virtuais para reforçar vínculos interpessoais, prática que tem se mostrado eficaz para reduzir o isolamento típico do trabalho remoto.

4.2.4 Síntese de Adaptações e Eficácia das Cerimônias *Scrum*

Este tópico sintetiza a percepção dos participantes sobre a implementação, a adaptação das cerimônias e a avaliação geral de eficácia do método *Scrum* em contextos de times distribuídos.

- Conforme as informações apresentadas abaixo, as cerimônias como a *Daily* e a *Planning* exigem adaptações específicas para o contexto distribuído
- Segundo a Tabela 14.1, as adaptações focam em horários intermediários e uso de quadros digitais para planejamento coletivo
- De acordo com a Tabela 14.2 e a Tabela 14.3, o método *Scrum* é considerado eficaz por garantir ritmo, transparência e reduzir retrabalho através de uma estrutura disciplinada

Tabela 14.1: Adaptação de Cerimônias (Entrevistas A-E)

Questão	Participante A (SM, Lisboa)	Participante B (Dev, BH)	Participante C (PO, Campina)	Participante D (Dev, Lisboa)	Participante E (QA, Campina)
Adaptação da Daily?	15min, horário intermediário; Webcam obrigatória	Disciplina, manter foco	Essencial para manter ritmo e alinhamento	Essencial para alinhamento remoto	Disciplina rigorosa
Adaptação da Planning?	Videoconferência, documentação	Foco na clareza e ritmo	Planejamento coletivo	Uso de quadros digitais (Miro)	Foco na padronização de código

Scrum é eficaz?	Sim; Estrutura mínima e flexível	Sim; Garante ritmo sustentável	Sim; Fluxo natural das entregas	Sim; Mantém alinhamento/ciclo inspeção	Sim; Aumenta a confiança na entrega
-----------------	----------------------------------	--------------------------------	---------------------------------	--	-------------------------------------

Tabela 14.2: Adaptação de Cerimônias e Ferramentas (Entrevistas F-J)

Questão	Participante F (TL, Lisboa)	Participante G (PO, Campinas)	Participante H (QA, Madri)	Participante I (Devops, BH)	Participante J (AN, Lisboa)
Adaptação da Daily?	Disciplina é o principal	Cadência mantém a conexão	Ajuda a compensar fuso	Deve ser concisa e focada	Aumenta visibilidade técnica
Adaptação da Planning?	Definição clara dos critérios de aceite	Priorização global no backlog	Exige clareza de requisitos	Integração do design	Alinhamento com a infraestrutura (DevOps)
Ferramentas cruciais?	Slack, Zoom (Equilíbrio de fuso)	Jira, Miro (Coerência do backlog)	Jira, Confluence (Rastreamento)	Miro, Figma (Colaboração Design)	CI/CD, Jira (Visibilidade técnica)
Scrum é eficaz para DDS?	Sim; Reduz retrabalho/Aumenta confiança	Sim; Disciplina mantém conexão	Sim; Traz clareza e previsibilidade	Sim; Ajuda a manter o foco nas pessoas	Sim; Potencializado pela cultura DevOps

Tabela 14.3: Adaptação de Cerimônias e Ferramentas (Entrevistas K-O)

Questão	Participante K (Dev, Campinas)	Participante L (SM, Lisboa)	Participante M (UX, Campinas)	Participante N (QA, Madrid)	Participante O (PM, Lisboa)
Adaptação da Daily?	Reduz ambiguidade/Apoia comunicação	Essencial para controle de riscos	Aumenta o foco técnico	Valida continuamente a arquitetura	Reforça o senso de unidade
Adaptação da Planning?	Clareza dos requisitos	Favorece acompanhamento gerencial	Ajuda a evitar dispersão	Define o ritmo para a validação	Pontos de convergência entre culturas

Ferramentas cruciais?	Zoom, Confluence	Jira Dashboards	Revisão de Código, Pareamento	Confluence, PlantUML	Ferramentas colaborativas (Miro/Figma)
<i>Scrum</i> é eficaz para DDS?	Sim; Depende da comunicação estruturada	Sim; Garante previsibilidade/Transparência	Sim; Oferece estrutura disciplinada	Sim; Permite inspeção e adaptação	Sim; Fortalece a colaboração multidisciplinar

4.2.5 Comunicação, colaboração e cultura

A comunicação síncrona e assíncrona equilibrada garante agilidade e registro histórico. Diferenças culturais sutis influenciam estilos de comunicação, mitigadas por empatia e feedback contínuo. O QA (Participante E) mencionou que, em situações de conflito de prioridades, o time adota o princípio da “presunção positiva”, assumindo que o outro está agindo de boa-fé. Essa prática cultural reduz tensões e favorece o clima de colaboração.

“Sabemos que podemos confiar nas entregas porque todas estão visíveis e auditáveis no Jira.” (Participante D)

Além disso, os participantes destacaram que o *Scrum Master* atua como mediador cultural, facilitando o entendimento entre os diferentes contextos regionais. Essa mediação é essencial para reduzir ruídos e alinhar expectativas.

O Participante G (*Product Owner*, Campinas) complementou que, em reuniões de planning ou refinamento, o *Scrum Master* frequentemente reformula expressões e exemplos de cada país para garantir que todos compreendam o mesmo significado das histórias de usuário. Segundo ele, “não é raro que uma palavra tenha conotações diferentes entre o português do Brasil e o de Portugal, e o *Scrum Master* ajuda a traduzir esse contexto cultural”.

O Participante H (QA, Madrid) reforçou a importância de evitar suposições culturais precipitadas, destacando que “na Espanha, tende-se a usar uma comunicação mais assertiva, mas sem a mesma carga emocional; no início, os colegas brasileiros interpretavam como frieza”. Com o tempo, o time aprendeu a diferenciar estilo de personalidade de intenção, o que fortaleceu o vínculo profissional.

Vários entrevistados também mencionaram a criação de rituais informais, como cafés virtuais quinzenais e canais de conversas livres no *Slack*, que funcionam como espaços de socialização e descontração. Esses momentos não são apenas recreativos, mas desempenham papel importante na manutenção da coesão emocional entre membros geograficamente dispersos. Conforme o Participante B (Desenvolvedor *Backend*, Belo Horizonte) relatou, “esses cafés ajudam a lembrar que estamos trabalhando com pessoas, não apenas telas”.

O Participante J (Analista de Negócios, Lisboa) destacou que a colaboração só é efetiva quando existe clareza nas intenções e abertura para o diálogo: “Quando há um mal-entendido, nós o tratamos abertamente na retrospectiva; esconder desconfortos é mais prejudicial que o próprio conflito”. Essa prática de feedback transparente nas cerimônias *Scrum* foi apontada como um dos pilares do clima de confiança.

Outro aspecto recorrente nas entrevistas foi a influência do idioma. Embora o português e o inglês sejam o idioma predominante, o uso ocasional do inglês técnico e a exposição a sotaques distintos exigem atenção redobrada em reuniões. O Participante M (UX, Campinas) comentou que “a escrita padronizada no Jira é uma forma de neutralizar ambiguidades linguísticas; todos consultam a mesma fonte de verdade”. Essa padronização textual reforça a comunicação assíncrona, reduzindo dependência de interações síncronas.

O uso equilibrado de ferramentas também se mostrou decisivo para o sucesso da comunicação. Ferramentas síncronas (*Zoom*, *Teams*) são usadas para cerimônias e alinhamentos rápidos, enquanto as assíncronas (*Jira*, *Confluence*, *Slack*) concentram registros e decisões. O Participante F (*QA Engineer*, Belo Horizonte) enfatizou que “tudo que é decidido oralmente é registrado no *Confluence* em até 24 horas; isso evita

confusões entre turnos de trabalho”. Essa prática institucionaliza o aprendizado coletivo e reduz a perda de contexto entre os fusos horários.

Alguns participantes reforçaram que a comunicação não é apenas operacional, mas um elemento cultural e emocional. Segundo o Participante I (*DevOps*, Belo Horizonte), “a confiança entre pessoas em diferentes países não acontece por acaso; ela é construída quando todos têm visibilidade do trabalho e do impacto das suas ações no entregável final”.

O Participante J (Analista de Negócios, Lisboa) relatou que a clareza documental tornou-se elemento-chave para evitar retrabalho: “Aprendemos que é melhor gastar mais tempo descrevendo bem um requisito do que perder o dobro reconstruindo algo que foi mal interpretado”.

A presença de membros espanhóis trouxe discussões mais diretas nas Retrospectives, o que inicialmente gerou desconfortos para brasileiros, mas foi ajustado com acordos explícitos de comunicação.

Conforme Borges (2012), diferenças culturais refletem diretamente no estilo de negociação e feedback. No caso estudado, essas diferenças foram mediadas pelo *Scrum Master* (Participante L), que assumiu o papel de facilitador cultural, algo não explicitado no framework original, mas necessário na prática.

Culturalmente, observou-se que os brasileiros tendem a uma comunicação mais informal e relacional, os portugueses a uma abordagem mais direta e pragmática, e os espanhóis a um estilo mais assertivo e estruturado. Apesar dessas diferenças, todos relataram uma progressiva convergência para um “idioma ágil comum”, caracterizado por objetividade, empatia e foco em valor. Essa evolução linguística e comportamental é um exemplo prático de aculturação ágil dentro de um time distribuído.

O *Scrum Master* (Participante A) sintetizou esse processo afirmando: “No início, cada país tinha seu próprio ritmo e tom de voz; agora falamos a mesma língua ágil, priorizamos o cliente e a entrega, não a hierarquia”. Essa fala reflete a internalização dos

valores do Manifesto Ágil (Beck et al., 2001), em especial a valorização de indivíduos e interações acima de processos e ferramentas.

Por fim, as entrevistas revelaram que a cultura de confiança e transparência estabelecida no time permitiu que as diferenças deixassem de ser barreiras e se tornassem fonte de aprendizado. A colaboração multicultural foi associada a maior inovação, criatividade e sensibilidade na solução de problemas. O Participante J (Analista de Negócios, Lisboa) observou que “a troca de perspectivas culturais enriquece as discussões de design, porque cada um traz um olhar distinto sobre o usuário”.

Portanto, conclui-se que a comunicação e a colaboração em equipes distribuídas da *Lean Software* são sustentadas por três pilares: (1) mediação cultural ativa exercida pelo *Scrum Master*; (2) práticas de documentação e transparência digital que reduzem ambiguidade; e (3) uma cultura organizacional de empatia e *feedback* constante. Esses elementos transformam a diversidade geográfica e cultural em vantagem competitiva, reforçando o alinhamento com os princípios do *Scrum*.

As tabelas apresentadas a seguir sintetizam as percepções dos participantes sobre a gestão da diversidade cultural, estilos de comunicação e práticas de colaboração no contexto de equipes distribuídas.

- A Tabela 15.1 detalha as visões dos Participantes A a E, destacando a importância da documentação e o papel do *Scrum Master* como mediador cultural
- A Tabela 15.2 reúne as perspectivas dos Participantes F a J, focando na transparência visual e no uso de ferramentas de automação para reduzir atritos
- A Tabela 15.3 encerra a síntese com os Participantes K a O, enfatizando a clareza documental e a integração entre as diferentes áreas de especialidade

Tabela 15.1: Perspectivas sobre Colaboração e Cultura (Participantes A a E)

Questão	Participante A (SM, Lisboa)	Participante B (Dev, BH)	Participante C (PO, Campinas)	Participante D (Dev, Lisboa)	Participante E (QA, Campinas)

Comunicação Síncrona vs Assíncrona?	Ambas essenciais	Gravações de reuniões e canais assíncronos	Documentação clara é crucial	Tudo visível e auditável no Jira	Feedback contínuo
Gestão da Diversidade Cultural?	<i>Scrum Master</i> como mediador cultural	Confiança construída pela visibilidade do trabalho	<i>Scrum Master</i> reformula expressões (Português/Espanhol)	Presunção positiva e confiança	Liderança situacional/Confiança nas pessoas
Práticas de Colaboração Informal?	Momentos informais virtuais	Cafés virtuais quinzenais (Socialização)	Uso intensivo de Miro e Mural	Retrospectiva como ajuste emocional e técnico	Mentoria técnica periódica
Maior Fator de Sucesso?	Falamos a mesma "Língua Ágil"	Equilíbrio Síncrono/Assíncrono	Clareza na intenção e abertura ao diálogo	Transparência no Jira/Confluence	Padrões de código unificados

Tabela 15.2: Perspectivas sobre Colaboração e Cultura (Participantes F a J)

Questão	Participante F (TL, Lisboa)	Participante G (PO, Campinas)	Participante H (QA, Madri)	Participante I (DevOps, BH)	Participante J (AN, Lisboa)
Comunicação Síncrona vs Assíncrona?	Decisões orais registradas no Confluence (24h)	Transparência visual do backlog (Jira)	Uso de webcams e microfones abertos	Empatia é o maior elo (Design/Dev)	CI/CD reduz atrito comunicacional
Gestão da Diversidade Cultural?	Maior sensibilidade cultural do time	PO/SM traduzem contexto cultural	Diferenciar estilo de personalidade de intenção	Troca de perspectivas enriquece design	Comunicação constante e cooperação
Práticas de Colaboração Informal?	Canais de conversas livres (Slack)	Foco na priorização global	Acordos explícitos de comunicação	Sessões colaborativas (Miro/Figma)	Automação e Integração Contínua

Maior Fator de Sucesso?	Automação de testes remota	Disciplina nas cerimônias	Clareza na comunicação (evitar suposições)	Foco nas pessoas	Visibilidade das tarefas técnicas
-------------------------	----------------------------	---------------------------	--	------------------	-----------------------------------

Tabela 15.3: Perspectivas sobre Colaboração e Cultura (Participantes K a O)

Questão	Participante K (Dev, Campinas)	Participante L (SM, Lisboa)	Participante M (UX, Campinas)	Participante N (QA, Madrid)	Participante O (PM, Lisboa)
Comunicação Síncrona vs Assíncrona?	Clareza dos requisitos depende de comunicação empática	Uso combinado de ferramentas (Jira/Zoom)	Escrita padronizada no Jira (Neutraliza ambiguidades)	Arquitetura como linguagem comum	Cerimônias como pontos de convergência cultural
Gestão da Diversidade Cultural?	Conforto em tratar mal-entendidos abertamente	<i>Scrum Master</i> como mediador cultural	Foco em valor (Objetividade)	Documentação arquitetural	Abertura para diálogo (Assertividade vs Frieza)
Práticas de Colaboração Informal?	Documentação técnica remoto estruturado	Dashboards no Jira	Revisões de código e pareamento remoto	Uso de PlantUML para compartilhamento	Criação de "ritmo natural de troca"
Maior Fator de Sucesso?	Clareza documental (Evitar retrabalho)	Previsibilidade e transparência	Paciência e comunicação	<i>Scrum</i> fornece o ritmo para validação	Integração entre Design, Dev e Produto

4.2.6 Perfil e percepções dos participantes

O grupo de entrevistados foi composto por quinze profissionais da empresa *Lean Software*, atuando em diferentes localidades Lisboa, Belo Horizonte, Campinas e Madri e ocupando papéis complementares dentro do *framework Scrum*, como *Scrum Masters*, *Product Owners*, desenvolvedores, engenheiros de qualidade, líderes técnicos, designers e gestores. Essa diversidade de funções e experiências proporcionou uma

visão ampla sobre a implementação e adaptação do *Scrum* em contextos de desenvolvimento distribuído de software (DDS).

O Participante A, *Scrum Master* localizado em Lisboa, possui cinco anos de experiência com *Scrum* (CSM) e atuação em três localidades simultâneas. Ele destacou que o principal desafio é “manter o engajamento do time e o ritmo das cerimônias, mesmo com a distância”. O time adaptou os horários das reuniões e tornou obrigatório o uso de webcams nas *dailies* para fortalecer a comunicação visual. Segundo o entrevistado, “o *Scrum* é a estrutura mínima, mas flexível, que mantém a equipe alinhada”, ressaltando o papel das retrospectivas participativas como principal fator de sucesso.

O Participante B, *Desenvolvedor Backend* em Belo Horizonte, relatou quatro anos de experiência com *Scrum* e atuação prévia em equipes distribuídas. Ele destacou o uso de Slack e Zoom, ferramentas que garantem transparência e confiança nas entregas. Na sua percepção, “o *Scrum* garante ritmo sustentável e clareza de prioridades”, permitindo que todos acompanhem o progresso das tarefas e a evolução do backlog.

O Participante C, *Product Owner* sediado em Campinas, com seis anos de experiência, enfatizou que “quando o planejamento é coletivo, as entregas fluem naturalmente”. Ele ressaltou a importância da documentação clara e da documentação técnica detalhada, práticas que asseguram coesão e entendimento das histórias de usuário, promovendo alinhamento entre as equipes.

O Participante D, *Desenvolvedor Frontend* em Lisboa, com quatro anos de experiência, apontou que a coordenação do *Scrum Master* e o uso de Jira e Confluence foram determinantes para o sucesso. Em suas palavras, “o *Scrum* é essencial para manter o alinhamento remoto”, uma vez que estabelece um ciclo constante de inspeção e adaptação.

O Participante E, *Engenheiro de Qualidade (QA)* em Campinas, com seis anos de experiência, destacou que a automação de testes remota e o uso de Slack e Zoom foram essenciais para equilibrar os fusos horários. Conforme relatou, “a automatização foi

essencial para sincronizar testes entre fusos horários distintos”. Ele observou que o foco na qualidade contínua reduziu retrabalho e aumentou a confiança entre times.

O Participante F, *Tech Lead* também em Lisboa, possui nove anos de experiência (PSM I) e destacou a dificuldade de exercer liderança à distância sem recorrer ao controle excessivo. Ele afirmou que “o maior desafio é liderar à distância sem controlar; é preciso confiar nas pessoas”. Para garantir qualidade e padronização, implementou padrões de código unificados e manteve reuniões de mentoria técnica periódicas, reforçando o conceito de liderança situacional.

O Participante G, *Product Owner* de Campinas, com oito anos de experiência, reforçou a relevância da disciplina das cerimônias e da clareza dos papéis. Para ele, “sem um PO presente em todas as cerimônias, o *Scrum* perde o ritmo”. Utilizando Jira e Miro, ele enfatizou que a priorização global do *backlog* é essencial para garantir coerência entre diferentes frentes de trabalho.

O Participante H, *QA Engineer* de Madri (Espanha), com cinco anos de experiência, relatou que “trabalhar com diferentes países exige disciplina e clareza; as diferenças são mais de estilo do que de competência”. O time adotou testes assíncronos e relatórios automáticos para contornar defasagens de horário, assegurando rastreamento das atividades e comunicação estruturada.

O Participante I *DevOps Engineer* localizado em Belo Horizonte, com seis anos de experiência, ressaltou que a integração contínua e a automação de pipelines são fundamentais para o sucesso em DDS. Ele afirmou: “a cultura DevOps potencializa o *Scrum*, porque reduz o atrito entre entregas e infraestrutura”. O entrevistado destacou que o *Scrum* amplia a visibilidade das tarefas técnicas, frequentemente invisíveis em modelos tradicionais.

O Participante J, *Analista de Negócios* em Lisboa, com sete anos de experiência, apontou que a Documentação técnica remoto é uma das fases mais críticas. Para ele, “a clareza dos requisitos depende de uma comunicação estruturada e empática”,

ressaltando a importância do *Product Owner* em traduzir a visão do cliente em histórias compreensíveis para os desenvolvedores.

O Participante K, *Desenvolvedora Full Stack* em Lisboa, com cinco anos de experiência, destacou a relevância das revisões de código e do pareamento remoto. Para ele, “trabalhar à distância exige mais paciência e comunicação; revisões frequentes reduzem retrabalho e aumentam a qualidade”. O entrevistado observou que o *Scrum* oferece estrutura disciplinada, que evita dispersão de prioridades e melhora o foco técnico.

O Participante L, *Scrum Master* em Lisboa, com oito anos de experiência, enfatizou que o *Scrum* garante previsibilidade e transparência mesmo em equipes distribuídas. Em sua análise, “a cadência do *Scrum* ajuda a controlar riscos mesmo quando os times estão em diferentes países”. Ele ressaltou que o uso combinado de dashboards no Jira e reuniões de revisão favorece o acompanhamento gerencial e estratégico.

O Participante M, *UX Designer* baseado em Campinas, com seis anos de experiência, observou que a distância afeta principalmente as fases de concepção de design. Para mitigar o impacto, a equipe passou a realizar sessões colaborativas via Miro e Figma, integrando o design às demais etapas do desenvolvimento. Segundo ele, “a empatia é o maior elo entre design e desenvolvimento; o *Scrum* ajuda a manter o foco nas pessoas, mesmo à distância”.

O Participante N, *Arquiteto de Software* em Lisboa, com nove anos de experiência, afirmou que a documentação arquitetural é o elo entre as equipes distribuídas. Segundo ele, “a arquitetura atua como linguagem comum, e o *Scrum* fornece o ritmo para validá-la continuamente”. O profissional destacou a importância de ferramentas como Confluence e PlantUML para registro e compartilhamento de decisões técnicas.

Por fim, o Participante O, Gestor de Projeto localizado em Lisboa, com dez anos de experiência, salientou o impacto positivo do *Scrum* na colaboração multidisciplinar. Em suas palavras, “o *Scrum* aproxima design, desenvolvimento e produto, criando uma

cadência natural de troca”. O entrevistado destacou que as cerimônias, especialmente as *reviews*, funcionam como pontos de convergência entre culturas e linguagens distintas, reforçando o senso de unidade do time.

4.2.7 Síntese interpretativa

A análise das entrevistas evidencia que o sucesso do *Scrum* em ambientes distribuídos depende da integração entre aspectos técnicos e humanos. A confiança mútua, a comunicação estruturada, o uso consistente de ferramentas colaborativas e a clareza dos papéis emergem como pilares fundamentais. As falas dos participantes corroboram as conclusões de Borges (2012), ao destacar que a comunicação é o fator central para a eficácia de equipes distribuídas, e de Prikladnicki e Audy (2002), que enfatizam a necessidade de coordenação e visibilidade contínua nos projetos de DDS. Dessa forma, observa-se que o *Scrum*, quando bem adaptado, atua não apenas como uma metodologia de gestão, mas como um mecanismo de integração cultural e técnica entre equipes dispersas geograficamente.

Fonte: Dados da pesquisa (2025).

4.3 Síntese dos Resultados

A análise integrada das observações do time, das entrevistas com os quinze participantes e da literatura especializada permite afirmar que a *Lean Software* consolidou um modelo de operação distribuída maduro, estável e alinhado às boas práticas internacionais de DDS (Desenvolvimento Distribuído de Software). Os resultados qualitativos obtidos confirmam tendências amplamente discutidas na literatura, ao mesmo tempo em que revelam especificidades do contexto organizacional estudado.

Inicialmente, observou-se que a empresa alcançou níveis elevados de transparência, coordenação estruturada e disciplina comunicacional, aspectos cruciais para o funcionamento de equipes distribuídas. Essa constatação dialoga diretamente com os achados de Borges (2012), que destaca a visibilidade contínua das tarefas e a

clareza de responsabilidades. O uso sistemático de ferramentas como *Jira*, *Confluence*, *Slack*, *Zoom* e *Miro* demonstrou que a *Lean Software* estruturou um ambiente digital capaz de sustentar fluxos de comunicação síncronos e assíncronos, garantindo rastreabilidade e registro histórico das decisões.

Além disso, verificou-se que as práticas do *Scrum* funcionam como um mecanismo de estabilização organizacional, reduzindo a variabilidade de comportamento entre as equipes distribuídas. As cerimônias *Planning*, *Daily*, *Review* e *Retrospective* atuam como pontos de ancoragem que compensam a ausência de interação presencial, reforçando um senso de ritmo, alinhamento e inspeção contínua. Esse resultado reforça os argumentos de Prikladnicki e Audy (2002), segundo os quais *frameworks* ágeis oferecem estrutura suficiente para lidar com a dispersão geográfica, desde que adaptados às necessidades locais do time.

Conforme evidenciado na Tabela 14.1 (Síntese de Adaptações e Eficácia das Cerimônias *Scrum*), a totalidade dos participantes indicou que a cadência das cerimônias aumenta o alinhamento, diminui a ambiguidade e fortalece a coesão da equipe. Para muitos dos entrevistados, sobretudo *Product Owners* e *Scrum Masters*, o *Scrum* funciona como o “fio condutor” do trabalho, assegurando que todos compreendam prioridades e restrições, independentemente da localização física. Estudos recentes colaboram com essa experiência: Shahzad et al. (2023) destacam que equipes distribuídas com rituais estruturados e comunicação frequente apresentam maior estabilidade emocional e maior percepção de eficácia.

Outro ponto recorrente na análise é o papel da diversidade cultural. A integração de profissionais localizados em Portugal, Brasil e Espanha ampliou o repertório cognitivo do grupo e permitiu a construção de soluções mais criativas, fruto de perspectivas distintas e complementares. Esse fenômeno é discutido amplamente por Noll et al., (2020), que argumentam que a heterogeneidade cultural, quando bem mediada, gera inovação e aumenta a maturidade organizacional. Nos depoimentos colhidos, os participantes destacaram que a diversidade influencia positivamente a qualidade das

discussões, reduz vieses de decisão e estimula abordagens alternativas para problemas complexos.

Contudo, a diversidade também introduz desafios significativos. A análise evidencia que diferenças linguísticas sutis, estilos divergentes de comunicação e expectativas culturais distintas em relação a prazos e formalidades exigem mediação ativa por parte do *Scrum Master* e da liderança técnica. Essa mediação é descrita pelos entrevistados como um fator crítico para que o time não apenas “funcione”, mas atinja um nível elevado de colaboração. Borges (2012) aponta que conflitos e mal-entendidos culturais são inevitáveis em DDS, mas podem ser mitigados por meio de *feedback* contínuo, empatia e práticas deliberadas de integração, exatamente como observado no caso analisado. A Tabela 14.1 (Síntese Adaptações e Eficácia das Cerimônias *Scrum*) ilustra que o *Scrum Master* (Participante A) e outros líderes atuam ativamente como mediadores culturais, facilitando o entendimento entre os contextos de Portugal, Brasil e Espanha.

Os efeitos do fuso horário representam outro eixo relevante. Ainda que a diferença entre Portugal, Brasil e Espanha seja relativamente pequena quando comparada a cenários globais amplos, os participantes relataram a necessidade de ajustes finos nos horários de cerimônias e na organização do trabalho assíncrono. As experiências descritas reforçam a tese de Prikladnicki e Audy (2002) de que a sobreposição mínima de horários constitui um fator crítico para a eficácia do DDS, especialmente em tarefas que exigem tomada de decisão coletiva. Como resumido na Tabela 12.2 (Síntese de Experiência em DDS), as principais estratégias de mitigação foram a automação de testes (Participantes F e H) e a adoção de horários intermediários para as cerimônias síncronas.

A investigação também ressaltou a importância da autonomia e da motivação intrínseca dos membros do time. Os resultados confirmam estudos recentes que apontam que equipes remotas de alta performance compartilham níveis elevados de autorregulação e senso de responsabilidade coletiva (Shahzad et al., 2023). Na *Lean Software*, observou-se que a autonomia é sustentada pelo equilíbrio entre liberdade

operacional e processos bem definidos, o que cria um ambiente seguro para experimentação e melhoria contínua.

Outro achado importante diz respeito à complementaridade entre ferramentas e práticas humanas. Embora a empresa utilize um conjunto abrangente de tecnologias de suporte, como painéis *Kanban* acessíveis em tempo real, automação de testes, *pipelines DevOps* e documentação colaborativa, os entrevistados destacaram que a tecnologia somente se torna eficaz quando acompanhada de práticas humanas de escuta ativa, flexibilidade e cooperação. Esse alinhamento entre técnica e cultura confirma o argumento de Beecham et al. (2022) de que a tecnologia é um facilitador, mas não substitui a necessidade de relações interpessoais saudáveis.

Tabela 16 - Síntese dos Pilares do Sucesso do *Scrum* em DDS na *Lean Software*

Dimensão de Sucesso	Elementos Chave Observados (Práticas)	Suporte Teórico Confirmado
Estrutura	Cadência rigorosa das Cerimônias <i>Scrum</i> (Dailys, Reviews) e Papéis claros (PO, SM).	Previsibilidade e Inspeção Contínua Schwaber & Beedle(2002).
Tecnologia	Uso intensivo de ferramentas assíncronas (Jira, Confluence) e automação de testes/DevOps.	Transparência digital e Rastreabilidade (Borges, 2012); Tecnologia como Facilitador (Beecham et al., 2022).
Cultura	Mediação cultural ativa do <i>Scrum</i> Master, Empatia Intercultural e “Presunção Positiva”.	Mitigação de Conflitos Culturais (Borges, 2012); Geração de Inovação pela Heterogeneidade (Noll, Razzak & Beecham, 2020).

Por fim, a análise global dos resultados permite concluir que a *Lean Software* obteve um modelo distribuído eficiente porque conseguiu integrar três dimensões fundamentais, conforme sumarizado na Tabela 16:

- Estrutura: processos claros, rituais de *Scrum*, padrões de documentação e governança.

- Tecnologia: ferramentas de suporte adequadas, automação disciplinada e ambientes colaborativos.
- Cultura: confiança, empatia, comunicação transparente e liderança mediadora.

Essa combinação confirma e reforça a literatura revisada, mostrando que equipes distribuídas de alta maturidade dependem tanto de fatores técnicos quanto de fatores humanos. Assim, a síntese dos resultados demonstra que a empresa estudada não apenas adota práticas recomendadas, mas também internaliza rotinas de colaboração que favorecem desempenho, previsibilidade e satisfação da equipe.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo analisar os desafios enfrentados por equipes distribuídas de desenvolvimento de software e compreender como a metodologia ágil *Scrum* pode apoiar a colaboração, a comunicação e a entrega de valor nesse contexto. Para isso, foi realizado um estudo de caso aprofundado na empresa *Lean Software*, envolvendo análise documental, observação dos processos internos e entrevistas com profissionais distribuídos entre Portugal (Lisboa), Brasil (Campinas e Belo Horizonte) e Espanha (Madrid). A partir dessa investigação, foi possível identificar práticas, dificuldades, adaptações e fatores de sucesso que caracterizam a atuação de um time ágil distribuído em múltiplas localidades geográficas e culturais.

Os resultados obtidos reforçam que trabalhar com equipes distribuídas exige não apenas domínio técnico, mas sobretudo maturidade organizacional, disciplina comunicacional e um conjunto sólido de práticas de colaboração. Ao longo do estudo verificou-se que a *Lean Software* desenvolveu, ao longo dos anos, um ecossistema próprio de trabalho remoto, apoiado tanto em ferramentas digitais quanto em um conjunto de valores e rituais alinhados ao *Scrum*. Essa combinação foi decisiva para garantir a coordenação eficiente entre profissionais localizados em fusos distintos e com níveis diferentes de senioridade, formação e carga cultural.

5.1 Conclusão Geral do Estudo

Considerando que o objetivo deste trabalho foi descrever os desafios de um time distribuído em diferentes cidades e países, e demonstrar como é possível realizar entregas de qualidade mantendo uma comunicação eficiente por meio da metodologia ágil *Scrum*, pode-se concluir que esse propósito foi plenamente alcançado. O estudo revelou que o *Scrum* desempenhou um papel central na coordenação do projeto analisado, atuando como uma estrutura mínima, porém sólida, capaz de organizar o fluxo de trabalho, facilitar alinhamentos e promover a inspeção contínua.

Após análise do estudo de caso, constatou-se que a metodologia *Scrum* foi uma grande aliada para o sucesso do projeto. Os “ritos” *Sprint Planning*, *Daily Meetings*, *Sprint Review* e *Sprint Retrospective*, foram conduzidos de forma rigorosa e se mostraram fundamentais para manter o time sincronizado. As reuniões diárias favoreceram a transparência e o acompanhamento do progresso; os planejamentos de *sprint* permitiram clarificar responsabilidade e prioridades; e os alinhamentos contínuos garantiram que todos compreendessem exatamente o estágio do projeto, mesmo atuando a milhares de quilômetros de distância.

As entrevistas reforçam um ponto amplamente discutido na literatura: equipes distribuídas dependem fortemente de cadência e clareza para evitar ruído e desalinhamento (Stray & Moe, 2020; Noll et al., 2020). Assim como apontam Borges (2012) e Prikladnicki e Audy (2002), o trabalho remoto intensifica problemas típicos de comunicação, atrasos, ambiguidade, falta de contexto exigindo mecanismos formais de coordenação. O *Scrum* se mostrou efetivo justamente por oferecer rituais estruturados que reduzem essas lacunas.

Outro ponto de destaque foram as reuniões de retrospectiva, que se evidenciaram como um mecanismo essencial de aprendizagem contínua. Por meio delas, o time pôde identificar rapidamente problemas, propor planos de ação e ajustar o processo colaborativo ao longo das semanas. Em alguns momentos, essas retrospectivas levaram a decisões estratégicas, como mobilizar parte da equipe a uma mesma base presencial

para reforçar laços e diminuir barreiras relacionais ação que aumentou a empatia, o senso de pertencimento e a fluidez da comunicação.

O projeto analisado também passou por inúmeras situações de incerteza e alteração de escopo, cenário comum no desenvolvimento de software e enfatizado por autores ágeis como Schwaber & Beedle(2002). Nesses momentos, a união do time e a comunicação franca tornaram-se essenciais. Reuniões extraordinárias de planejamento foram realizadas com maior frequência para garantir que todos compreendessem o impacto das mudanças e conseguissem reagir de forma coordenada.

Esse aspecto tornou explícita a dimensão social do trabalho ágil: desenvolver software não é apenas executar tarefas técnicas, mas principalmente desenvolver pessoas, apoiá-las, acolhê-las e criar condições para que exerçam seu potencial em ambientes complexos e de alta pressão.

Por outro lado, o trabalho remoto também apresentou desafios significativos de adaptação. Foi necessária uma mudança radical na forma de trabalhar: o “script” do *Scrum* foi seguido de maneira rigorosa porque era o principal mecanismo para garantir a fluidez da comunicação. A adaptação no modo de se expressar de forma remota não foi algo imediato; exigiu esforço, disciplina e paciência de todos os envolvidos. Os líderes: *Scrum Master*, *Product Owner* e Arquiteto de Software, tiveram papel determinante nesse processo, atuando como facilitadores, mediadores e mantenedores da motivação coletiva.

Assim, conclui-se que o *Scrum*, por ser uma metodologia centrada nas pessoas e guiada pelos valores de foco, coragem, franqueza, compromisso e respeito, é particularmente adequado para times distribuídos. Esses valores, quando observados, tornam-se um antídoto natural para as dificuldades impostas pela distância geográfica, garantindo que a comunicação e a clareza das expectativas permaneçam como pilares do trabalho, independentemente da localização física dos membros da equipe.

5.1.1 Considerações Específicas sobre o Caso da *Lean Software*

A análise específica da *Lean Software* permitiu identificar características que diferenciam sua atuação no contexto remoto:

1. Distribuição geográfica altamente assimétrica, com concentração de funções específicas por base:
 - a. Lisboa: *Scrum Masters*, *Tech Lead*, *Frontend*, Analista de Negócios, Gestor de Projeto
 - b. Campinas: *Product Owners*, QA, UX, *Full Stack*
 - c. Belo Horizonte: *Backend* e DevOps
 - d. Madrid: QA Sênior e Designer de Produto
2. Modelo de comunicação híbrido (síncrono + assíncrono), alinhado ao que Stray e Moe (2020) identificam como fator crítico de sucesso em equipes globais.
3. Uso intensivo de ferramentas (Jira, Confluence, Slack, Zoom, Miro), que reforçam a visibilidade das atividades e minimizam incertezas, conforme também indicado por Shahzad et al. (2023).
4. Adaptações específicas do *Scrum*, como:
 - a. Reuniões técnicas conduzidas pelo Arquiteto
 - b. *Sprint Review* intermediária (mid-sprint check)
 - c. Encontros presenciais estratégicos para reforçar vínculos
5. Forte cultura de empatia, especialmente relevante em contextos com profissionais com necessidades específicas de comunicação.

Esses pontos demonstram que o sucesso do time não dependeu apenas do *framework Scrum* em si, mas de um ecossistema organizacional coerente, que inclui cultura, ferramentas, disciplina e práticas alinhadas aos valores ágeis.

5.1.2 Considerações Finais Ampliadas

Com base no que foi observado, este estudo contribui para o entendimento de como equipes distribuídas podem alcançar alta performance mesmo enfrentando desafios como distância geográfica, diferenças culturais, fusos horários e diversidade de perfis profissionais. A experiência da *Lean Software* mostra que:

- Times distribuídos podem ser tão eficientes quanto times co-localizados.
- A distância não impede a colaboração, desde que existam estrutura, disciplina e empatia.
- O *Scrum*, quando aplicado com rigor e flexibilidade, oferece mecanismos fortes para preservar comunicação, alinhamento e foco.
- A qualidade das relações humanas é tão importante quanto a qualidade do código.
- Diversidade cultural e funcional pode ser convertida em vantagem competitiva.
- A presença de lideranças facilitadoras é determinante para a coesão e o engajamento do time.

Em síntese, o estudo evidencia que o sucesso de um time distribuído não é acidental: ele é resultado de escolhas metodológicas, disciplina comportamental e um compromisso constante com a melhoria contínua.

5.2 Limitações do Estudo

Apesar de este trabalho oferecer uma análise aprofundada sobre os desafios e práticas de um time distribuído utilizando a metodologia *Scrum*, algumas limitações precisam ser explicitadas para contextualizar o alcance dos resultados e orientar interpretações mais precisas.

Primeiramente, o estudo concentra-se em um único caso, referente à empresa *Lean Software*, o que restringe a generalização dos achados. Embora estudos de caso permitam uma compreensão rica e detalhada de fenômenos organizacionais, sua aplicação tende a refletir características específicas do contexto investigado, como cultura organizacional, maturidade ágil e dinâmica interna dos times. Assim, os resultados aqui apresentados podem não se replicar em organizações com estruturas ou práticas substancialmente distintas.

Além disso, a análise está fundamentada predominantemente em um único método ágil o *Scrum*, adotado como referência para compreender a coordenação e a comunicação em times distribuídos. Embora o *Scrum* seja amplamente utilizado na indústria de software, outras abordagens ágeis ou híbridas, como Kanban, XP, DevOps

ou SAFe, não foram objeto de investigação neste trabalho. Dessa forma, não é possível afirmar se metodologias alternativas apresentariam resultados semelhantes no mesmo contexto.

Outro ponto a ser considerado refere-se ao contexto cultural relativamente homogêneo no qual o estudo foi conduzido. Embora os times estejam distribuídos entre Portugal, Brasil e Espanha, trata-se de países que compartilham elementos culturais relacionados ao modelo ocidental de organização do trabalho, comunicação corporativa e práticas de engenharia de software. Times distribuídos com participação de culturas asiáticas, africanas ou do leste europeu, por exemplo, poderiam apresentar dinâmicas significativamente distintas.

Por fim, as análises refletem um momento específico do projeto investigado, marcado por mudanças de escopo, reorganizações internas e diferentes níveis de maturidade técnica entre os membros. Esse cenário pode não reproduzir a realidade de projetos mais estáveis ou de organizações com longos históricos de trabalho distribuído. Assim, as conclusões devem ser interpretadas à luz dessas particularidades.

5.3 Sugestões para Estudos Futuros

A partir das limitações identificadas e das oportunidades reveladas durante a análise do caso, algumas direções de investigação podem contribuir para o aprofundamento do conhecimento sobre o desenvolvimento distribuído de software e a aplicação do *Scrum* nesse contexto.

Uma primeira possibilidade consiste na diversificação de estudos de caso, envolvendo organizações de distintos portes, setores e maturidades ágeis. A inclusão de empresas com times distribuídos em países culturalmente mais diversos especialmente em contextos multilíngues e multiculturais pode ampliar a compreensão sobre como fatores culturais influenciam práticas de comunicação, coordenação e autonomia em ambientes distribuídos. Estudos dessa natureza podem permitir comparações mais amplas e robustas entre diferentes modelos organizacionais.

Outra vertente promissora envolve a exploração de métodos ágeis alternativos ao *Scrum*. Embora o presente estudo tenha evidenciado como o *framework* contribuiu positivamente para a coordenação e para a clareza do trabalho, metodologias como *Kanban*, *Extreme Programming (XP)*, *DevOps* ou *frameworks* escalados como SAFe podem apresentar vantagens específicas quando aplicadas a times distribuídos. Investigações comparativas poderiam avaliar o impacto dessas abordagens em áreas como fluxo de trabalho, autonomia do time, produtividade e alinhamento entre equipes geograficamente dispersas.

Também se identifica um crescente interesse no papel da tecnologia no suporte ao trabalho distribuído, especialmente com o avanço de ferramentas digitais de colaboração, automação e inteligência artificial. Estudos futuros podem examinar como soluções emergentes como plataformas de comunicação com análise inteligente de interações, assistentes virtuais para coordenação de tarefas ou sistemas de documentação automatizada influenciam a eficiência, a qualidade das entregas e a experiência dos membros do time. Tais tecnologias podem redesenhar práticas fundamentais do *Scrum*, como reuniões diárias, revisões de *sprint* e refinamentos do *backlog*.

Além disso, seria relevante investigar aspectos humanos e comportamentais no trabalho distribuído sob a ótica de teorias contemporâneas de motivação, autonomia e desenvolvimento de equipes. Fatores como empatia, inteligência emocional e inclusão que emergiram como elementos centrais neste estudo podem ser explorados com maior profundidade em pesquisas qualitativas ou quantitativas, contribuindo para *frameworks* mais sensíveis às necessidades de times diversos.

Por fim, sugere-se o desenvolvimento de modelos híbridos de gestão ágil que integrem práticas de diferentes métodos para lidar melhor com a complexidade de projetos distribuídos. A construção e validação desses modelos, por meio de experimentações controladas ou estudos longitudinais, poderia trazer contribuições relevantes para organizações que buscam aumentar sua maturidade em desenvolvimento distribuído de software.

REFERÊNCIAS

- Boehm, B. & Li Guo Huang. (2003). Value-based software engineering: A case study. *Computer*, 36(3), 33–41. <https://doi.org/10.1109/MC.2003.1185215>
- Borges, C. C. (2012). *Processos para Desenvolvimento Distribuído de Software*.
- Camargo, K. G. A., & Galegale, N. V. (2024). Aumento da maturidade de equipes SCRUM de desenvolvimento de software. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 54, 22–38. <https://doi.org/10.17013/risti.54.22-38>
- Cibotto, R. A. G., Pagno, R. T., & Calvi, T. F. (2009). *Uma Análise da Dimensão Sócio-Cultural no Desenvolvimento Distribuído de Software*.
- Cohn, M. (2009). *Desenvolvimento de Software com Scrum Aplicando métodos ágeis com sucesso de Mike Cohn* (Bookman).
- Gren, L., Torkar, R., & Feldt, R. (2017). Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *Journal of Systems and Software*, 124, 104–119. <https://doi.org/10.1016/j.jss.2016.11.024>
- Júnior, A. N. B., Kamei, F. K., Alves, P. R. C., Azevedo, R. R., & Meira, S. (2011). *A Utilização de Práticas Scrum no Desenvolvimento de Software com Equipes Grandes e Distribuídas: Um Relato de Experiência*.
- Kalluri, R. (2024). Psychological Antecedents to Emergence of Team Autonomy in Agile Scrum Teams. *International Journal of Software Engineering & Applications*, 15(3), 13–29. <https://doi.org/10.5121/ijsea.2024.15302>
- Kniberg, H. (2008). *Scrum and XP from the trenches: How we do Scrum; [an agile war story]*. C4Media.
- Koscianski, A., & Soares, M. dos S. (2007). *Qualidade de Software - 2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. Novatec Editora.
- Kostin, D., & Strode, D. (2022). *Effective communication in globally distributed Scrum teams*.
- Noll, J., Razzak, M. A., & Beecham, S. (2017). Motivation and Autonomy in Global Software Development: An Empirical Study. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 394–399. <https://doi.org/10.1145/3084226.3084277>

- Pehls, G. T. da R. (2016). Desenvolvimento distribuído de software: Um estudo preliminar do estado da arte. *ResearchGate*. <https://www.researchgate.net/publication/309424554>
- Pereira, P., Torreão, P., & Marcal, A. S. (2007). *Entendendo Scrum para Gerenciar Projetos de Forma Ágil*. https://www.academia.edu/28954250/Entendendo_Scrum_para_Gerenciar_Projetos_de_Forma_%C3%81gil
- Prasetya, K. D., Suharjito, & Pratama, D. (2021). Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development. *Procedia Computer Science*, 179, 103–111. <https://doi.org/10.1016/j.procs.2020.12.014>
- Pressman, R. S. (2009). *Engenharia de Software de Pressman* (6.^a ed.). McGraw Hill. <https://www.wook.pt/livro/engenharia-de-software-6-edicao-roger-s-pressman/9623959>
- Prikladnicki, R., & Audy, J. L. N. (2002). *Um Modelo de Referência para Desenvolvimento Distribuído de Software*.
- Rato, F. de O. (2017). *DevOps em Sistemas de Informação*.
- Sabbagh, R. (2016). *Scrum Gestão ágil para projetos de sucesso—Casa do Código—Gestão ágil para projeto | Docsity*. Casa do Código. <https://www.docsity.com/pt/docs/scrum-gestao-agil-para-projetos-de-sucesso-casa-do-codigo/4879652/>
- Schwaber, K., & Beedle, M. (2002). *Agile Software Development With Scrum*. PEARSON EDUCATION (US). <https://www.wook.pt/livro/agile-software-development-with-scrum-mike-beedle/16250989>
- Shahzad, B., Awan, W., Fazal-e-Amin, Abro, A., Shoaib, M., & Alyahya, S. (2023). Framework for Effective Utilization of Distributed Scrum in Software Projects. *Computer Systems Science and Engineering*, 44(1), 407–422. <https://doi.org/10.32604/csse.2023.022601>
- Soares, F. S. F., Rodrigues, J. P., Neto, M. G., Bastos, P. R., Pereira, D. T. V., & Albuquerque, J. (2009). *Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software*.
- Soares, M. D. S. (2004). *Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software*.
- Sommerville, I. (2007). *Engenharia de software*. Pearson Prentice Hall.
- Spiegler, S. V., Heinecke, C., & Wagner, S. (2019). *Leadership Gap in Agile Teams: How Teams and Scrum Masters Mature* (Vol. 355, pp. 37–52). https://doi.org/10.1007/978-3-030-19034-7_3

- Spiegler, S. V., Heinecke, C., & Wagner, S. (2021). An empirical study on changing leadership in agile teams. *Empirical Software Engineering*, 26(3), 41. <https://doi.org/10.1007/s10664-021-09949-5>
- Stray, V., & Moe, N. B. (2020). Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *Journal of Systems and Software*, 170, 110717. <https://doi.org/10.1016/j.jss.2020.110717>
- Tomás, M. R. (2009). Métodos ágeis: Características, pontos fortes e fracos e possibilidades de aplicação [Agile methods: features, strong and weak aspects and application possibilities]. *IET Working Papers Series*, Artigo 09/2009. <https://ideas.repec.org//p/ieu/wpaper/18.html>
- Verwijns, C., & Russo, D. (2022). *A Theory of Scrum Team Effectiveness* (No. arXiv:2105.12439). arXiv. <https://doi.org/10.48550/arXiv.2105.12439>
- Wei, X., Prybutok, V., & Sauser, B. (2021). Review of supply chain management within project management. *Project Leadership and Society*, 2, 100013. <https://doi.org/10.1016/j.plas.2021.100013>
- Wonohardjo, E. P., Sunaryo, R. F., & Sudiyono, Y. (2019). A Systematic Review of SCRUM in Software Development. *JOIV : International Journal on Informatics Visualization*, 3(2), 108–112. <https://doi.org/10.30630/joiv.3.2.167>

ANEXO A – GUIÃO DA ENTREVISTA

1. Perfil do entrevistado

- Função no time
- Localização geográfica
- Tempo de experiência com Scrum
- Tempo de experiência em equipes distribuídas

2. Comunicação e colaboração

- Como são realizadas as interações diárias entre os membros do time?
- Quais ferramentas são utilizadas?
- Há desafios relacionados ao fuso horário?

3. Adaptações do Scrum ao contexto distribuído

- Como as cerimônias são conduzidas?
- Houve necessidade de modificar horários, formatos ou duração?

4. Cultura e integração

- Existem diferenças culturais perceptíveis entre os membros?
- Como essas diferenças são tratadas no trabalho cotidiano?

5. Percepção da eficácia do processo

- O Scrum contribui positivamente para o desempenho da equipe?
- Quais pontos podem ser melhorados?

ANEXO B – RESPOSTAS INTEGRAS DAS ENTREVISTAS (A–O)

ENTREVISTA A : SCRUM MASTER (Lisboa)

1. Perfil do entrevistado

Função no time: Scrum Master - coordenação, facilitação, remoção de impedimentos.

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 7 anos (CSM).

Tempo de experiência em equipes distribuídas: 5 anos (projetos com 2–4 locais).

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Interações diárias via Daily Meeting (Zoom/Teams), updates escritos no canal #daily do Slack para quem não pode participar, e uso do Jira para acompanhamento de tarefas.

Sou responsável por garantir que impedimentos sejam escalados e documentados.

Quais ferramentas são utilizadas?

Zoom/Teams (cerimônias síncronas), Slack (comunicação assíncrona), Jira (backlog e tickets), Confluence (documentação), Miro (workshops), Gitlab/Bitbucket (controle de versão).

Há desafios relacionados ao fuso horário?

Sim - sobreposição limitada com Campinas; resolvemos definindo um horário fixo de daily que maximize participação (horário intermediário) e aceitando updates escritos quando alguém não pode participar. No início, cada país tinha seu próprio ritmo e tom de voz; agora falamos a mesma língua ágil, priorizamos o cliente e a entrega, não a hierarquia

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Planning em 2 blocos (negócio + técnica), Daily de 15 minutos com webcam obrigatória, Reviews com demonstrações gravadas, Retros com Miro e votação (dot voting).

Houve necessidade de modificar horários, formatos ou duração?

Sim - shortenings em semanas intensas, dividir Plannings em sessões para fusos diferentes e gravações obrigatórias das sessões para consulta.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - portugueses tendem a ser mais diretos; brasileiros mais contextuais e conversacionais.

Como essas diferenças são tratadas no trabalho cotidiano?

Através de acordos explícitos (normas de comunicação), mediação do Scrum Master e rituais informais (cafés virtuais).

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - providencia ritmo, visibilidade e oportunidades regulares de alinhamento.

Quais pontos podem ser melhorados?

Automação de testes, melhor onboarding cultural para novos membros e revisões periódicas da cadência para evitar fadiga.

ENTREVISTA B — DESENVOLVEDOR BACKEND (Belo Horizonte)

1. Perfil do entrevistado

Função no time: Desenvolvedor Backend.

Localização geográfica: Belo Horizonte, Brasil.

Tempo de experiência com Scrum: 5 anos.

Tempo de experiência em equipes distribuídas: 4 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Daily por videoconferência; uso intenso de threads no Slack para dúvidas técnicas e PRs no Git para revisão.

Quais ferramentas são utilizadas?

Slack, Zoom, Jira, Gitlab.

Há desafios relacionados ao fuso horário?

Sim - iniciar o dia já com mensagens acumuladas do turno europeu; solução: checagem matinal e priorização via backlog.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Planning com pré-leitura das user stories, estimativas via Planning Poker online; Retros com Miro.

Houve necessidade de modificar horários, formatos ou duração?

Sim - Planning costuma ser dividido para reduzir tempo em videoconferência contínua; Dailys estritos a 15 minutos.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - estilos de comunicação e expectativas sobre respostas imediatas.

Como essas diferenças são tratadas no trabalho cotidiano?

Acordos na equipe e uso de mensagens escritas quando a sincronia falha.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - entrega incremental facilita teste e release.

Quais pontos podem ser melhorados?

Melhor integração entre sistemas legados e maior automação para reduzir bloqueios de integração.

ENTREVISTA C : PRODUCT OWNER (Campinas)

1. Perfil do entrevistado

Função no time: Product Owner.

Localização geográfica: Campinas, Brasil.

Tempo de experiência com Scrum: 6 anos.

Tempo de experiência em equipes distribuídas: 5 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Participo do Daily algumas vezes por semana; comunico prioridades via backlog no Jira; sessões de alinhamento com stakeholders via Zoom.

Quais ferramentas são utilizadas?

Jira, Confluence, Slack, Zoom.

Há desafios relacionados ao fuso horário?

Sim - reorganizo sessões de stakeholders para sobreposição mínima com Portugal.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Documentação técnica frequente para garantir Definition of Ready; Planning com objetivos claros.

Houve necessidade de modificar horários, formatos ou duração?

Sim - Documentação técnica assíncrono (comentários nas histórias) e plannings com agendas rígidas.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - diferenças no nível de formalidade e detalhamento das comunicações.

Como essas diferenças são tratadas no trabalho cotidiano?

Documentação reforçada e sessões de contextualização antes de cada entrega crítica.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - reduz ambiguidade e melhora priorização.

Quais pontos podem ser melhorados?

Maior disponibilidade do PO para clarificar requisitos em horários sensíveis; mentoring para outros POs juniores.

ENTREVISTA D : DESENVOLVEDOR FRONTEND (Lisboa)

1. Perfil do entrevistado

Função no time: Desenvolvedor Frontend.

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 4 anos.

Tempo de experiência em equipes distribuídas: 3 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Dailys, revisão de PRs, pareamento remoto ocasional (Live Share).

Quais ferramentas são utilizadas?

Jira, Confluence, Slack, VS Code Live Share.

Há desafios relacionados ao fuso horário?

Alguns - sincronização para pair programming costuma ser agendada nos overlap days.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Dailys objetivos, Reviews com demo; Retros com follow-up imediato.

Houve necessidade de modificar horários, formatos ou duração?

Sim - Requests para que demos sejam gravadas e documentação técnica complementada.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - prefiro comunicação direta; ajustei meu tom ao colaborar com brasileiros.

Como essas diferenças são tratadas no trabalho cotidiano?

Discussões explícitas em retrospectivas sobre tom e feedback. Sabemos que podemos confiar nas entregas porque todas estão visíveis e auditáveis no Jira.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - mantém prioridades visíveis e reduz rework.

Quais pontos podem ser melhorados?

Mais tempo para pair programming e suíte de testes frontend mais robusta.

ENTREVISTA E : QA ENGINEER (Campinas)

1. Perfil do entrevistado

Função no time: QA Engineer.

Localização geográfica: Campinas, Brasil.

Tempo de experiência com Scrum: 6 anos.

Tempo de experiência em equipes distribuídas: 4 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Dailys, integração com pipelines, reportes de bloqueios via Slack, testes coordenados em horários de overlap.

Quais ferramentas são utilizadas?

TestRail, Jenkins/Gitlab CI, Slack, Jira.

Há desafios relacionados ao fuso horário?

Sim - agendamento de testes de integração exige coordenação para não impactar os times em outro horário.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Retros focadas em qualidade, definição clara de critérios de aceitação e testes automatizados como deliverable.

Houve necessidade de modificar horários, formatos ou duração?

Sim - sessões de teste conjunto são programadas em overlap days; reporting automatizado para quem não participa.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - nuances de comunicação sobre severidade e urgência.

Como essas diferenças são tratadas no trabalho cotidiano?

Uso de SLAs internos e checklists padronizados para reduzir subjetividade.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - facilita a implementação de qualidade por *sprint*.

Quais pontos podem ser melhorados?

Maior investimento em automação e cobertura de testes; treinamento cross-site em ferramentas de teste.

ENTREVISTA F : TECH LEAD (Lisboa)

1. Perfil do entrevistado

Função no time: Tech Lead / Arquiteto técnico.

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 9 anos.

Tempo de experiência em equipes distribuídas: 6 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Revisões técnicas semanais, design sessions via Miro, PR reviews e dailys para bloqueios.

Quais ferramentas são utilizadas?

Gitlab, Jira, Confluence, Miro, Slack.

Há desafios relacionados ao fuso horário?

Sim - coordenar deploys entre fusos sem causar downtime exige janelas programadas.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Planning com split técnico/negócio; design reviews assíncronos quando necessário.

Houve necessidade de modificar horários, formatos ou duração?

Sim - sessões de design são gravadas e seguidas de documentos resumidos.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - estilos de documentação e expectativas de resposta.

Como essas diferenças são tratadas no trabalho cotidiano?

Padronização de templates e coaching técnico para alinhamento.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - permite visibilidade e decisões rápidas.

Quais pontos podem ser melhorados?

Melhor governança de dependências e integração entre *squads*.

ENTREVISTA G : PRODUCT OWNER (Campinas)

1. Perfil do entrevistado

Função no time: Product Owner.

Localização geográfica: Campinas, Brasil.

Tempo de experiência com Scrum: 8 anos.

Tempo de experiência em equipes distribuídas: 6 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Daily esporádicos comigo; alinhamentos com stakeholders; uso contínuo do backlog para priorização.

Quais ferramentas são utilizadas?

Jira, Miro, Slack, Zoom.

Há desafios relacionados ao fuso horário?

Sim - disponibilidade para demos e alinhamentos com Portugal exige ajuste de agenda.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Planning com foco em value delivery; Reviews com gravação; Retros com ações como tickets.

Houve necessidade de modificar horários, formatos ou duração?

Sim - reuniões com stakeholders em horários que atendam ambas as regiões.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - percepção de urgência e modo de interação.

Como essas diferenças são tratadas no trabalho cotidiano?

Esforço de contextualização e explicitação de critérios de aceitação.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - mantém o time focado em entregas com valor.

Quais pontos podem ser melhorados?

Maior presença do PO em todas as cerimônias e mais alinhamento pré-planning.

ENTREVISTA H : QA ENGINEER (Madrid, Espanha)

1. Perfil do entrevistado

Função no time: QA Engineer (Sênior).

Localização geográfica: Madrid, Espanha.

Tempo de experiência com Scrum: 5 anos.

Tempo de experiência em equipes distribuídas: 5 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Dailys, Slack técnico, e runbooks para incidentes; relatórios automáticos para logs de teste.

Quais ferramentas são utilizadas?

Jira, Slack, Jenkins, TestRail.

Há desafios relacionados ao fuso horário?

Sim - o horário de testes automatizados precisa ser programado para não conflitar com horários de suporte em outro país.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Cerimônias síncronas com sumários escritos; oportunidades de feedback registradas.

Houve necessidade de modificar horários, formatos ou duração?

Sim - Dailys às vezes acontecem em horários diferentes quando há entrega crítica.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - abordagem ao feedback e tom de comunicação.

Como essas diferenças são tratadas no trabalho cotidiano?

Explicitação de normas de comunicação e sessões de team building virtuais.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - aumenta previsibilidade e permite testes contínuos.

Quais pontos podem ser melhorados?

Automação mais ampla e documentação de playbooks de incidentes.

ENTREVISTA I : DEVOPS ENGINEER (Belo Horizonte)

1. Perfil do entrevistado

Função no time: DevOps Engineer - CI/CD, infraestrutura como código.

Localização geográfica: Belo Horizonte, Brasil.

Tempo de experiência com Scrum: 6 anos.

Tempo de experiência em equipes distribuídas: 5 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Monitoramento em dashboards, alertas em Slack, reuniões de deploy e coordenação de janelas.

Quais ferramentas são utilizadas?

Jenkins/Gitlab CI, Prometheus, Grafana, Slack, Jira.

Há desafios relacionados ao fuso horário?

Sim - coordenação de deploys entre fusos; resolvido por janelas programadas e runbooks.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Planning inclui checkpoints de release; Dailys para incidentes críticos; Retros focadas em operações.

Houve necessidade de modificar horários, formatos ou duração?

Sim - reuniões de deploy são agendadas para janelas de overlap e possuem checklists.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Leves diferenças no senso de urgência; mais pragmatismo entre europeus.

Como essas diferenças são tratadas no trabalho cotidiano?

Acordos operacionais e runbooks claros.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - suporte para integração contínua e entrega frequente.

Quais pontos podem ser melhorados?

Automação de segurança e testes pré-deploy mais robustos.

ENTREVISTA J : ANALISTA DE NEGÓCIOS (Lisboa)

1. Perfil do entrevistado

Função no time: Analista de Negócios / Liaison com stakeholders.

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 7 anos.

Tempo de experiência em equipes distribuídas: 6 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Workshops de discovery, revisão de requisitos e alinhamentos assíncronos documentados.

Quais ferramentas são utilizadas?

Confluence, Jira, Slack, Miro.

Há desafios relacionados ao fuso horário?

Sim - necessidade de confirmar decisões com stakeholders em horários distintos; mitigado por documentação.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Documentação técnica assíncronas com comentários no Jira e sessões sintéticas para

alinhamento final.

Houve necessidade de modificar horários, formatos ou duração?

Sim - menor duração para sessões presenciais remotas e mais preparação prévia.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - estilos de negociação e formalidade.

Como essas diferenças são tratadas no trabalho cotidiano?

Templates e glossários para termos de negócio; clarificação de expectativas.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - traz foco em valor e prioridades.

Quais pontos podem ser melhorados?

Ritualizar a captura de decisões de negócio em formato padronizado.

ENTREVISTA K : DESENVOLVEDORA FULL STACK (Campinas)

1. Perfil do entrevistado

Função no time: Desenvolvedora Full Stack.

Localização geográfica: Campinas, Brasil.

Tempo de experiência com Scrum: 5 anos.

Tempo de experiência em equipes distribuídas: 4 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Pareamento remoto, dailys e PR reviews.

Quais ferramentas são utilizadas?

VS Code Live Share, Gitlab, Slack, Jira.

Há desafios relacionados ao fuso horário?

Sim - pareamento é feito nos horários de overlap; quando não é possível, we do asynchronous pair via recorded sessions.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Dailys curtos; pair programming formalizado em calendário; reviews com demonstrações.

Houve necessidade de modificar horários, formatos ou duração?

Sim - ajuste nos horários de pair e uso de gravações.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - diferenças de ritmo e nível de detalhamento.

Como essas diferenças são tratadas no trabalho cotidiano?

Combinações de mensagens e protocolos de revisão para minimizar ambiguidade.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - facilita colaboração e entrega contínua.

Quais pontos podem ser melhorados?

Mais tempo dedicado a *pair* e *code review*; integração entre devs e design mais frequente.

ENTREVISTA L : SCRUM MASTER (Lisboa)

1. Perfil do entrevistado

Função no time: Scrum *Master* (*senior*).

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 8 anos (PSM I).

Tempo de experiência em equipes distribuídas: 7 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Dailys, *checklists* de impedimentos, e 1:1 regulares para clima e bem-estar.

Quais ferramentas são utilizadas?

Zoom, Slack, Jira, Confluence.

Há desafios relacionados ao fuso horário?

Sim - cuidar de energia e fadiga dos membros por conta de reuniões fora do horário local.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Retros com foco em melhorias de processo; *Planning* com pré-read e *timeboxes* rígidos.

Houve necessidade de modificar horários, formatos ou duração?

Sim - sessões de cultura e *onboarding* foram incluídas no cronograma.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - diferenças de estilo e expectativa sobre escalonamento.

Como essas diferenças são tratadas no trabalho cotidiano?

Mediação ativa e criação de regras de convivência digital.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - facilita foco e adaptação.

Quais pontos podem ser melhorados?

Políticas para limitar reuniões e proteger blocos de desenvolvimento profundo.

ENTREVISTA M : UX DESIGNER (Campinas)

1. Perfil do entrevistado

Função no time: UX *Designer*.

Localização geográfica: Campinas, Brasil.

Tempo de experiência com Scrum: 6 anos.

Tempo de experiência em equipes distribuídas: 4 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Sessões de sketching remoto, *feedbacks* assíncronos via *Figma*, e *workshops* de alinhamento.

Quais ferramentas são utilizadas?

Figma, Miro, Slack, Zoom.

Há desafios relacionados ao fuso horário?

Alguns - testes de usabilidade remotos são programados para ajustar aos horários de usuários e equipes.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Incorporação de sessões de design no *backlog*; *reviews* com *stakeholders* remotos.

Houve necessidade de modificar horários, formatos ou duração?

Sim - sessões de co-criação fracionadas para acomodar fusos.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - percepções de usabilidade podem variar por região.

Como essas diferenças são tratadas no trabalho cotidiano?

Pesquisa de usuário multi-local e inclusão de personas diversas.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - possibilita entregas iterativas e validação rápida com usuários.

Quais pontos podem ser melhorados?

Maior sincronização entre design e desenvolvimento durante a *sprint*.

ENTREVISTA N : QA SÊNIOR (Madrid, Espanha)

1. Perfil do entrevistado

Função no time: QA Sênior.

Localização geográfica: Madrid, Espanha.

Tempo de experiência com Scrum: 9 anos.

Tempo de experiência em equipes distribuídas: 7 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Dailys, testes compartilhados em pipelines, e checklists de release.

Quais ferramentas são utilizadas?

Jenkins, TestRail, Jira, Slack.

Há desafios relacionados ao fuso horário?

Sim - coordenação de janelas de testes e deploy é necessária.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Cerimônias com sumários escritos e pontos de ação claros; sessões de co-teste programadas.

Houve necessidade de modificar horários, formatos ou duração?

Sim - testes de integração ocorreram em horários noturnos planejados por todos.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - leve diferença de estilo; espanhóis mais diretos.

Como essas diferenças são tratadas no trabalho cotidiano?

Explicitação de prioridades e definição clara de critérios de aceitação.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - organização e previsibilidade para testes.

Quais pontos podem ser melhorados?

Mais automação e documentação de casos de regressão.

ENTREVISTA O : GESTOR DE PROJETO / PM (Lisboa)

1. Perfil do entrevistado

Função no time: Gestor de Projeto / Program Manager.

Localização geográfica: Lisboa, Portugal.

Tempo de experiência com Scrum: 10 anos.

Tempo de experiência em equipes distribuídas: 8 anos.

2. Comunicação e colaboração

Como são realizadas as interações diárias entre os membros do time?

Coordenação entre *squads*, dailys de sincronização e reports semanais de progresso.

Quais ferramentas são utilizadas?

Jira, Confluence, Slack, Teams.

Há desafios relacionados ao fuso horário?

Sim - principalmente em alocações de recursos e reuniões estratégicas; mitigamos com agendas fixas.

3. Adaptações do Scrum ao contexto distribuído

Como as cerimônias são conduzidas?

Reviews consolidados inter-*squad*, plannings coordenados e checkpoints de integração.

Houve necessidade de modificar horários, formatos ou duração?

Sim - plannings envolvendo múltiplas *squads* são fracionados e com entregas intermediárias.

4. Cultura e integração

Existem diferenças culturais perceptíveis entre os membros?

Sim - diferente abordagem para resolução de conflitos e tomada de decisão.

Como essas diferenças são tratadas no trabalho cotidiano?

Políticas de governança e templates de decisão para uniformizar práticas.

5. Percepção da eficácia do processo

O Scrum contribui positivamente para o desempenho da equipe?

Sim - favorece alinhamento e adaptações rápidas.

Quais pontos podem ser melhorados?

Maior integração entre áreas não-técnicas (comercial, operações) e *squads*.