

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO  
ENGENHARIA DE PRODUÇÃO CIVIL

João Vitor Maurício de Abreu

**Fatores críticos para aplicação do *Scrum* em projetos de desenvolvimento de *software***

Florianópolis

2022

João Vitor Maurício de Abreu

**Fatores críticos para aplicação do *Scrum* em projetos de desenvolvimento de *software***

Trabalho Conclusão do Curso de Graduação em Engenharia de Produção Civil do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro Civil, habilitado em Engenharia de Produção.

Orientador: Prof. Lynceo Falavigna Braghirolli,

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Abreu, João Vitor Maurício de  
Fatores críticos para aplicação do Scrum em projetos de desenvolvimento de software / João Vitor Maurício de Abreu ; orientador, Lynceo Falavigna Braghirolli, 2022.  
74 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, , Graduação em Engenharia de Produção Civil, Florianópolis, 2022.

Inclui referências.

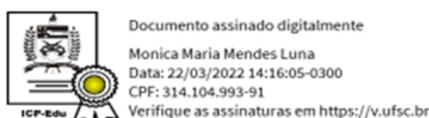
1. Engenharia de Produção Civil. 2. Gestão de Projetos. 3. Metodologias Ágeis. 4. Scrum. 5. Scrumbut. I. Braghirolli, Lynceo Falavigna . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Produção Civil. III. Título.

João Vitor Maurício de Abreu

**Título:** Fatores críticos para aplicação do *Scrum* em projetos de desenvolvimento de *software*

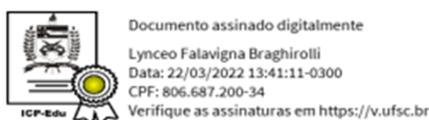
Este Trabalho Conclusão de Curso foi julgado adequado e aprovado em sua forma final pelo curso de graduação em Engenharia de Produção Civil da Universidade Federal de Santa Catarina.

Local, 22 de março de 2022.

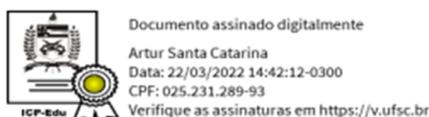


Prof.(a) Mônica Maria Mendes Luna, Dra.  
Coordenadora do Curso

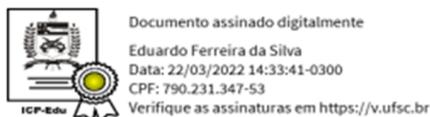
**Banca Examinadora:**



Prof. Lynceo Falavigna Braghirolli, Dr.  
Orientador  
Instituição UFSC



Prof. Artur Santa Catarina, Dr.  
Avaliador  
Instituição UFSC



Prof. Eduardo Ferreira da Silva, Dr.  
Avaliador  
Instituição UFSC

## RESUMO

Diante do cenário de crescimento exponencial do setor de tecnologia nas últimas décadas e a complexidade inerente à criação de produtos de *software*, as metodologias ágeis de gerenciamento de projetos vêm ganhando cada vez mais popularidade, e entre estas, o método *Scrum* se destaca indisputavelmente como o *framework* ágil mais utilizado no mundo. Entretanto, apesar de sua popularidade e seus benefícios comprovados em eficiência e qualidade, as práticas e processos pregados na teoria e em especial pelo Guia *Scrum*, são raramente seguidos conforme as definições claras dos autores do método, os quais não caracterizam aplicações incompletas do *framework* como *Scrum*. Desta forma, estes desvios comuns da prática ideal do método ocorrem à medida que as empresas apresentam dificuldades em adequar o *Scrum* aos seus contextos e optam assim por adaptações nos processos relacionados ao *framework*. Neste contexto, este trabalho busca demonstrar os principais pontos críticos e impeditivos relacionados à aplicação do *Scrum* em empresas de desenvolvimento de *software*, descrevendo as principais causas de sucesso e fracasso do método, e investigando a aplicação em projetos reais, levantando a percepção dos gestores e times *Scrum* acerca do método e analisando cronologicamente sua implementação em uma empresa desenvolvedora de *software*. O levantamento das percepções de colaboradores do setor foi realizado através de entrevistas, e a análise dos times *Scrum* foi efetuada através de uma pesquisa-ação, onde o autor participou ativamente nas escolhas realizadas pela empresa em estudo. Por fim, foi possível elencar diversos anti-padrões do *Scrum* praticados nos projetos abordados pelo trabalho e identificar suas causas, garantindo uma melhor compreensão das escolhas que fomentam a tendência de desvirtuamento do *Scrum*.

**Palavras-chave:** Gestão de Projetos. Metodologias Ágeis. *Scrum*. *Scrumbut*. Engenharia de *Software*. Desenvolvimento de *Software*. Desenvolvimento de *software* ágil. *Team Effectiveness*.

## ABSTRACT

In face of the exponential technology industry growth in the last decades and the complexity inherent in the development of software products, agile methods of project management have been gaining even more popularity, and among those, the *Scrum* method stands out undisputedly as the most used agile framework worldwide. However, in spite of its popularity and proved benefits in efficiency and quality, the practices and processes preached in theory and in special the *Scrum* Guide, are rarely followed according to the clear definitions made by the method's authors, who do not consider incomplete applications of the framework as *Scrum*. In this way, these common deviations of the ideal way of practice of the method occur as companies show difficulties in adapting *Scrum* to their contexts and thus opt for adaptations in the framework related processes. With this in mind, this paper seeks to demonstrate critical issues and blocks related to *Scrum* execution in software development companies, describing the main causes of success and failure of the method, and investigating the implementation in real projects, identifying the perception of managers and *Scrum* teams on the method and chronologically analyzing its adoption in a software development organization. The gathering of the industry's workers perceptions was made through interviews, and the *Scrum* teams analysis executed as an action research, where the author actively took part in the decisions made by the company under study. Lastly, it was possible to list several *Scrum* anti-patterns practiced in the projects covered in this paper and also identify their causes, assuring a better comprehension on the choices that instigate the tendency to distort *Scrum*.

**Keywords:** *Project Management. Agile Methods. Scrum. Scrumbut. Software Engineering. Software Development. Agile Software Development. Team Effectiveness.*

## LISTA DE FIGURAS

Figura 1: Comparação entre modelo Cascata e modelo Ágil de projeto. ....	17
Figura 2: Comparação de linha do tempo entre projeto ágil e tradicional .....	18
Figura 3: Representação da Sprint e seus eventos. ....	23
Figura 4: Principais razões para empresas adotarem metodologias ágeis. ....	28
Figura 5: Principais áreas da empresa beneficiadas pela implementação do Agile. ....	29
Figura 6: Principais barreiras para adotar e escalar as práticas Agile nas empresas. ....	30
Figura 7: Relação entre anti-padrões e conceitos Scrum. ....	31
Figura 8: Etapas da pesquisa-ação. ....	45
Figura 9: Funções padrão para cada função nas <i>Sprints</i> . ....	57
Figura 10: Fluxo padrão de requisitos em uma Sprint. ....	58
Figura 11: Esquema de relacionamento de causas, anti-padrões e conceitos Scrum. ....	67

## LISTA DE QUADROS

Quadro 1: Resumo de anti-padrões, motivações e estratégias do capítulo 2.....	40
Quadro 2: Roteiro da Entrevista .....	43
Quadro 3: Nível de Domínio Scrum - Classificação .....	43
Quadro 4: Informações dos Entrevistados.....	47
Quadro 5: Nível de Domínio - Entrevistados .....	48
Quadro 6: Projetos da Empresa A .....	53
Quadro 7: Lista de causas diretas ao anti-padrões e desvios do Scrum. ....	64
Quadro 8: Anti-padrões e desvios do Scrum e suas causas diretas. ....	65

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>9</b>
1.1	PROBLEMA DE PESQUISA .....	10
1.2	OBJETIVOS .....	10
<b>1.2.1</b>	<b>Objetivo Geral.....</b>	<b>10</b>
<b>1.2.2</b>	<b>Objetivos Específicos .....</b>	<b>11</b>
1.3	JUSTIFICATIVA .....	11
1.4	DELIMITAÇÃO DO ESTUDO .....	12
1.5	ESTRUTURA DO TRABALHO .....	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1	METODOLOGIAS ÁGEIS.....	15
2.2	MÉTODO <i>SCRUM</i> .....	19
<b>2.2.1</b>	<b>Filosofia <i>Scrum</i>.....</b>	<b>20</b>
<b>2.2.2</b>	<b><i>Scrum Team</i> .....</b>	<b>21</b>
<b>2.2.3</b>	<b>Eventos <i>Scrum</i>.....</b>	<b>23</b>
2.2.3.1	<i>Sprint</i> .....	23
2.2.3.2	<i>Sprint Planning</i> .....	24
2.2.3.3	<i>Daily Scrum</i> .....	25
2.2.3.4	<i>Sprint Review</i> .....	25
2.2.3.5	<i>Sprint Retrospective</i> .....	26
2.3	DADOS SOBRE A APLICAÇÃO DE AGILE E <i>SCRUM</i> .....	26
2.4	<i>SCRUMBUT</i> .....	30
2.5	DESDOBRAMENTOS DO REFERENCIAL TEÓRICO .....	40
<b>3</b>	<b>METODOLOGIA DE PESQUISA .....</b>	<b>41</b>
3.1	Etapas da pesquisa .....	41
<b>3.1.1</b>	<b>Contextualização das Empresas .....</b>	<b>41</b>

3.1.2	<b>Entrevistas</b> .....	42
3.1.3	<b>Pesquisa-ação</b> .....	44
4	<b>RESULTADOS</b> .....	46
4.1	Entrevistas .....	46
4.1.1	<b>Bloco 1 - Conhecimento do respondente sobre o <i>Scrum</i></b> .....	47
4.1.2	<b>Bloco 2 - Características positivas do <i>Scrum</i></b> .....	48
4.1.3	<b>Bloco 3 - Viabilidade do método 100% conforme o Guia <i>Scrum</i></b> .....	49
4.1.4	<b>Bloco 4 – Opiniões gerais do <i>Scrum</i></b> .....	50
4.1.5	<b>Análise geral e discussão sobre as entrevistas</b> .....	51
4.2	<b>RESULTADOS DA PESQUISA-AÇÃO</b> .....	52
4.2.1	<b>PROJETOS E EQUIPES</b> .....	53
4.2.2	<b>MÉTODO DE GESTÃO ADOTADO PELA EMPRESA A</b> .....	56
4.2.3	<b>PRINCIPAIS AÇÕES DE SUPORTE AO MÉTODO</b> .....	59
4.2.4	<b>ANÁLISE E DISCUSSÃO</b> .....	60
4.3	<b>CAUSAS DIRETAS DE ANTI-PADRÕES E DESVIOS NO <i>Scrum</i></b> .....	64
5	<b>CONSIDERAÇÕES FINAIS</b> .....	68
5.1	<b>TRABALHOS FUTUROS</b> .....	69
	<b>REFERÊNCIAS</b> .....	70
	<b>APÊNDICE A – Cronologia da evolução do <i>Scrum</i> na Empresa A</b> .....	72

## 1 INTRODUÇÃO

Desde sua criação em meados da década de 90, as metodologias ágeis para gerenciamento de projetos têm se tornado cada vez mais populares, à medida que a demanda global por produtos de alta complexidade e entregues o mais rápido continua a crescer. Ainda que a popularização de metodologias ágeis tenha permitido uma adesão gradual em diversas áreas, atualmente, os principais e mais populares métodos continuam sendo aplicados em desenvolvimento de *software*, pois foram criados especificamente para buscar alternativas mais eficientes neste setor.

Sob a ótica atual de uma variada gama de novas técnicas, ferramentas de gestão e *frameworks* de desenvolvimento de software, o método *Scrum* provou-se ao longo dos anos, majoritariamente, como o método ágil mais utilizado pelas empresas. Impulsionado ainda mais pela crise durante a pandemia do COVID-19, em 2021, o *Scrum* e suas variações são utilizados em 81% das organizações que utilizam métodos ágeis, de acordo com o *15th State of Agile Report* (2021).

Também conhecido simplesmente por *agile*, o método ágil tem como premissa permitir que desenvolvedores trabalhem com requisitos de produtos muito bem definidos, produzam em curtos intervalos de tempo e incorporem novos requisitos do cliente/usuário rapidamente (MORANDINI et al., 2021). Aliados aos dogmas do *Scrum* cada vez mais apoiados pela comunidade de equipes de desenvolvimento, segundo Hron e Obwegeser (2018), os praticantes do método continuam aumentando suas expectativas sobre o que as abordagens de desenvolvimento ágil conseguem entregar. Desta forma, os resultados prometidos pelo método são os grandes motivadores para sua implementação.

Os grandes benefícios que o *Scrum* assegura nos projetos de desenvolvimento são referentes à ampliação da colaboração, transparência, melhoria contínua e adaptação às necessidades dos clientes, através da consideração da imprevisibilidade do processo de desenvolvimento de *software*, adotando uma abordagem incremental, minimizando o planejamento, estimativas e outras tarefas operacionais, e ainda estabelecendo comunicação contínua entre o time e com o cliente. Suas abordagens para alcançar esses objetivos são consideradas mais flexíveis que os métodos tradicionais de gestão de projetos, e, portanto, tendem a ser mais eficientes ao se adaptar a mudanças repentinas (HASSANI-ALAOUI; CAMERON; GIANNELIA, 2020).

Uma das principais características das equipes *Scrum* é o autogerenciamento do time, o qual tem a autoridade e responsabilidade de promover, seguindo as regras do *Scrum*, as escolhas mais efetivas para o ritmo e qualidade das entregas. Entretanto, segundo Eloranta et. al (2016), na prática, o *Scrum* tende a sofrer alterações nos seu *framework*, devido às características e circunstâncias únicas de cada empresa e seus ambientes de trabalho.

Desta forma, as equipes naturalmente criam adaptações ao método e reduzem o potencial de suas entregas ao se desvirtuarem das regras definidas pelo *Scrum*, ainda que o Guia *Scrum* (2020) alerte que todos os seus procedimentos são imutáveis e é necessária a realização do conjunto completo destas práticas, ou seja, a sua implementação parcial não é caracterizada como *Scrum*. (HASSANI-ALAOUI; CAMERON; GIANNELIA, 2020).

## 1.1 PROBLEMA DE PESQUISA

No cenário atual, segundo Eloranta et. al (2016), as práticas oficiais do *Scrum* não são seguidas de maneira consistente na indústria, e sabe-se que as organizações querem respeitar os procedimentos estabelecidos pelo *Scrum*, mas por diversos motivos não se sentem capazes. Desta forma, é necessário entender quais são os principais desvios do método e tendências não saudáveis do *Scrum* nas empresas e então definir os pontos críticos que as levam a este cenário, da maioria das empresas de TI não aplicarem o *Scrum* em sua plenitude e, por consequência, não alcançarem os resultados prometidos pelo modelo de gestão de projetos de desenvolvimento mais adotado no mundo.

## 1.2 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste trabalho de conclusão de curso.

### 1.2.1 Objetivo Geral

Demonstrar os principais fatores críticos e impeditivos à aplicação do método ágil *Scrum* de gerenciamento de projetos de desenvolvimento de software.

## 1.2.2 Objetivos Específicos

Com o intuito de demonstrar as dificuldades encontradas por empresas de desenvolvimento de *software* na implementação íntegra do *Scrum*, este trabalho tem como objetivos específicos:

- Descrever as principais causas de sucesso ou fracasso na implementação do método ágil *Scrum* presentes na literatura.
- Levantar a percepção dos gestores e equipes de projetos sobre o uso de métodos ágeis em uma empresa de TI.
- Analisar a implementação do método ágil *Scrum* em uma empresa de TI.

## 1.3 JUSTIFICATIVA

Desde o início da década de 90, segundo Lei et al. (2017), com o crescimento exponencial das indústrias de engenharia de *hardware* e *software*, metodologias ágeis de gerenciamento de projetos tornaram-se igualmente populares, provando-se muito eficazes em ajudar organizações a alcançarem ótimos resultados em seus produtos. Aliado a esta ideia, Kussunga e Ribeiro (2019), afirmam que metodologias ágeis já provaram entregar produto de alta qualidade em menor tempo e impulsionar maior satisfação dos clientes, e que nos últimos anos conseguiram, com sucesso, otimizar em geral os processos de projetos de *software*.

Estes métodos surgiram, pois, também conforme Kussunga e Ribeiro (2019), os métodos tradicionais de desenvolvimento de *software* costumam ter processos onerosos que prestam mais atenção à indústria e padronização, e não são assim mais adequados, desta forma geralmente resultam em ciclos de desenvolvimento muito longos e mais caros. Assim, deste cenário que clamava por práticas mais eficientes de gestão de projetos de *software*, surgiu o *Scrum*.

O *Scrum* promete ajudar as empresas em um mercado cada vez mais competitivo, com projetos cada vez mais complexos devido aos requisitos de clientes sempre diversos e em constante mudança (KUSSUNGA; RIBEIRO, 2019). Segundo o *15th State of Agile Report* (2021), a taxa de crescimento da adoção de métodos ágeis de 37% em 2020, que já era alta, cresceu para 86% em 2021 devido à pandemia da *COVID-19*, considerando que 81% destas empresas utilizam o *Scrum* ou derivações do *Scrum* tradicional.

Neste cenário de pandemia e obrigatoriedade do trabalho remoto, também segundo o *15th State of Agile Report* (2021), apenas 3% das empresas de TI pretendem retornar com o

trabalho presencial de forma integral, desta forma, os desafios para manter uma equipe engajada e que alcança resultados concisos em um projeto de desenvolvimento de *software* tornou-se muito maior. Por outro lado, as razões para que empresas adotassem o *Agile* continuam as mesmas por muitos anos, sendo que 64% afirmaram querer aumentar a capacidade de gerenciar mudanças de prioridades e acelerar a entrega de *software* e 47% pretendem aumentar a produtividade do time, melhorar o alinhamento entre o negócio e o setor de TI. Desta forma, estas empresas tem a escolha do *Scrum* justificada pelo cenário no setor e apresentam grandes desafios sem precedentes.

Apesar de toda a popularidade do *Scrum* no ramo de TI, 46% das empresas pesquisadas no *15th State of Agile Report (2021)* afirmam apresentar inconsistências em seus processos e práticas ágeis, 43% possuem divergências relacionadas aos valores ágeis e 46% resistência geral e organizacional quanto a mudanças. Além destes, motivos como falta de experiência e treinamento, falta de participação da liderança e suporte insuficiente dos gestores, se destacam entre as principais barreiras para a adoção e crescimento do *Scrum* nas organizações pesquisadas. E ainda, infere-se que, todos esses obstáculos resultam em grandes perdas para essas organizações, pois culminam diretamente em prazos maiores de entrega de produtos e insatisfação dos clientes.

Outro viés com o qual justifica-se este estudo, é representado no meio acadêmico, pois de acordo com Lei et Al. (2017), apesar do *Scrum* ter se tornado uma metodologia comum até então, existem estudos que apontam que apenas 3% da evidência científica do desenvolvimento ágil de *software* é focado no *Scrum*”.

Por fim, este estudo trará benefícios diretos à empresa participante da pesquisa, pois a partir da análise de seus processos e seus obstáculos de produtividade quanto aos seus processos *Agile*, esta poderá implementar mudanças efetivas em seus projetos, beneficiando por consequência da melhor qualidade de seus serviços, seus clientes.

#### 1.4 DELIMITAÇÃO DO ESTUDO

Com o intuito de definir a amplitude da abordagem deste trabalho, é preciso delimitar os objetos de estudo da pesquisa. Este trabalho aborda literatura e casos referentes especificamente a empresas de desenvolvimento de *software*, pois estes compõem a imensa maioria das aplicações do *Scrum*.

Este trabalho utilizou apenas uma empresa e seus projetos para identificar causas de desvios do *Scrum* e, portanto, os resultados encontrados estão fortemente vinculados às características e contextos desta empresa e suas equipes.

Quanto à empresa entrevistada e abordada em maiores detalhes, é necessário estabelecer que os dados referentes às informações de clientes, contas, pessoas e outros dados privados foram alterados ou ocultados, a pedido da empresa, para fins de segurança e privacidade. Além disso, é válido ressaltar que nem todas as sugestões de mudanças levantadas neste trabalho poderão ser acompanhadas, pois sua execução poderá ocorrer após a janela de tempo do estudo, desta forma, o autor e pesquisador não se compromete a acompanhar quaisquer ações tomadas pela empresa após o término do estudo.

Alinhado a esta limitação do período de estudo, torna-se necessário também destacar que ainda que o objetivo da empresa ao participar deste estudo seja direcionar a organização para a aplicação integral do *Scrum*, este estado de execução plena do método é inviável dentro da duração do estudo, o qual é referente ao período de estágio do autor na empresa. Portanto, este trabalho não se compromete também a acompanhar a empresa estudada até o alcance desejado da empregabilidade do *Scrum*.

Por fim, ressalta-se que o objetivo deste trabalho é voltado para fins de análise e não de execução de mudanças decorrentes dos resultados do estudo, ainda que possam ser feitas sugestões de melhorias nos processos da empresa.

## 1.5 ESTRUTURA DO TRABALHO

O estudo é dividido em 5 capítulos, dos quais o primeiro se refere a introdução, contextualizando a natureza dos métodos ágeis e em específico o impacto do *Scrum* no setor de TI atualmente. Além disso são abordados os problemas relacionados à aplicação da metodologia, os objetivos deste trabalho, as motivações de execução da pesquisa e suas delimitações, e por fim, a estrutura.

Em seguida, o segundo capítulo apresenta a revisão da literatura, com o objetivo de apresentar em maiores detalhes os principais conceitos abordados ao longo do trabalho, seguindo a sequência lógica das fases do estudo. Os temas deste capítulo referem-se aos conceitos básicos das metodologias ágeis e do *framework Scrum*, introduzindo e detalhando sua metodologia. Na sequência, é apresentado os dados de pesquisas mais recentes sobre o uso dos métodos ágeis e do *Scrum* em empresas de desenvolvimento de *software* e por fim, são

detalhados os anti-padrões encontrados na aplicação do *Scrum*, as motivações por traz destes desvios e possíveis soluções para as empresas nestas situações.

No terceiro capítulo, é apresentado o conjunto de métodos e ferramentas de pesquisa utilizados para a realização deste estudo, descrevendo o roteiro de entrevistas e também a natureza de uma pesquisa-ação e como este trabalho se enquadra neste viés, conforme a Associação Brasileira de Engenharia de Produção (ABEPRO).

O quarto capítulo inicia com apresentação dos resultados das entrevistas realizadas e a análise geral e discussão dos padrões identificados e pontos importantes levantados pelos entrevistados. Em seguida, apresenta-se os resultados da pesquisa-ação, descrevendo os projetos e equipes observados durante o período de estudo, o método e processos de gestão de projetos adotado pela empresa, as principais ações de suporte ao *Scrum* efetuadas e a análise e discussão dos eventos ocorridos. Por fim, foram agrupadas todas as causas diretas de anti-padrões no *Scrum* da empresa sob estudo, tanto identificadas nas entrevistas como na pesquisa-ação, e então representadas as relações destas causas com os anti-padrões e conceitos *Scrum*.

Por último, o quinto capítulo apresenta as considerações finais do trabalho, concluindo o que foi exposto ao longo dos outros 4 capítulos e a informações pertinentes para trabalhos futuros. Na sequência, encontram-se a lista de referências do estudo e o apêndice A, onde constam os eventos e datas observados na pesquisa-ação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Através de uma revisão da literatura, esta seção tem o objetivo de esclarecer as principais ideias abordadas neste trabalho e estabelecer o embasamento teórico para o desenvolvimento do estudo. Esta seção apresenta, respectivamente, os conceitos básicos e dados atuais das metodologias ágeis e em específico o *Scrum*, e os desvios que as empresas praticam nos processos do *Scrum*, detalhando seus contextos e ainda possíveis soluções.

### 2.1 METODOLOGIAS ÁGEIS

Existem muitos métodos de desenvolvimento de *software* tradicionais, como as abordagens de projetos em cascata, iterativos, incrementais, em espiral, evolutivo, etc. Estas abordagens mais clássicas são chamadas de métodos planejados de desenvolvimento de software ou métodos pesados, as quais são bastante úteis para desenvolver um *software* extenso e complexo, onde podem ajudar a eliminar aspectos informais do desenvolvimento e entregar um produto de alta qualidade de maneira sistemática, atendendo todos os requisitos do usuário e em um tempo pré-determinado (AL-SAQQA; SAWALHA; ABDELNABI, 2020). O problema é que para utilizar estes métodos pesados, também são necessários processos igualmente pesados, como realizar o plano de todo o projeto antecipadamente, definir todos os requisitos, desenvolver o design para todos estes requisitos para então iniciar a construção do código e por fim testar todo o *software*, a fim de verificar se o que foi construído atende os requisitos levantados no início do projeto (AL-SAQQA; SAWALHA; ABDELNABI, 2020).

Segundo Al-Saqqa et al. (2020), o desenvolvimento ágil de *software*, ou simplesmente *Agile*, é uma abordagem leve que foi proposta para suprir as limitações dos métodos tradicionais e reduzir as despesas e custos operacionais enquanto promove flexibilidade para adotar mudanças nos requisitos do produto em qualquer momento. Estas características desejadas estão relacionadas aos resultados frustrantes que atuantes no desenvolvimento de *softwares* encontravam em meados da década de 90, principalmente nas fases iniciais dos projetos, onde a documentação “completa” da lista de requisitos, seguida dos *designs* de alta fidelidade e de arquitetura de um produto tão complexo, tornam-se praticamente impossíveis de serem feitas de maneira antecipada com precisão e clareza, pois partem da premissa que o cliente sabe exatamente o que e como construir. Além disso, gera-se desta forma grandes quantidades de retrabalho, advindos da necessidade natural de mudanças no projeto em andamento.

Ainda conforme Al-Saqqa et al. (2020), a maioria dos métodos pesados apresentaram grandes problemas, especialmente na manutenção e nas mudanças baseadas em pedidos dos usuários. Algumas destas mudanças podem gerar grandes alterações no produto, o que altera todo o planejamento do projeto e desenvolve assim uma reação em cadeia de incertezas e falta de definição quanto ao trabalho a ser feito, resultando em problemas críticos para o projeto. Desta forma, tornou-se necessária a introdução de métodos mais “leves”, onde o foco deveria estar em acelerar o desenvolvimento e reagir de forma eficiente às mudanças exigidas (AL-SAQQA; SAWALHA; ABDELNABI, 2020). Vale a pena ressaltar que a transição dos métodos mais pesados para os ágeis não aconteceu de forma repentina, mas gradual, à medida que novas melhorias foram sendo implementadas e novos métodos surgindo. Além disso, os métodos ágeis tendem a funcionar melhor com equipes menores, o que se tornou cada vez mais comum no mercado, e os tradicionais em equipes grandes, onde a comunicação precisa ser mais rigorosa e onde se possuem mais recursos para aplicar estes processos. Assim, sob este cenário do *boom* das tecnologias de *software* das últimas décadas, em que os clientes se tornam cada vez mais incapazes de definir previamente todas as necessidades e exigências do seu produto a ser criado, diversos atuantes do setor desenvolveram de forma independente métodos e práticas para atender à inevitável mudança que eles estavam vivenciando (COHEN; LINDVALL; COSTA, 2003).

O ponto de virada do movimento migratório para os métodos ágeis ocorreu no inverno de 2001, em Utah, EUA, onde 17 representantes dos mais novos métodos ágeis de desenvolvimento de *software* da época se encontraram para discutir as práticas e o novo rumo que esta área estava tomando. Deste encontro, surgiu o Manifesto para Desenvolvimento Ágil de Software, que serve até hoje como guia para nortear valores e características de quaisquer trabalhos que se denominem *agile*.

O Manifesto inicia com 4 situações onde são necessárias tomadas de decisão sobre prioridades (BECK et al., 2001):

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Em seguida, o documento apresenta 12 princípios, que podem ser simplificados conforme Domingues (2018):

1. Satisfação do cliente em 1º lugar;
2. Abraçar a mudança em qualquer etapa do projeto;
3. Entregas incrementais frequentes (semanalmente);
4. Times multidisciplinares formados por pessoas técnicas e de negócio;
5. Time motivado e marcado por uma relação de confiança;
6. Comunicação por meio de conversas cara a cara;
7. Entregas funcionais são a principal medida de progresso do projeto;
8. Colaboração constante entre clientes, usuários, patrocinadores e time;
9. Construir tudo com qualidade técnica e bom design;
10. Simplicidade é maximizar a quantidade de trabalho não feito;
11. Equipes auto-organizadas e autônomas;
12. O time reflete constantemente sobre formas de se tornar melhor.

A partir deste manifesto, confeccionado pelos mais experientes e bem sucedidos profissionais de desenvolvimento de *software* da época, o crescimento e a adesão ao uso da filosofia ágil cresceram exponencialmente pelo mundo. Hoje em dia, os métodos, ou melhor, as abordagens ágeis mais conhecidas e utilizadas são o *Kanban*, *Scrum*, *SAFe (Scaled Agile Framework)*, *LeSS (Large Scale Scrum)*, *XP (Extreme Programming)*, *DevOps* e *Crystal*. Em resumo, podemos afirmar que as metodologias ágeis surgiram de métodos pesados mais antigos com características iterativas, as quais são fundamentais no *Agile*. É possível, na Figura 1, compreender melhor seu princípio básico de iteratividade ao se comparar um projeto ágil com um projeto tradicional em cascata:

Figura 1: Comparação entre modelo Cascata e modelo Ágil de projeto.

### CASCATA (Waterfall)



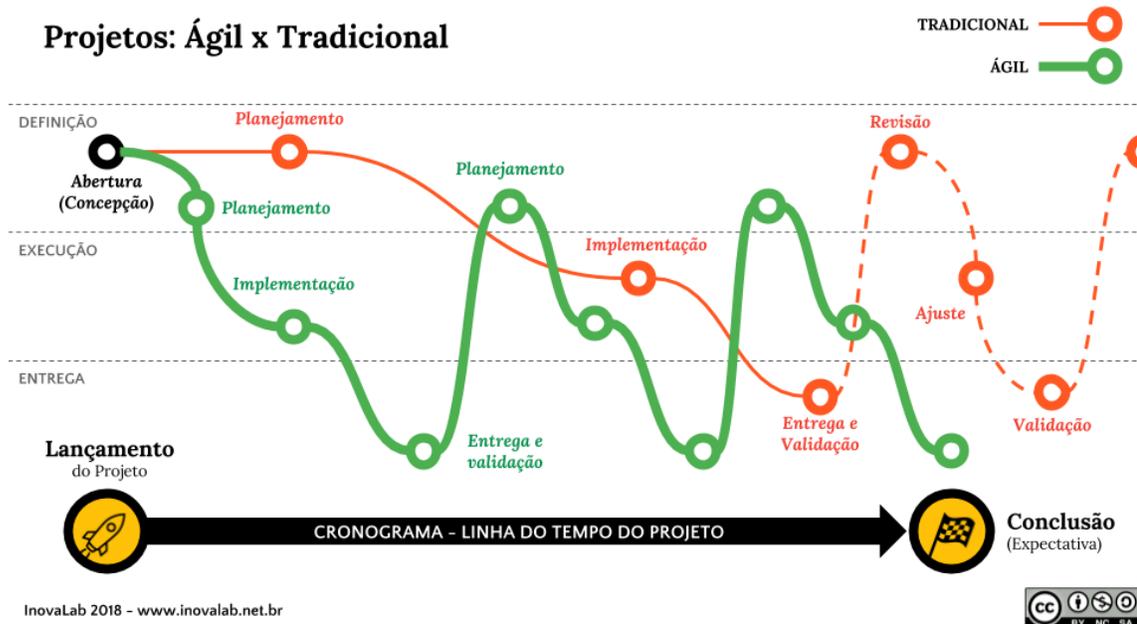
### ÁGIL (Agile)



O grande diferencial entre estes tipos de projetos está na adoção de diversos ciclos iterativos que ocorrem em curtos períodos de tempo no *agile* (geralmente dentro de um mês),

como pode ser visto na Figura 2. Desta forma, adquire-se a importante característica de flexibilidade e adaptabilidade a quaisquer mudanças no escopo do projeto.

Figura 2: Comparação de linha do tempo entre projeto ágil e tradicional



Com a capacidade de realizar entregas menores e em curtos períodos de tempo, é preciso que se priorize o que será entregue ao cliente, desta forma, também são adotadas as práticas de MVPs (Mínimo Produto Viável), os quais Domingues (2018) resume como possuindo três funções:

1. Entregar valor aos clientes e usuários ao longo do projeto (e não somente no fim);
2. Ser um instrumento de *feedback* e aprendizagem em direção à melhoria contínua (entregar o melhor produto possível);
3. Ser a medida primária de progresso do projeto.

Outra característica interessante comentada por Domingues (2018), sobre os métodos ágeis, está relacionada à curva de aprendizagem do projeto. Esta curva refere-se ao aprendizado do cliente e da equipe sobre o próprio produto sendo criado, onde independente de métodos pesados ou leves, esta curva é, na teoria, igual para todo tipo de projeto. No modelo tradicional de desenvolvimento, no início do projeto durante a fase de planejamento, o nível de conhecimento desta curva de aprendizado é mínimo, o qual começa a se tornar mais maduro

durante a execução e somente ideal na entrega, onde não há mais tempo hábil ou orçamento disponível para grandes mudanças ou implementações não previstas na fase de planejamento. Por outro lado, em um projeto ágil, a cada iteração torna-se possível aplicar mudanças imprevistas no início do projeto, pois sempre haverá uma nova fase de planejamento, execução e entrega, em que a cada ciclo, espera-se um nível mais alto na curva de aprendizado e por consequência maior qualidade nas entregas e menor retrabalho.

Desta forma, atualmente torna-se indispensável e primordial a utilização de um projeto de tipo ágil para desenvolvimento de *software*, pois diversos fatores podem atrapalhar e até mesmo inviabilizar um projeto complexo como este. As principais, mas não únicas, causas estão relacionadas à grande dificuldade de listar os requisitos necessários para o produto, priorizar o que deve ser feito primeiro e a comunicação eficiente entre equipes e com o cliente.

Entretanto, cada metodologia tem suas vantagens e desvantagens, de forma em que não há uma metodologia ótima para todos os tipos de projetos e suas especificidades. Assim, escolher o método ágil ideal para um projeto de desenvolvimento, deve ser feito com extremo cuidado, baseando-se em todas as variabilidades encontradas. (AL-SAQQA; SAWALHA; ABDELNABI, 2020).

## 2.2 MÉTODO *SCRUM*

Neste tópico foram levantadas as etapas, práticas, filosofia e características que constituem o método ágil mais conhecido e utilizado no mundo, o *Scrum*. Segundo Kussunga e Ribeiro (2019), o *Scrum* é uma abordagem de gerenciamento de projetos que não necessita da especificação técnica global do projeto concluída, como nos métodos tradicionais, e é uma metodologia para gerenciar o desenvolvimento de sistemas da informação, o que enfatiza entregas rápidas nos processos de desenvolvimento.

Para que a descrição do método neste trabalho reflita de forma fiel as ideias de seus criadores, todas as informações no restante deste tópico foram extraídas da versão mais atualizada do Guia *Scrum* (2020), dos autores Ken Schwaber e Jeff Sutherland, autores também do Manifesto para Desenvolvimento Ágil de Software (2001).

Primeiramente, é interessante enaltecer o aviso trazido pelos autores no início do Guia *Scrum*: “Cada elemento do *framework* serve a um propósito específico que é essencial para o valor geral e os resultados obtidos com *Scrum*. Mudar o modelo central ou ideias do *Scrum*, remover elementos ou não seguir as regras do *Scrum*, encobre os problemas e limita os

benefícios do *Scrum*, potencialmente até tornando-o inútil.” (SCHWABER; SUTHERLAND, 2020, p.1). Tendo isto em vista, na sequência afirmam que é possível e bem-vindo o contrário, acrescentar elementos como padrões, processos e insights ao método, pois o uso do *Scrum* pode variar de acordo com o contexto, entretanto estas variações não foram abordadas no guia.

Ainda sobre as características gerais do método, o framework é propositalmente incompleto, definindo somente as partes necessárias para implementar a teoria *Scrum*, ou seja, em vez de fornecer às pessoas instruções detalhadas, as regras do Guia *Scrum* orientam seus relacionamentos e interações. (SCHWABER; SUTHERLAND, 2020). Desta forma, o *Scrum* instiga melhorias que partem de uma base comum, mais sobre princípios e uma filosofia pré-determinados do que processos engessados, os quais possuem sua eficácia à mercê do ambiente e de cada situação.

### **2.2.1 Filosofia *Scrum***

Antes de descrever as etapas e processos do método, sobre a filosofia do *Scrum* emprega-se: “*Scrum* é baseado no empirismo e *lean thinking*. O empirismo afirma que o conhecimento vem da experiência e da tomada de decisões com base no que é observado. O *lean thinking* (pensamento enxuto) reduz o desperdício e se concentra no essencial. *Scrum* emprega uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar o risco. *Scrum* envolve grupos de pessoas que, coletivamente, possuem todas as habilidades e conhecimentos necessários para fazer o trabalho e compartilhar ou adquirir essas habilidades conforme necessário.” (SCHWABER; SUTHERLAND, 2020, p.3).

A iteratividade do método se dá pela execução do evento chamado *Sprint*, que é na prática o período de tempo de execução das tarefas e segundo os autores do guia “o coração do *Scrum*”. Conforme o guia, o sucesso das *Sprints* é proveniente da implementação dos pilares empíricos do *Scrum*: transparência, inspeção e adaptação. Além disso, o método possui valores que o time deve perseguir para que os pilares empíricos se tornem realidade, estes valores são: Compromisso, foco, abertura, respeito e coragem.

### 2.2.2 *Scrum Team*

Em seguida, define-se qual é a equipe ideal para *Scrum*, descrevendo os diferentes papéis e responsabilidades de seus integrantes. Segundo Schwaber e Sutherland (2020), o *Scrum Team* é um pequeno grupo de profissionais de 10 ou menos pessoas, focado na meta do produto, multifuncional e autogerenciável, sem hierarquia e responsável por todas as atividades relacionadas ao produto, desde a colaboração com stakeholder, verificação, manutenção, operação, experimentação, pesquisa e desenvolvimento, e qualquer outra coisa que possa ser necessária.

Vale ressaltar que os *Scrum Teams* devem procurar permanecer pequenos quanto ao número de integrantes, para que a comunicação seja eficiente e ágil, porém grande o suficiente para que a entrega de valor ao produto no fim das *Sprints* seja considerável e satisfatória. Os papéis do time podem ser divididos entre *Developers*, *Product Owner* e *Scrum Master*.

***Developers:*** São aqueles que de fato criam e agregam a cada *Sprint* valor direto ao produto, e segundo o Guia *Scrum* são responsáveis por:

- Criar um plano para a *Sprint*, o qual consiste em uma lista de tarefas, chamado *Sprint Backlog*;
- Introduzir gradualmente qualidade aderindo a uma Definição de Pronto;
- Adaptar seu plano a cada dia em direção à meta da *Sprint*; e,
- Responsabilizar-se mutuamente como profissionais.

Na prática, estes integrantes não são exclusivamente programadores, os quais usualmente são chamados de desenvolvedores, os *Developers* podem ser programadores, mas também analistas de testes, designers, entre outros.

***Product Owner:*** Este papel é designado exclusivamente para uma pessoa no time, o qual deve manter a integridade conceitual do trabalho desenvolvido nas *Sprints*, criando a partir da demanda passada pelo cliente, histórias (funcionalidades do produto) que devem ser criadas pelos *Developers*. O *Product Owner* deve planejar e garantir a qualidade das entregas do time e sua decisão sobre o que constitui o *Backlog* é final.

O *P.O.*, como comumente o *Product Owner* é chamado, é responsável pelas seguintes funções, a quais ele tem o direito de delegar:

- Desenvolver e comunicar explicitamente a meta do produto;
- Criar e comunicar claramente os itens do *Product Backlog*;
- Ordenar os itens do *Product Backlog*; e,
- Garantir que o *Product Backlog* seja transparente, visível e compreensível.

“Para que os *Product Owners* tenham sucesso, toda a organização deve respeitar suas decisões. Essas decisões são visíveis no conteúdo e na ordem do *Product Backlog* e por meio do incremento inspecionável na revisão da *Sprint*.” (SCHWABER; SUTHERLAND, 2020, p.6).

***Scrum Master***: “O *Scrum Master* é responsável por estabelecer o *Scrum* conforme definido no Guia do *Scrum*. Eles fazem isso ajudando todos a entender a teoria e a prática do *Scrum*, tanto no *Scrum Team* quanto na organização.” (SCHWABER; SUTHERLAND, 2020, p.6).

O *Scrum Master* procura garantir a melhoria contínua dos processos e facilitar a realização das tarefas do restante da equipe. Segundo o Guia *Scrum* (2020), ele possui as seguintes funções:

Quanto ao *Scrum Team*:

- Treinar os membros do time em autogerenciamento e funcionalidades cruzadas;
- Ajudar o *Scrum Team* a se concentrar na criação de incrementos de alto valor que atendem à Definição de Pronto;
- Provocando a remoção de impedimentos ao progresso do *Scrum Team*; e,
- Garantir que todos os eventos *Scrum* ocorram e sejam positivos, produtivos e mantidos dentro do *Timebox* (período de tempo máximo pré-estabelecido).

Quanto ao *Product Owner*:

- Ajudar a encontrar técnicas para a definição eficaz de meta do Produto e gerenciamento do *Product Backlog*;
- Ajudar o *Scrum Team* a entender a necessidade de itens do *Product Backlog* claros e concisos;
- Ajudar a estabelecer o planejamento empírico do produto para um ambiente complexo; e,
- Facilitar a colaboração dos *stakeholders*, conforme solicitado ou necessário.

Quanto à organização (empresa):

- Liderar, treinar e orientar a organização na adoção do *Scrum*;
- Planejar e aconselhar implementações de *Scrum* dentro da organização;
- Ajudar os funcionários e os *stakeholders* a compreender e aplicar uma abordagem empírica para trabalhos complexos; e,
- Remover barreiras entre *stakeholders* e *Scrum Teams*.

### 2.2.3 Eventos *Scrum*

A cada *Sprint*, o ciclo iterativo do *Scrum* passa por certos eventos, os quais resumem-se em encontros da equipe que possuem o objetivo de aplicar a melhoria contínua nos processos do *Scrum* e do trabalho do *Scrum Team* em geral. “Cada evento no *Scrum* é uma oportunidade formal para inspecionar e adaptar os artefatos do *Scrum* (Trabalho ou valor empregado). Esses eventos são projetados especificamente para permitir a transparência necessária. A falha em operar quaisquer eventos conforme prescrito resulta em oportunidades perdidas de inspeção e adaptação. Os eventos são usados no *Scrum* para criar regularidade e minimizar a necessidade de reuniões não definidas no *Scrum*. O ideal é que todos os eventos sejam realizados no mesmo horário e local para reduzir a complexidade.”(SCHWABER; SUTHERLAND, 2020)

#### 2.2.3.1 *Sprint*

A *Sprint* pode ser caracterizada como o principal evento do *Scrum*, a qual na verdade é um período de tempo contendo todos os outros eventos de planejamento, execução e análise desta e outras *Sprints*. As etapas relacionadas à *Sprint* estão representadas na Figura 3.

Figura 3: Representação da *Sprint* e seus eventos.



Fonte: Espinha (2020)

Uma *Sprint* deve durar menos que um mês e geralmente é maior que 2 semanas, visando obter um período de tempo que as funcionalidades do produto desenvolvidas sejam completas, sem necessidade de quebra desta entrega em mais *Sprints* ou a realização de muitas

entregas em um período muito longo, portanto, segundo Schwaber e Sutherland (2020), esta definição do *Timebox* procura criar consistência, e além disso:

Durante a *Sprint*:

- Nenhuma mudança é feita que coloque em risco a meta da *Sprint*;
- A qualidade não diminui;
- O *Product Backlog* é refinado conforme necessário; e,
- O escopo pode ser esclarecido e renegociado com o *Product Owner* conforme mais é aprendido.

De acordo com o Guia *Scrum* (2020), ao analisarmos a estrutura de uma *Sprint*, podemos considerar cada *Sprint* um projeto curto, o qual fornece previsibilidade ao garantir a inspeção e adaptação em direção à meta do produto pelo menos uma vez ao mês. Esta garantia é assegurada através dos eventos *Scrum* de *Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint Retrospective*.

#### 2.2.3.2 *Sprint Planning*

A *Sprint Planning* inicia a *Sprint* ao definir em conjunto com todo o time, o trabalho a ser realizado na *Sprint*. O *Product Owner* garante que os participantes estejam preparados para discutir os itens mais importantes do *Product Backlog* e como eles são mapeados em relação à Meta ou planejamento do Produto (SCHWABER; SUTHERLAND, 2020).

A discussão dos pontos em uma reunião da *Sprint Planning* deve seguir a ordem dos seguintes tópicos:

- Por que esta *Sprint* é valiosa?
- O que pode ser feito nesta *Sprint*?
- Como o trabalho escolhido será realizado?

Neste evento, o *Product Owner* deve trazer um *Backlog* bem definido, mostrar o valor e garantir com que o time compreenda o quê e como o trabalho deve ser feito. Já o time deve definir a meta da *Sprint* e definir em discussão com o *P.O.* o quê do *Backlog* deve ser incluído na *Sprint*. Vale ressaltar que o maior desafio aqui, é estimar o quanto de trabalho o time consegue realizar até o fim da *Sprint*, a maturidade da equipe e o histórico do time são essenciais nesta etapa, para que se consiga evitar erros na quantidade de trabalho entregue ao fim da *Sprint*.

Quanto à estimativa de carga de trabalho necessária para entregar os requisitos do produto, as práticas em cada organização variam. O *Scrum* define que somente os *Developers*

devem estimar o trabalho de desenvolvimento, excluindo assim o *Product Owner* e *Scrum Master*. Entretanto, é comum entre as praticantes do *Scrum*, a realização de reuniões que visam estimar horas ou pontos de carga de trabalho para suas tarefas. A mais comum destas reuniões é o *Poker Planning*, onde dividem-se os requisitos em tarefas menores e mensuráveis, a quais os *Developers* votam, em grupo e secretamente, as estimativas para cada tarefa, resultando por fim em requisitos subdivididos em tarefas menores e estimadas, facilitando o planejamento e controle da *Sprint*.

### 2.2.3.3 Daily Scrum

A *Daily* é um evento de 15 minutos de duração, que ocorre todos os dias de trabalho. Estas reuniões visam manter o controle e ritmo do que está sendo desenvolvido na *Sprint*, onde geralmente todos os participantes contam o que desenvolveram desde a última *Daily*, o que estão fazendo, se possuem quaisquer impedimentos ou dificuldades, e o que farão até a próxima *Daily*, além de facilitar e incentivar a comunicação.

Ainda assim, os participantes tem a liberdade de escolher qualquer estrutura e técnicas que quiserem, desde que seu *Daily Scrum* foque no progresso em direção a Meta da *Sprint* e produza um plano de ação para o próximo dia de trabalho, o que cria foco e melhora o autogerenciamento. As *Daily Scrums* otimizam a comunicação, identificam os impedimentos, promovem a rápida tomada de decisões e conseqüentemente eliminam a necessidade de outras reuniões (SCHWABER; SUTHERLAND, 2020).

### 2.2.3.4 Sprint Review

“O propósito da *Sprint Review* é inspecionar o resultado da *Sprint* e determinar as adaptações futuras. O *Scrum Team* apresenta os resultados de seu trabalho para os principais *stakeholders* e o progresso em direção a Meta do Produto é discutido.”(SCHWABER; SUTHERLAND, 2020, p.9).

Este evento está mais relacionado ao trabalho desenvolvido e a evolução do produto em si. Segundo Schwaber e Sutherland (2020), estas reuniões devem se limitar no máximo a 4 horas de duração para um *Sprint* de 1 mês, no caso de *Sprints* menores, devem seguir a mesma lógica. Além disso, os participantes devem procurar não tornar este evento uma mera

apresentação dos resultados, mas aproveitar a oportunidade para gerar discussões e debates visando estabelecer melhorias quanto ao produto e ao *backlog* de requisitos.

#### 2.2.3.5 *Sprint Retrospective*

Apesar dos significados dos nomes semelhantes, a *Sprint Review* e a *Sprint Retrospective* diferenciam-se no objetivo do que está sendo revisado e discutido. O propósito da *Sprint Retrospective* é planejar maneiras de aumentar a qualidade e a eficácia da maneira que trabalham, inspecionando como foi a última *Sprint* em relação a indivíduos, interações, processos, ferramentas e sua Definição de Pronto. (SCHWABER; SUTHERLAND, 2020).

Neste evento, os participantes discutem o que ocorreu de forma satisfatória ou além do esperado durante a *Sprint*, quais dificuldades tiveram e como resolveram os problemas encontrados. Também é um momento para discutir o domínio técnico dos *Developers* e suas dificuldades nas mais variadas áreas que afetam seu dia de trabalho.

Através destas discussões, devem ser elencados e priorizados os pontos de melhoria para a próxima *Sprint*, reforçando uma espécie de plano de ações para a realização destas melhorias. A *Sprint Retrospective* marca o último evento do ciclo iterativo do *Scrum*, finalizando assim a *Sprint*.

### 2.3 DADOS SOBRE A APLICAÇÃO DE AGILE E SCRUM

Para melhor entender o estado atual das práticas e técnicas *Agile* no mundo, este trabalho baseou-se nos mais recentes dados do *15th State of Agile Report (2021)*, última edição da principal pesquisa referente às práticas ágeis, a qual é patrocinada pela empresa Digital.ai e contou com 4182 respostas de praticantes em empresas de *software* em mais de 100 países. Vale enfatizar que após um ano desde o início da pandemia da COVID-19, mais do que nunca as organizações estão agora cientes das oportunidades e desafios a frente, é preciso ter característica de agilidade no desenvolvimento e entrega de software de sucesso na era digital, assim como na estratégia de negócios e execução do operacional (*15th State of Agile Report, 2021*). Desta forma, o relatório analisa as empresas sob este novo cenário de grandes mudanças em que as organizações se encontram.

Baseado no fato de que a drástica migração do trabalho presencial para o remoto em 2020 impactou diretamente no ritmo e na qualidade das equipes de projetos ágeis, é necessário

compreender as tendências relacionadas a este fenômeno. Segundo o *15th State of Agile Report* (2021), apenas 3% das empresas pretendem retornar para o escritório de forma integral, 25% afirmam que continuarão totalmente remotos após a COVID e 56% são a favor de um sistema híbrido, frequentando o escritório regularmente, mas não todos os dias. Desta forma, de acordo com o *15th State of Agile Report* (2021) esta mistura entre trabalho remoto e híbrido promoverá um desafio para as empresas, que no momento, já estão acelerando a adoção de novos processos, práticas e tecnologias para apoiar mudanças nos produtos e na entrega dos serviços, decorrentes de pontos críticos e falhas encontradas durante a pandemia.

O *15th State of Agile Report* (2021) indica que neste último ano, o crescimento na adoção de métodos ágeis passou de 37% em 2020 para 86% em 2021. É válido ressaltar que o *Scrum* representa 66% dos métodos ágeis identificados na pesquisa e outros 15% são identificados como derivados do *Scrum*, como *Scrumban* e *Scrum/Xp*. Além disso, outra grande diferença notável entre os anos de 2020 e 2021 foi esta adoção do *Scrum* saltar de 44% para 66%, diferente da tendência de outros métodos como o *XP*, conhecido pela rigurosidade de seus processos, que vem diminuindo ao longo dos anos de 25% dos respondentes para apenas 1%.

Entre as principais técnicas e práticas utilizadas pelas organizações, todos os eventos do *Scrum* surgem entre os mais comuns, sendo a *Daily* (87%), *Sprint Retrospective* (83%), *Sprint Planning* (83%) e *Sprint Review* (81%) efetuadas por mais de 80% das empresas. É interessante apontar o crescimento do uso do *Kanban* como técnica empregada para gestão e controle visual de tarefas, de 1% na primeira *State of Agile Report* para 61% em 2021, e a prática de estimar tarefas do time ou *Planning Poker*, presente em 58% das organizações.

Ao analisar as ferramentas de planejamento e entrega para métodos ágeis, as mais utilizadas foram, respectivamente, Quadro *Kanban* (77%), Quadro de Tarefas (67%), Planilhas (66%) e Plataformas Nativas de Gerenciamento de Projetos Ágeis (64%).

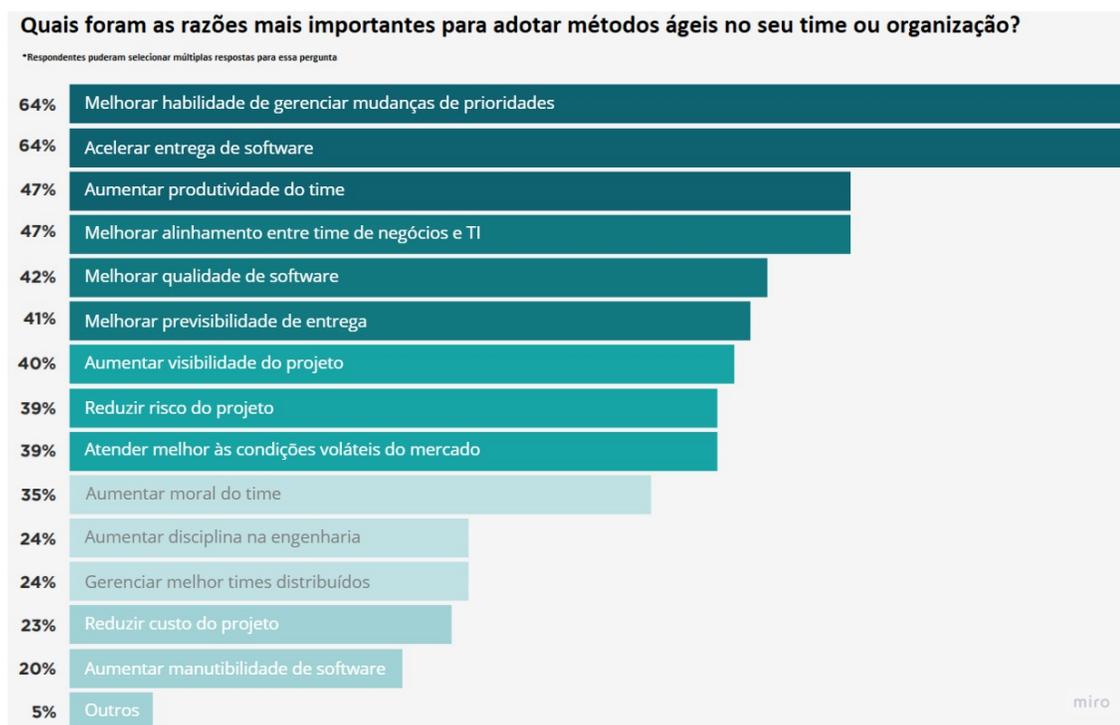
Das ferramentas utilizadas especificamente para gerenciar projetos ágeis, as 3 mais recomendadas são *Atlassian Jira* (81%), *Digital.ai Agility (formerly VersionOne)* (81%) e *Azure DevOps* (66%). Além destas, ainda é comum a utilização de plataformas genéricas como *Microsoft Excel* (32%) e *Google Docs* (48%).

Conforme o *15th State of Agile Report* (2021), as duas razões mais urgentes para adotar o *Agile*, são a velocidade e flexibilidade requeridas para trabalhar em ambientes que continuam sendo tanto imprevisíveis quanto voláteis. Além do mais, ressalta-se que estes motivos sempre estiveram entre os mais citados, entretanto ultimamente tem-se notado que estas empresas cada

vez mais conseguem alcançar esses objetivos, particularmente através de da melhoria de práticas ágeis e boas práticas de desenvolvimento.

Entre as outras principais razões para adoção de metodologias ágeis (Figura 4), destacam-se também as necessidades de aumentar a produtividade do time, melhorar o alinhamento entre os times de negócios e TI, elevar a qualidade do *software*, melhorar a previsibilidade das entregas, melhorar a visibilidade do projeto e reduzir seus riscos.

Figura 4: Principais razões para empresas adotarem metodologias ágeis.



Fonte: Adaptado do *15<sup>th</sup> State of Agile Report* (2021).

Quanto aos impactos positivos da implementação do *Agile* nas organizações (Figura 5), também segundo o *15<sup>th</sup> State of Agile Report* (2021), cerca de 70% destas empresas afirmam que os principais benefícios são administrar mudanças de prioridade, visibilidade e alinhamento entre os times de negócios e TI.

Figura 5: Principais áreas da empresa beneficiadas pela implementação do Agile.



Fonte: adaptado do *15th State of Agile Report* (2021).

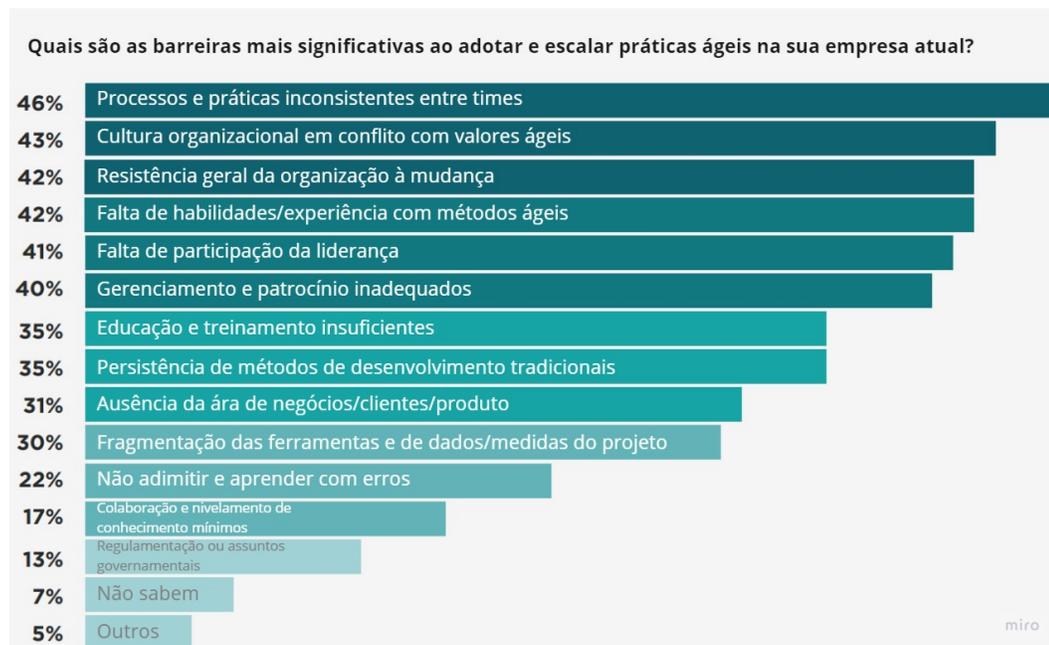
Um fato interessante sobre a confirmação dos benefícios do *Agile*, é retratado ao aferir que metade destas empresas mede o sucesso do *Agile* através de métricas externas, como satisfação do cliente e valor do negócio, adicionalmente, a velocidade de entrega também é citada por 45% das organizações. (15th State of Agile Report, 2021).

Sobre os grandes desafios enfrentados pelas empresas frente à adoção do *Agile*, encontram-se muito motivos diferentes que atrapalham os times ágeis destas organizações. Dos respondentes da *15th State of Agile Report* (2021), 30% afirmam que possuem ao menos 10 obstáculos e desafios diferentes, resultando entre os principais:

1. Inconsistências nos processos e práticas – 46%
2. Divergências culturais – 43%
3. Resistência organizacional geral a mudanças – 42%
4. Falta de habilidades e experiência – 42%
5. Ausência de participação da liderança – 41%
6. Suporte e patrocínio inadequado da gerência – 40%

Agora, ao analisar os principais desafios que as organizações enfrentam ao adotar um método ágil (Figura 6), estes têm permanecido basicamente os mesmos por muitos anos. Os quais resumem-se em obstáculos com processos inconsistentes, cultura organizacional, resistência à mudança, e também falta de suporte e habilidades (15th State of Agile Report, 2021).

Figura 6: Principais barreiras para adotar e escalonar as práticas Agile nas empresas.



Fonte: Adaptado do *15th State of Agile Report* (2021).

Um dos fatores válidos de serem ressaltados, é o fato de que segundo o *15th State of Agile Report* (2021), há alguns anos mais de 80% dos respondentes da pesquisa anual identificavam cultura organizacional como uma barreira significativa à adoção do *Agile*, agora, este número caiu para apenas 40%, o que prova que a cultura das empresas está cada vez mais alinhada com os valores ágeis.

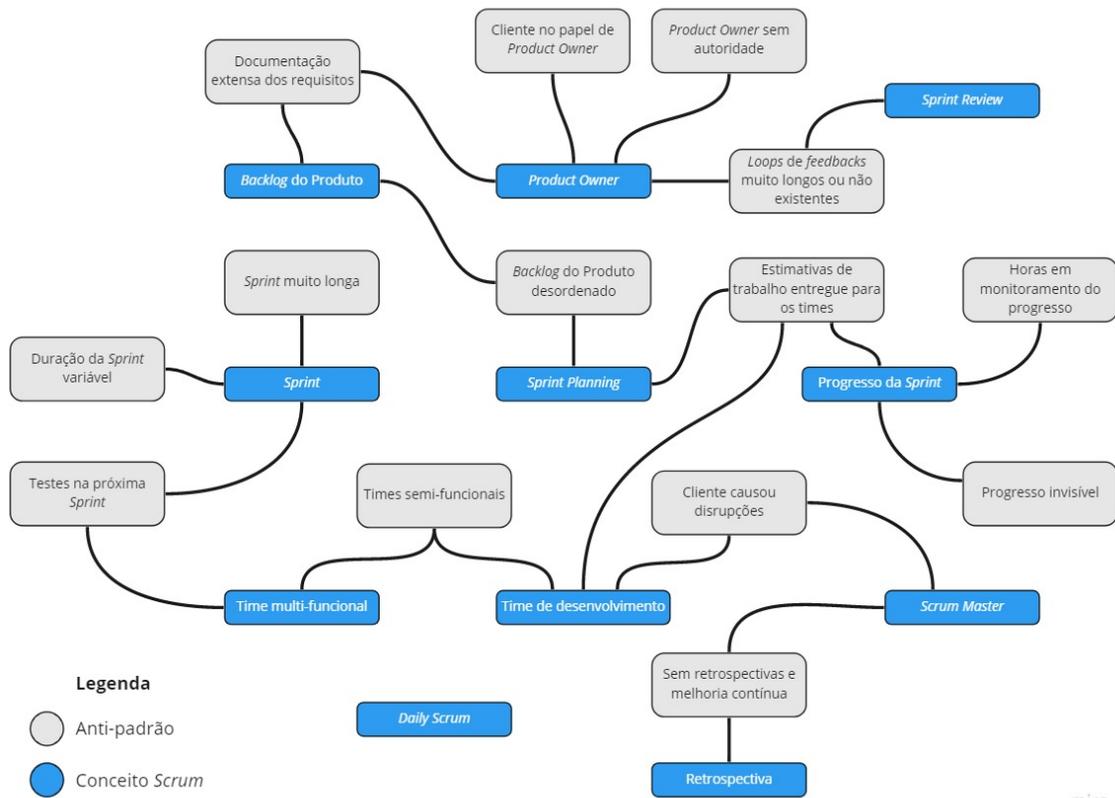
## 2.4 SCRUMBUT

Sabendo-se que a implementação prática do *Scrum* é de fato um enorme desafio, as organizações raramente seguem de forma fiel as diretrizes do Guia *Scrum*, à medida que desviam das práticas padrões do *Scrum* por diversos motivos diferentes. Apesar da vasta adoção e popularização do método ao redor do mundo, quando inspecionadas mais de perto e em maiores detalhes, com a frase “Nós somos *agile* no desenvolvimento de *software*”, as empresas geralmente querem dizer “Nós estamos usando algumas práticas e ideias do *Scrum*”. Omitir algumas práticas do *Scrum* ou alterá-las de alguma forma que não esteja alinhada com as diretrizes do *Scrum* é chamado de *Scrumbut* (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

Este desalinhamento entre a aplicação ideal do método e a realidade prática de adaptações das empresas tem sido culpado como fonte de perda de muitos benefícios do *Scrum*. Desta forma, torna-se necessário compreender como estas práticas ocorrem na realidade, identificando quais partes da metodologia impõem problemas para a indústria. Em particular, entender como e por que as organizações desviam do jeito recomendado de trabalhar com *Scrum*, já que como esta metodologia ágil é um *framework* compreensível ao invés de uma simples coleção de práticas, desvios das recomendações do *Scrum* são, em geral, consideradas potencialmente prejudiciais (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

Através da pesquisa realizada por Eloranta, Koskimies e Mikkonen (2016), a qual teve o objetivo de identificar os principais anti-padrões relacionados aos conceitos encontrados no Guia *Scrum*, foi construída a Figura 7, onde os anti-padrões do *Scrum* encontrados na indústria, assim como os contextos em que ocorrem, exceções e recomendações para possíveis soluções são apresentados.

Figura 7: Relação entre anti-padrões e conceitos Scrum.



Fonte: Adaptação do autor sobre o trabalho de Eloranta, Koskimies e Mikkonen (2016).

Desta forma, estes anti-padrões de *Scrumbut* quando realizados, mostram que a organização ou time não está realizando o *Scrum* de acordo com as premissas dos autores do

método, apesar de que em muitos casos, esta escolha é consciente ou o objetivo por trás do desvio é se adaptar a um contexto que exija certas mudanças a fim de agregar maior valor ao produto ou cliente.

A seguir é apresentada a lista de anti-padrões identificados por Eloranta, Koskimies e Mikkonen (2016):

**Anti-padrão 1:** Documentação extensa de requisitos.

**Recomendações do Scrum:** O *Backlog* do Produto contém todos os itens potenciais que serão implementados e esta lista é ordenada por valor, risco ou dependências por exemplo. Itens podem ser descritos como Histórias do Usuário ou casos de uso que são mais facilmente compreendidas como itens que produzem valor ao usuário final, sob o seu próprio entendimento. **Na prática:** Documentação tradicional extensa é produzida antes do desenvolvimento das *Sprints*, por pensamento “em cascata” do time ou prática enraizada pela empresa. **Consequências:** Requisitos do sistema como um todo não são bem entendidos pelo time nem são ordenados, portanto é difícil para o time saber quais requisitos são mais críticos para o negócio e decidir quais itens devem ser implementados em seguida. Todos os requisitos são reunidos em cima da hora e o trabalho pode ser descartado pois os requisitos podem mudar antes das *features* do produto serem implementadas. **Recomendações para a empresa:** Realizar reuniões de refinamento do *backlog* a cada 2 semanas, onde o *Product Owner* com a ajuda do time cria as Histórias do Usuário para a próxima *Sprint*, mas não para as *Sprints* futuras. Ser ágil não significa abrir mão por completo das especificações, mas documentos extensos em cima da hora demandam muito trabalho, o que nem sempre produz valor para o cliente (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 2:** Cliente no papel de *Product Owner*

**Recomendações do Scrum:** O papel de *Product Owner* é necessário para garantir que o trabalho da equipe produza valor para o cliente. **Na prática:** Um representante do cliente age como *Product Owner*, quando não há um bom entendimento deste papel, falta alguém apto para a função ou o cliente não confia no time. **Consequências:** O cliente pode estar muito ocupado para criar um bom *backlog* de requisitos, atrapalhando o entendimento e fluxo do time, além da comunicação poder ser dificultada por diferenças organizacionais. Neste caso, esta situação pode facilmente ocasionar o anti-padrão Cliente causou disrupções, quando o cliente tenta ditar o quê, como e quando o time deve fazer, onde provavelmente ocorre uma priorização pela

economia no investimento. **Recomendações para a empresa:** Convencer o cliente da importância de existir um *Product Owner* da própria empresa no time, ou ainda criar um time de *Product Owners* onde o *Product Owner* do cliente faça parte (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

#### **Anti-padrão 3:** *Product Owner* sem autoridade

**Recomendações do Scrum:** O *Product Owner* deve gerenciar o *Backlog* e decidir quais itens agregam mais valor para o produto. **Na prática:** Falta conhecimento sobre o papel do *P.O.* e o encarregado não promove as práticas *Scrum*. Assim, geralmente um membro da equipe de desenvolvimento é promovido a *P.O.*, pois os gerentes do produto não se interessam o suficiente para utilizar o *Scrum*. **Consequências:** O *P.O.* não tem autoridade para decidir o que vai ser feito e dessa forma a priorização perde o sentido, onde no fim tudo é implementado. Também pode haver gerentes acima do *P.O.*, cortando o contato direto do *P.O.* com o cliente e impedindo o conhecimento do valor das entregas. **Recomendações para a empresa:** Como a adoção do *Scrum* está incompleta, devido a falta de visão de valor no método, são recomendados treinamentos para os gerentes, a fim de demonstrar os benefícios do papel do *Product Owner* na equipe (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

#### **Anti-padrão 4:** *Loops* de *feedbacks* muito longos ou não existentes

**Recomendações do Scrum:** A *Sprint Review* é organizada no fim de cada *Sprint*, onde o *P.O.* e *stakeholders* chave (cliente) revisam o incremento adicionado ao produto e dão seus *feedbacks*. **Na prática:** Ocorre o anti-padrão do *P.O.* sem autoridade, sem contato com o cliente, onde desta forma este cliente não se interessa pelo projeto, apenas quer ver o produto final. Além disso, como nenhum *feedback* é passado para a equipe, os desenvolvedores não têm certeza se o que estão produzindo está de acordo com as expectativas do cliente. **Consequências:** Mudanças do produto em estágio avançado, pois o *feedback* só acontece quando o cliente de fato utiliza o produto, podendo ocorrer muito retrabalho sobre o que foi desenvolvido. Além disso, o time pode começar a tomar decisões sem a consulta do cliente, podendo resultar no desenvolvimento de *features* que não agregam valor ao cliente. **Recomendações para a empresa:** Introduzir de forma sucinta o *Scrum* e desenvolvimento ágil ao cliente. Assim os clientes começam a entender o ritmo das entregas e desenvolvimento contínuo, eles geralmente gostam e ainda incentivam a sua utilização (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 5:** *Backlog* do Produto desordenado

**Recomendações do Scrum:** Deve ser feita a ordenação do *Backlog* para que haja uma priorização do que agrega mais valor ao produto e deve ser implementado antes. **Na prática:** Já ocorrem os anti-padrões 2, 3 e 4, além da falta de competência do *Product Owner*. Assim, o time ordena por conta própria o *Backlog*, sem a visão do cliente. **Consequências:** Como não há prioridade do que deve ser implementado, pode faltar entendimento do risco ou valor dos elementos a serem desenvolvidos. Além disso, apenas as tarefas mais simples ou mais satisfatórias de serem feitas, na visão dos desenvolvedores, serão desenvolvidas. Tarefas difíceis acabam ficando por último no *backlog*, aumentando o risco de problemas em estágios avançados do projeto. **Recomendações para a empresa:** Realizar reuniões de refinamento do *Backlog* periodicamente, a fim de organizar o *Backlog* e ordenar a *features* do topo desta lista, levando em consideração a relação entre estes itens (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 6:** Estimativas de trabalho entregue para os times

**Recomendações do Scrum:** Os times devem estimar o trabalho por conta própria. **Na prática:** Isso acontece em empresas com prática hierárquicas que definem a necessidade de estimativas a fim de saber com antecedência o custo de desenvolvimento, assim um Gerente de Projetos ou equivalente produz as estimativas sozinho. **Consequências:** Caso as estimativas sejam muito superiores à realidade, haverá falta de comprometimento do time, e caso sejam muito inferiores, poderá haver desmotivação geral. Para que estes erros sejam reduzidos, é necessário que o time estime as tarefas e não alguém que não vai desenvolver a tarefa. **Recomendações para a empresa:** Realizar o *Planning Poker* (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 7:** Horas em monitoramento do progresso

**Recomendações do Scrum:** Estimar com “pontos” da história ao invés de horas. **Na prática:** Utilizam-se as horas de estimativas para gerar valores de custo do projeto ou *features*, assim as histórias são estimadas em horas e não pontos de esforço. **Consequências:** Muito tempo gasto para estimar e os erros de estimativas (que são naturais) têm um impacto maior. **Recomendações para a empresa:** Utilizar pontos da história para estimar, usar a primeira *Sprint* para ter noção da velocidade com que o time consegue entregar os pontos (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 8:** Times semi-funcionais

**Recomendações do Scrum:** Times devem ser multi-funcionais, possuindo todo o conhecimento necessário para desenvolver o produto. **Na prática:** Ocorrem o anti-padrão 14, falta de *expertise* e divisão das equipes baseado nestas diferenças de conhecimento. Assim, o time produz de maneira limitada. **Consequências:** Time menos engajado e motivado, e divisão de responsabilidades. Geralmente ocorre o seguinte fluxo da tarefa: O desenvolvedor cria e manda para o testador, o testador testa, encontra bugs e manda de volta para o desenvolvedor, que então corrige os erros. Este processo com interrupções atrasa a entrega de *feedbacks* e da *feature*. **Recomendações para a empresa:** Testadores e designers de interface no mesmo time (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 9:** Cliente causou disrupções

**Recomendações do Scrum:** O time é protegido de qualquer disrupção externa. **Na prática:** Ocorre o anti-padrão 3 e interação intensa com cliente, onde o cliente consegue se comunicar diretamente com membros do time e ainda pedir a inclusão de novas *features* ou demandas do suporte. **Consequências:** Interrupção do fluxo de trabalho e redução da eficiência do time. Mais trabalho é incluído sem a supervisão do *Product Owner*, sem aparecer no *backlog*, assim tarefas não prioritárias acabam entrando na *Sprint*, e ainda por cima sem supervisão. **Recomendações para a empresa:** Definir uma frequência ou período de tempo com que o cliente pode se comunicar com o time. O *Scrum Master* deve proteger o time de quaisquer fatores externos (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 10:** Sem retrospectivas e melhoria contínua

**Recomendações do Scrum:** Sempre realizar a *Sprint Retrospective* ao fim da *Sprint*. **Na prática:** A empresa e time não tem conhecimento suficiente sobre o *Scrum*, a importância de seus eventos ou ainda de um *Scrum Master*. Desta forma, as retrospectivas não ocorrem, ou quando acontecem são muito curtas, de forma que não são gerados planos de ação para a próxima *Sprint*. **Consequências:** Tende-se a extinguir a melhoria contínua, através da falta de promoção da mesma, podendo assim ocorrer por consequência diversos outros anti-padrões. **Recomendações para a empresa:** Alimentar uma cultura ágil e de melhoria contínua, planejar melhor as retrospectivas e tentar reduzir os ciclos de *feedbacks* (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 11:** Progresso invisível

**Recomendações do Scrum:** O progresso do trabalho deve poder ser visualizado através de gráficos como o *burndown*. **Na prática:** Ocorrem práticas hierárquicas da empresa, quando por exemplo o *P.O.* produz os gráficos para si e não compartilha as informações com o restante do time, ou nem mesmo os produz. **Consequências:** Falta de compreensão do progresso e visão de valor por parte do time. O time não consegue estimar ao certo o trabalho necessário para a próxima *Sprint*, *Stakeholders* podem não enxergar os problemas enfrentados pelo time. **Exceções:** Projetos muito pequenos onde as pessoas se conhecem bem e discutem com facilidade o progresso. **Recomendações para a empresa:** Visualizar o fluxo de trabalho de forma adequada e possuir o(s) gráfico(s) de progresso em local público (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 12:** Duração da *Sprint* variável

**Recomendações do Scrum:** 2 a 4 semanas. **Na prática:** Falta de disciplina e anti-padrões 2 e 9, assim novos itens são inseridos durante a *Sprint*, sua data de término não é pré-definida e se estende para que tudo seja feito. **Consequências:** Falta de comprometimento com objetivos da *Sprint* e dificuldade de visualizar progresso, além de não existir exatidão na previsão de entregas do produto. **Recomendações para a empresa:** Testar diferentes tamanhos de *Sprint* no projeto até o time acordar na melhor duração para o contexto em que estão inseridos. (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

**Anti-padrão 13:** *Sprint* muito longa

**Recomendações do Scrum:** 2 a 4 semanas de duração. **Na prática:** Enquanto a empresa está iniciando a implementação do *Scrum*, já está acostumada com processos em cascata ou ainda os clientes não aceitam ciclos curtos de interação, desta forma os intervalos de desenvolvimento passam de 1 mês. **Consequências:** Perde-se o foco e a eficiência do time quando os prazos são muito prolongados, onde ao invés de permitir que as tarefas sejam entregues com maior folga em relação ao prazo, na prática as equipes tendem a não conseguir entregar essas tarefas completas no fim da *Sprint*. Os intervalos de *feedbacks* muito longos também atrapalham quando o cliente decide mudar as prioridades de entregas e não há como ou tempo hábil para mudar algo que está sendo trabalhado há várias semanas, o correndo o anti-padrão 9. Além disso, a previsibilidade das entregas é afetada quanto mais se estende a duração da *Sprint*. **Exceções:** Em alguns casos o trabalho da equipe depende de fatores externos que

tomam mais tempo que as *Sprints*, podendo ser justificadas ciclos maiores de desenvolvimento.

**Recomendações para a empresa:** Experimentar tamanhos diferentes de *Sprints* a fim qual é mais benéfica para seu contexto (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

#### **Anti-padrão 14:** Testes na próxima *Sprint*

**Recomendações do Scrum:** Desenvolver e testar uma *feature* na mesma *Sprint*, a fim de obter-se um entregável no fim deste período. **Na prática:** O teste pode ocorrer na *Sprint* seguinte devido a pensamento em cascata do time, time de testadores separado do time de desenvolvimento, falta de testes automatizados, anti-padrão 12 ou anti-padrão 13.

**Consequências:** Novos códigos podem ser escritos em cima de código não testado. Além disso, o retrabalhado gerado pelo teste é muito mais custoso pois passam-se dias ou semanas após o desenvolvimento da *feature*, assim quando algum *bug* é encontrado, torna-se muito mais complicado voltar ao ritmo e contexto de desenvolvimento daquele material e arrumá-lo rapidamente, pois aumenta-se a dificuldade cognitiva dos desenvolvedores, que já estão atuando em tarefas novas. Por fim, é necessária também muita comunicação do time para que não haja longas etapas de espera e *vai-e-volta* de tarefas entre desenvolvedores e testadores.

**Recomendações para a empresa:** Parar as tarefas de desenvolvimento de código alguns dias antes do fim da *Sprint*, para que o time possa focar nos testes do que já foi desenvolvido. Para isso é interessante descobrir quantos dias são necessários para testar e considerar finalizadas apenas tarefas que já foram testadas (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

Além da pesquisa de Eloranta, Koskimies e Mikkonen, o artigo *Scrum in practice: an overview of Scrum adaptations*, de Hron e Obwegeser (2018), parte do mesmo pressuposto da generalização de adaptações do *Scrum* em empresas de desenvolvimento de *software*. Entretanto, neste caso foi realizada uma pesquisa na literatura, através de um algoritmo de palavras-chave capaz de agrupar e classificar, entre 31 estudos relevantes, 7 motivações distintas para modificações do método e 6 estratégias genéricas para atingir os resultados esperados pelas modificações:

#### **Motivações:**

1. Trabalho remoto;
2. Combinação com outros *frameworks* ou métodos;
3. Foco crescente em UX (*User Experience*) e usabilidade;

4. Escalonamento vertical – Adoção do *Scrum* em aspectos organizacionais maiores, como planejamento estratégico;
5. Escalonamento em tamanho – *Scrum* em projetos médios e grandes;
6. Ferramentas para usar o *Scrum*;
7. Contexto específico.

### **Estratégias:**

1. Combinação: Uma mistura de elementos do *Scrum* com elementos de outros processos/métodos.
2. Pré-desenvolvimento: Processos, artefatos ou papéis adicionais que lidam com tarefas como arquitetura técnica, articulação da visão do produto ou criação de marcos para o desenvolvimento, antes do desenvolvimento de fato iniciar, ou seja, tarefas preparatórias para o desenvolvimento;
3. Orientação do método: Anotações, instruções e guias sobre como aplicar o método em cenários, contextos ou circunstâncias específicas, ressaltando pontos como aplicação conforme o Guia *Scrum*, lembretes de princípios ágeis até conselhos práticos na forma de orientação de melhores práticas;
4. Procedimentos, artefatos, funções: Mudança ou introdução de novos artefatos, funções ou processos ao *Scrum*.
5. Multiplicidade: Multiplicação de certos aspectos do *Scrum* (Artefatos, processos, funções ou o próprio time). Os elementos multiplicados do *Scrum* podem ser usados para propósitos diferentes. Por exemplo, pode ser sugerido dois *backlogs*, um para desenvolvimento e outro para gestão.
6. Ferramentas: Propostas de ferramentas que não modificam diretamente o *Scrum*, mas ajudam a realizar certas tarefas. Tais ferramentas podem geralmente ser vistas como algum tipo de extensão ao método original, e elas podem ser aplicadas passivamente sem mudar diretamente o funcionamento do método.

Deste modo, diversas modificações e desvios do método são conhecidos como os principais impeditivos quanto a melhores resultados com o *Scrum*, porém além disso, para que se alcance os benefícios esperados das metodologias ágeis, antes de tudo é necessária a otimização do principal recurso de qualquer projeto ágil, as pessoas. Moe e Dingsøyr (2008) exaltam a diferença entre efetividade e produtividade do time, ressaltando que a produtividade está mais atrelada à entrega de tarefas e a efetividade a uma ótica mais holística, levando em consideração a interação do time e dependendo de fatores internos.

Salas et al. (2005) condensou diversos conceitos de trabalho em equipe nos “5 Grandes”, os quais consistem de um *framework* que procura garantir a efetividade da equipe. Os “5 Grandes” componentes que as equipes devem possuir são: Liderança do Time, Monitoramento de Performance Mútuo, Comportamento de *Backup*, Adaptabilidade e Orientação do Time. Além disso, os “5 Grandes” devem ser apoiados por 3 mecanismos de

coordenação: Modelos Mentais Compartilhados, Comunicação em *Loop* Fechado e Confiança Mútua.

O estudo de caso de Moe e Dingsøy (2008) contou com mais de 60 observações de um time *Scrum* completo e visava compreender como o *Scrum* garante na prática um ambiente propício aos “5 Grandes”, e consequentemente a uma melhor efetividade do time. O trabalho baseou-se em duas perguntas para nortear o estudo e obteve os seguintes resultados mais relevantes:

*Como o Scrum suporta os fatores que influenciam o trabalho em equipe efetivo?*

- Liderança do Time: Não é devidamente endereçado no *Scrum*, pois o método apesar de ressaltar que o time deve ser auto-gerenciável e possuir uma liderança difusa e sem hierarquia, não é claro em como fazer isto.
- Comportamento de *Backup*: Definido como a habilidade de antecipar outros membros da equipe, este comportamento não é esclarecido no *Scrum*, apesar deste promover times auto-organizáveis e o Comportamento de *Backup* ser essencial nestes casos. Uma solução bem vista na indústria é a implementação da prática de *pair programming*, onde os *developers* desenvolvem em conjunto uma mesma tarefa, compreendendo assim o modo de pensar e trabalho da sua dupla.
- Confiança Mútua: O *Scrum* não esclarece como desenvolver confiança mútua no time, a qual também é uma característica dos times auto-organizáveis.

*Como podem os desafios ao adotar o Scrum serem explicados pelos fatores influentes sobre trabalho em equipe?*

Segundo Moe e Dingsøy (2008), os principais desvios das práticas recomendadas pelo *Scrum* foram primeiramente a falta de planejamento a longo prazo, lidar com problemas e estabelecer Confiança Mútua. O Monitoramento de Performance Mútuo também não ocorreu devidamente devido à falta de gráficos de acompanhamento da *Sprint*, como o Gráfico de *Burndown* ou *Burnup*. Além disso, a especialização das funções atrapalhou a orientação da equipe, resultando em membros focando somente nas suas tarefas particulares. Por fim, a falta de Confiança Mútua ocasionou em problemas não relatados e falta de auto-organização.

## 2.5 DESDOBRAMENTOS DO REFERENCIAL TEÓRICO

Através deste capítulo, foi possível compreender o que são metodologias ágeis e quais foram os contextos para a criação do método ágil mais famoso atualmente, o *Scrum*. A partir do entendimento da filosofia que norteia *Scrum*, das definições dos papéis de um time *Scrum* (*Product Owner*, *Scrum Master* e *Developers*) e dos eventos *Scrum*, permitiu-se entender o que se espera na teoria de um projeto que utiliza deste *framework*, e então analisar o cenário atual da indústria e seus padrões mais recentes acerca das metodologias ágeis.

Já na seção 2.4, foi introduzido o termo *Scrumbut*, o qual caracteriza em suma os desvios e adaptações realizados pelas empresas que procuram adotar o *Scrum* e também foram trazidos questionamentos sobre como a metodologia oferece suporte ao trabalho em equipe. Desta forma, buscando atender o objetivo específico “Descrever as principais causas de sucesso ou fracasso na implementação do método ágil *Scrum* presentes na literatura”, foi elaborado na Quadro 1 o resumo dos anti-padrões evidenciados nas organizações, possíveis motivações para adaptações no método e estratégias que norteiam estas mudanças.

Quadro 1: Resumo de anti-padrões, motivações e estratégias do capítulo 2.

Anti-padrões		Motivações	Estratégias
Documentação extensa dos requisitos	Times semi-funcionais	Trabalho remoto	Combinação de métodos
Cliente no papel de <i>Product Owner</i>	Cliente causou disrupções	Combinação com outros frameworks ou métodos;	Pré-desenvolvimento
<i>Product Owner</i> sem autoridade	Sem retrospectivas e melhoria contínua	Foco crescente em UX (User Experience) e usabilidade	Orientação do método
<i>Loops</i> de <i>feedbacks</i> muito longos ou não existentes	Progresso invisível	Escalonamento vertical – Adoção do <i>Scrum</i> em aspectos organizacionais maiores	Procedimentos, artefatos, funções novos ou alterados
<i>Backlog</i> do Produto desordenado	Duração da <i>Sprint</i> variável	Escalonamento em tamanho – <i>Scrum</i> em projetos médios e grandes;	Multiplicidade
Estimativas de trabalho entregue para os times	<i>Sprint</i> muito longa	Ferramentas para usar o <i>Scrum</i>	Ferramentas
Horas em monitoramento do progresso	Testes na próxima <i>Sprint</i>	Contexto específico	

Fonte: Autor

### 3 METODOLOGIA DE PESQUISA

Dentro das áreas de conhecimento definidas pela ABEPRO (Associação Brasileira de Engenharia de Produção), este estudo é classificado como pertencente à Engenharia Organizacional, mais precisamente da subárea de Gestão de Projetos. Além disso, este estudo sobre Gestão de Projetos Ágeis através da metodologia *Scrum* encontra-se sob uma ótica de pesquisa exploratória, buscando aprimorar a compreensão do tema abordado e em seguida aplicando uma pesquisa explicativa, visando elencar causas e efeitos, embasados em levantamentos de dados de empresas de TI.

#### 3.1 ETAPAS DA PESQUISA

Iniciando a sistematização das informações necessárias para a confecção deste trabalho, a qual busca atingir o primeiro objetivo específico (Seção 1.2.2), foi redigida a fundamentação teórica no capítulo 2, onde foi possível esclarecer o contexto da metodologia analisada por este estudo, além da teoria por trás da sua aplicação e seu uso prático em organizações, evidenciando as divergências do guia *Scrum* com o mundo real e esclarecendo as principais causas de sucesso e fracasso nestes projetos de desenvolvimento *software* na atualidade.

Após a etapa inicial de revisão da literatura e estudos de caso, este trabalho foi dividido em duas etapas de investigação, a primeira, de entrevistas com colaboradores de diferentes cargos em duas empresas do setor de desenvolvimento de *softwares* (Seção 4.1), e a segunda, referente a uma pesquisa-ação nas mesmas empresas de TI que utilizam o método *Scrum* (Seção 4.2).

##### 3.1.1 Contextualização das Empresas

A Empresa A foi fundada em 2016, e denomina-se como uma *software house*, organização prestadora de diversos serviços relacionados à criação, desenvolvimento e suporte de *softwares*. Com a vasta experiência e antigos contatos e parceiros do fundador da Empresa A, rapidamente foram se concretizando projetos de desenvolvimento de websites e sistemas, onde alguns deles seguem em progresso até hoje. Além dos clientes de longa data, a empresa

já realizou mais de 30 projetos nestes 5 anos e atualmente conta com 12 projetos *Scrum* em andamento, variando suas soluções entre *websites*, sistemas de integração, UX/UI e *Apps*.

Representando um destes clientes de longa data, a Empresa B é a segunda organização sob estudo neste trabalho, a qual viu suas operações realmente evoluírem ao iniciar sua parceria com a Empresa A em 2016. A empresa de tecnologia, com produto de *software* de gamificação e engajamento de colaboradores, possui cerca de 30 funcionários, com times internacionais na área de vendas e negócios, e mão-de-obra da Empresa A na área de tecnologia e desenvolvimento.

O estágio do autor na Empresa A iniciou-se em junho de 2021 e foi realizado na área de projetos da organização, a qual sofreu fortes turbulências nos últimos anos. Com a saída de dois gestores nível sênior, muitos processos e informações importantes foram perdidos na mudança de gestão, deixando a área sem nenhum funcionário qualificado para o cargo de gerenciamento de projetos. No início do mês de julho, com a contratação de dois estagiários e a promoção de um novo gestor para a área de projetos, o qual já atuava há 3 anos na empresa, deu-se partida a um novo período de reestruturação da empresa e seus projetos, procurando primeiro criar meios de coleta de dados e avaliação de desempenho dos times e em seguida implementar melhorias graduais relacionadas à gestão e à implementação do *Scrum*.

6 meses depois, já em dezembro de 2021, com mão de obra disponível e incentivo a novas ideias acerca da organização dos projetos e implementação de métodos e práticas mais eficientes, construiu-se um ambiente de maior segurança e transparência para a empresa e seus clientes, aplicando diariamente pequenas mudanças, que no escopo geral ajudam a transformar e impactar positivamente os resultados nos projetos.

### **3.1.2 Entrevistas**

Na primeira fase, foram realizadas entrevistas com gestores e equipes de desenvolvimento da Empresa A, a qual se denomina como uma *Software House*, termo designado para organizações especializadas em desenvolver diferentes tipos de serviços de TI, como *websites* e aplicativos com sistemas personalizados para os clientes, e da empresa B, a qual possui um time de desenvolvimento para seu produto, uma plataforma online de gamificação de *feedbacks* de funcionários e clientes. As entrevistas tiveram o objetivo de investigar opiniões dos colaboradores quanto ao método *Scrum* e suas particularidades, aplicado aos contextos pessoais de cada entrevistado.

As 8 entrevistas foram guiadas através do roteiro do Quadro 2, dividido em 4 blocos de perguntas abertas e interpretativas. O Bloco 1 busca estabelecer um comparativo do nível de conhecimento geral dos entrevistados sobre métodos ágeis e *Scrum*, o Bloco 2 busca levantar opiniões sobre características positivas do *Scrum*, o Bloco 3 sobre a adoção do método em empresas e qual a dificuldade inerente, e por fim o Bloco 4 tem o objetivo de levantar opiniões gerais do *Scrum*, nas quais é possível extrair características negativas evidenciadas pelos entrevistados.

Quadro 2: Roteiro da Entrevista

<b>Bloco 1 - Conhecimento do respondente sobre o Scrum</b>	
Qual a sua definição de método ágil?	
O que é o <i>Scrum</i> ?	
Qual o seu nível de domínio sobre o <i>Scrum</i> ? (Quadro 3)	
Você acha que o <i>Scrum</i> deve ser seguido 100% ou é desejável a adaptação do método conforme o contexto? E se isso não for recomendado pelo Guia <i>Scrum</i> ?	
<b>Bloco 2 - Características positivas do Scrum</b>	
Qual a principal vantagem que você enxerga ao aplicar o <i>Scrum</i> em um projeto de desenvolvimento de software?	
Como a adoção do <i>Scrum</i> impacta a relação com o cliente?	
<b>Bloco 3 - Viabilidade de implantação do método 100% ou conforme o Guia Scrum</b>	
Em geral, você acha que as empresas adotam 100% do <i>Scrum</i> ? (Se não) Por que não?	
Quais fatores em específico atrapalham na execução do <i>Scrum</i> ?	
Você acha que as empresas sabem como melhorar sua adoção <i>Scrum</i> ?	
<b>Bloco 4 – Opiniões gerais do Scrum</b>	
Qual a principal desvantagem do <i>Scrum</i> ?	
O que você acha dos processos e eventos do método?	
Qual o impacto das ferramentas de gestão no <i>Scrum</i> ?	
O que você acha dos papéis da equipe no <i>Scrum</i> ?	
Qual a sua opinião sobre a relação da cultura da empresa/hierarquia com a performance do <i>Scrum</i> ?	

Fonte: Autor

Quadro 3: Nível de Domínio Scrum - Classificação

<b>Nível de Domínio</b>	<b>Descrição</b>
Básico	Tem pouca experiência prática e pouco conhecimento teórico.
Intermediário	Tem experiência prática e pouco conhecimento teórico.
Avançado	Tem experiência prática e conhecimento teórico

Fonte: Autor

As entrevistas foram executadas de forma remota, através de videoconferência. Além disso, as reuniões foram gravadas, mediante aprovação dos entrevistados. Por fim, após a execução das entrevistas, todas as falas foram sintetizadas em uma análise comparativa entre os colaboradores e uma análise entre o que foi levantado no referencial teórico e as opiniões destas pessoas que atuam com *Scrum* na prática.

### 3.1.3 Pesquisa-ação

Segundo Engel, (2020), uma pesquisa-ação procura intervir na prática do estudo, já no decorrer do próprio processo de pesquisa e não somente como possível consequência de recomendações na etapa final do projeto e é, portanto, uma forma de pesquisa em situações em que o pesquisador também é uma pessoa da prática estudada.

“A pesquisa-ação pode ser aplicada em qualquer ambiente de interação social que se caracterize por um problema, no qual estão envolvidos pessoas, tarefas e procedimentos”. (ENGEL, 2000, p.183). Também segundo Engel (2000), entre as principais características deste tipo de pesquisa, destaca-se a necessidade do processo fornecer aprendizagem para todos os participantes e existir a separação entre sujeito e objeto de pesquisa, nesta lógica, sugere-se a utilidade dos dados para os clientes e ainda é estabelecido que “as modificações introduzidas na prática são constantemente avaliadas no decorrer do processo de intervenção e o *feedback* obtido do monitoramento da prática é trazido em modificações, mudanças de direção e redefinições, conforme necessário”. (ENGEL, 2000, p.184).

Busca-se através da pesquisa-ação neste trabalho, atingir o último objetivo específico apresentado na seção 1.2.2, “Analisar a implementação do método ágil Scrum em uma empresa de TI”. A partir da observação da rotina da Empresa A, onde o autor atuou como estagiário, buscou-se contrastar as práticas da empresa com o “Guia *Scrum*” e comparar a prática da organização com os levantamentos resultantes dos desdobramentos do referencial teórico (Seção 2.5). Ao longo das interações diárias, foram discutidas intensivamente com os colaboradores as razões das adaptações que ocorriam ou pudessem ocorrer e seu impacto nas características do *Scrum* na empresa, o que contribuiu para a compreensão dos fatores críticos e impeditivos à aplicação método.

Considerando o caráter iterativo da pesquisa-ação, seguindo na segunda fase deste trabalho, foi realizado um registro cronológico dos principais acontecimentos referentes à

evolução da aplicação do *Scrum* na empresa e em seus projetos, entre julho de 2021 e dezembro de 2021. Este acompanhamento foi realizado diariamente, através do estágio realizado pelo autor na Empresa A, o qual se iniciou em junho de 2021. Os eventos descritos nesta seção foram registrados através de notas pessoais e registros de documentos e conversas dos times de projetos. As ideias, mudanças e problemas relacionados ao *Scrum* na organização, seguiram naturalmente o ciclo encontrado na Figura 8, e foram analisados na etapa final de investigação da empresa, sob a ótica crítica da metodologia ágil. Assim, a pesquisa caracteriza-se em uma abordagem quali-quantitativa, baseada em dados primários e secundários.

Figura 8: Etapas da pesquisa-ação



Fonte: Costa et al. (2014)

Com todas as informações necessárias levantadas no referencial teórico e nas entrevistas, foi realizada a análise e discussão da pesquisa-ação, identificando os eventos e situações mais relevantes na evolução dos projetos da empresa e causas adicionais diretas de anti-padrões no *Scrum*, além do impacto, de forma geral, destas nos projetos da empresa.

Por fim, através do levantamento de causas diretas de anti-padrões nas entrevistas e também na pesquisa-ação, foi possível construir um esquema gráfico de relacionamento destas causas com seus anti-padrões e conceitos *Scrum* por meio do *website* Miro.

## 4 RESULTADOS

Neste capítulo encontram-se os resultados das entrevistas realizadas com membros de diferentes times *Scrum*, identificando nos 4 blocos de perguntas as opiniões mais comuns e relevantes, além de apresentar a análise e discussão dos padrões encontrados nas opiniões dos respondentes. Na sequência, apresenta-se os resultados da pesquisa-ação, a qual primeiramente descreve a empresa sob estudo, detalhando componentes primordiais do ambiente de trabalho, como projetos e suas equipes, os métodos e processos aplicados pelos times *Scrum* e as principais mudanças e acontecimento impulsionadores de melhorias na aplicação do *Scrum* na empresa.

### 4.1 ENTREVISTAS

Para os fins desta análise da aplicação do *Scrum* na prática e sob diferentes óticas e vivências, é válido ressaltar a escolha de diferentes níveis hierárquicos para os 8 entrevistados. Além da divisão entre 5 colaboradores na Empresa A (Prestadora de serviço *software house*, com diversos projetos) e 3 na empresa B (Trabalham em um único produto nativo, próprio da organização).

A seguir, o Quadro 4 agrupa os dados gerais dos entrevistados N°1 ao N°8. Em azul estão representados os dados primários, referentes à empresa onde trabalham, o identificador e o cargo. Já em branco, encontram-se os dados secundários, como formação acadêmica, experiência no mercado de *software*, experiência com *Scrum*, experiência com métodos tradicionais de projetos e o tempo de atuação na empresa citada.

Estas informações foram primordiais para o levantamento de padrões e melhor entendimento do contexto e origem das opiniões dos respondentes, tendo em vista que além das conexões mais óbvias com estes dados, os próprios participantes recorrentemente atribuíam suas falas e motivos a experiências passadas.

Quadro 4: Informações dos Entrevistados

Empresa	Nº	Cargo	Formação Acadêmica	Experiência no Mercado (Tempo total e cargos)	Experiência com Scrum	Experiência Método Tradicional	Tempo na Empresa
Empresa A	1	CEO / Diretor	Ciências da Comp. e MBA Gestão da Tec.	25 anos – Dev., Analista de Req. Gestor e Ger. de Proj.	Cursos e prática implementando em empresas	Apenas cursos	Fundou há 5 anos
	2	Gerente de Projetos	Sistemas da Informação	4 anos - Testador	Faculdade, cursos e prática	Há alguns anos (2) na Empresa A	4 anos
	3	Analista Requisitos	Eng. Da Computação	8 anos – Suporte, QA e Analista de Requisitos	Cursos e prática	Na última empresa	7 meses
	4	Analista de QA	Análise e Dev. De Sistemas	8 anos aprox. – Estágio, Dev., QA e P.O.	Apenas na Empresa A	Na empresa antiga	3 anos e meio
	5	Desenvolvedor Pleno	Sistemas da Informação (em curso)	2 anos – Estagiário, Dev. Júnior	Algo na faculdade	N/A	2 anos aprox..
Empresa B	6	<i>Product Owner</i>	Sistemas de Info. e Ciência de Dados	13 anos – Gestão de Projetos	Cursos, certificações e anos de prática	Na prática durante 5 anos	2 meses
	7	Desenvolvedor Sênior	Análise e Dev. De Sistemas	13 anos – Desenvolvedor e Gestor	Curso e prática	7 anos aprox..	4 meses
	8	Desenvolvedor Pleno	Análise de Dev. De Sistemas	4 anos - Dev. Júnior	Algo na faculdade e na empresa antiga	No primeiro emprego	8 meses

Fonte: Autor

A seguir estão descritas de forma sintetizada as opiniões dos entrevistados acerca das perguntas do roteiro da entrevista.

#### 4.1.1 Bloco 1 - Conhecimento do respondente sobre o Scrum

Para as duas primeiras perguntas, “Qual a sua definição de método ágil?” e “O que é Scrum?” foi notória a diferença entre os entrevistados com cargos menos gerenciais para o restante. Os desenvolvedores plenos e o analista de QA (*Quality Assurance*) apresentaram respostas mais simples e confusas ao tentar diferenciar Scrum após definir metodologia ágil, além de citarem alguns eventos Scrum na resposta para a primeira pergunta. Estes três definiram métodos ágeis como maneiras de organização do time que visam velocidade e qualidade, já o restante dos entrevistados trouxe definições mais elaboradas e embasadas, focando na capacidade de reação do time, com foco em correção rápida de erros, sob quaisquer mudanças de contexto. Da mesma forma, para a definição do Scrum, estes 5 focaram em termos mais amplos como filosofia, *framework* e metodologia, ao passo que os outros 3 trouxeram exemplos de eventos específicos ou responderam da mesma maneira que na primeira pergunta.

A terceira pergunta visava ajudar a esclarecer o nível de domínio do Scrum para cada respondente, desta forma, esta classificação, apresentada no Quadro 2, foi efetuada através do consenso entre entrevistador e entrevistado acerca de competências teóricas e práticas. O

Quadro 5 apresenta os níveis de domínio da metodologia *Scrum* assim definidos em conjunto com os participantes.

Quadro 5: Nível de Domínio - Entrevistados

Nº Entrevistado	Cargo	Nível de Domínio
1	CEO / Diretor	Avançado
2	Gerente de Projetos	Intermediário
3	Analista Requisitos	Avançado
4	Analista de QA	Intermediário
5	Desenvolvedor Pleno	Básico
6	<i>Product Owner</i>	Avançado
7	Desenvolvedor Sênior	Avançado
8	Desenvolvedor Pleno	Básico

Fonte: Autor

Para a última pergunta do Bloco 1, “*Você acha que o Scrum deve ser seguido 100% ou é desejável a adaptação do método conforme o contexto? E se isso não for recomendado pelo Guia Scrum?*”, obteve-se um consenso interessante em que sempre deve-se adaptar o método à realidade, e nunca o contrário. Além disso, foi notável a maior confiança nesta ideia nos mesmos entrevistados com cargos mais gerenciais e por consequência com mais experiência, os quais deram exemplos de adaptações e contextos, com resposta firmes quando mesmo a mudança não for recomendada pelo Guia *Scrum*.

#### 4.1.2 Bloco 2 - Características positivas do *Scrum*

Para a pergunta “*Qual a principal vantagem que você enxerga ao aplicar o Scrum em um projeto de desenvolvimento de software?*”, obteve-se em síntese as seguintes respostas, em ordem decrescente das categorias mais enfatizadas:

- Adaptabilidade a mudanças, agilidade na tomada de decisão e previsibilidade;
- Planejamento e controle do trabalho e entregas, através de visibilidade, transparência e priorização;
- Engajamento e comunicação da equipe;
- Rápido aprendizado.

Já para a pergunta “*Como a adoção do Scrum impacta a relação com o cliente?*”, onde já se esperavam respostas no mínimo neutras ou positivas, pois o método ágil *Scrum*

baseia-se no Manifesto Ágil, que define como seu primeiro princípio “Satisfação do cliente em 1º lugar”. Entretanto, apesar de todos responderem que pode haver um grande impacto, 6 dos 8 entrevistados alertaram que são necessários certos requisitos para que o impacto seja positivo, e ainda que a tendência é que o impacto que o *Scrum* emprega na relação entre time e cliente seja geralmente neutra ou até negativa. Foram listadas as seguintes razões que podem prejudicar a relação com cliente neste contexto:

- Cliente não se importa com detalhes do desenvolvimento, não faz parte do processo e apenas espera resultados;
- Cliente não entende o porquê de a metodologia empregar certos processos ou porque ela é utilizada;
- Falta de comunicação entre time e cliente;
- *Scrum* mal empregado pelo time, refletindo de forma transparente para o cliente.

Assim, caso estas situações não ocorram, todos os entrevistados concordaram que há um impacto positivo, seja de forma indireta através da qualidade e eficiência da entrega, ou através dos *loops de feedback* com o cliente, visibilidade, velocidade na tomada de decisão, evolução incremental e bom relacionamento que o *Scrum* bem aplicado proporciona.

#### **4.1.3 Bloco 3 - Viabilidade do método 100% conforme o Guia *Scrum***

Para que não houvesse grandes divergências em interpretações para esta pergunta, foi explicado para os entrevistados o aviso dos autores do Guia *Scrum*, comentado no Capítulo 2.2 deste trabalho, sobre o quão indispensáveis são todos os elementos do método apresentados no guia e ainda como também são bem-vindas adições e incrementos ao método conforme o contexto, definindo assim como “100%” a aplicação sem remoções dos artefatos, papéis e eventos padrão do *Scrum*.

Em resposta à pergunta “*Em geral, você acha que as empresas adotam 100% do Scrum? E caso não, por quê?*”, foi unânime a opinião de que as empresas, em geral, não adotam o *Scrum* em “100%”, tanto por não conseguirem como as vezes por nem mesmo quererem. Além disso, enfatizaram que em nenhum caso a dificuldade ocorre devido a processos ou elementos do método, e sim devido às empresas e suas pessoas. As seguintes razões para tal foram levantadas:

- Cultura da empresa e falta de foco em melhoria de processos;
- Membros da equipe não especializados;

- Priorização de horas de desenvolvimento e teste sobre horas em eventos *Scrum*, devido a prazos curtos, baixo orçamento para horas do projeto e/ou equipe limitada;
- Falta de educação de todos os *stakeholders* acerca do método;
- Falta o papel do *P.O.* ou este é compartilhado por vários agentes;
- Falta de estrutura e estabilidade do produto.

Em seguida os entrevistados responderam se acreditam que as empresas sabem como melhorar a adoção do *Scrum*. As respostas, apesar de serem fortemente influenciadas por experiências individuais, acabaram de certa forma sendo complementares entre si, podendo-se afirmar como uma resposta geral dos 8 entrevistados que caso existam encarregados aptos para os papéis previstos pelo *Scrum*, sabe-se o que é preciso fazer para alcançar melhorias conforme o contexto da organização, porém esta busca pode ser inefetiva ou ainda interminável.

#### **4.1.4 Bloco 4 – Opiniões gerais do *Scrum***

As perguntas realizadas neste bloco tinham como intuito inicial investigar características negativas do método, porém conforme o Bloco 3, todas as causas de insucesso de aplicação do método foram atribuídas a fatores externos da metodologia. Desta forma, o Bloco 4 descreve opiniões gerais sobre elementos e impactos do *Scrum*, onde vale ressaltar que os aspectos negativos levantados não representam grandes empecilhos à aplicação do método, mas são de certa forma inconveniências sob a ótica de certos participantes das entrevistas.

Respondendo à pergunta “*Qual a principal desvantagem do Scrum?*”, todos os respondentes que atuam com gestão e análise de requisitos afirmaram não existir desvantagem, já os desenvolvedores e testador (com experiência em desenvolvimento) levantaram a difícil curva de aprendizado da aplicação nas equipes e da visão de valor do cliente, e novamente as complicações com produtos instáveis e priorização entre horas de desenvolvimento e de eventos *Scrum*.

Quanto às opiniões sobre o impacto da ferramenta de gestão na performance do time *Scrum*, com exceção do Diretor da Empresa A, todos acreditam que o impacto é imenso, de tal forma que o entendimento do trabalho passado e atual, e a produtividade estão direta e fortemente atrelados à escolha da ferramenta de gestão. Já para o Diretor da Empresa A, quem também possui o maior tempo de experiência no mercado, a ferramenta em si independe, pois com poucos recursos pode-se compartilhar o trabalho em andamento da equipe, garantir comunicação efetiva, e assegurar a conformidade sob o pilar de transparência do *Scrum*, como

é o caso de equipes em modelo de trabalho presencial, que utilizam de anotações e cartões colados em paredes.

Em relação aos papéis do *Scrum*, todos concordam que se aplicados e distribuídos corretamente são excelentes para as entregas e qualidade do time, entretanto metade dos respondentes ressaltou o quão improvável este cenário é na prática. Quanto aos 3 diferentes papéis do *Scrum*, foram comentados:

- *Product Owner*: Os entrevistados nº 1 e 7 comentaram sobre a dificuldade de separar as atribuições do *P.O.* e as do *Scrum Master*, e ainda, tanto os entrevistados nº 5 quanto o nº 7, os quais atuam em empresas distintas, afirmaram que o cliente no papel do *Product Owner* (anti-padrão 2) pode ser algo proveitoso.
- *Scrum Master*: O entrevistado nº7 constatou grande insatisfação com o papel do *Scrum Master*, atrelando a este um desperdício de investimentos e processos de monitoria desgastantes e desnecessários. Já a respondente nº 8 afirmou a necessidade do ambiente e restante da equipe do *Scrum Master* estarem funcionando em harmonia e de maneira eficiente para que ele possa executar seu trabalho.
- *Developers*: Sem comentários.

Por fim, foi discutida em maiores detalhes a relação entre cultura e hierarquia da empresa com a performance do método *Scrum*. Todos concordaram que a relação é total e estes estão intimamente interligados, e ainda que o principal direcionador é a cultura, sob a qual as decisões tomadas devem obter total apoio da gerência. Foram levantadas as seguintes características da cultura e hierarquia que fomentam uma melhor performance nas equipes *Scrum*:

- Cultura de inovação;
- Ótima comunicação e abertura para resolução de problemas e conflitos;
- Favorecimento ao erro, não punir erros;
- Pensamento em time, trabalho em equipe;
- Apoio da empresa e gestão como um todo;
- Comprometimento com a mudança;
- Valores da empresa aliados aos valores ágeis.

#### 4.1.5 Análise geral e discussão sobre as entrevistas

Após o término das entrevistas, tornaram-se claros alguns aspectos gerais relacionados às opiniões deste grupo heterogêneo de participantes. Apesar dos diferentes níveis de conhecimento teórico sobre metodologias ágeis e o *Scrum*, a vivência empresarial dos entrevistados parece nortear a maior parte de suas opiniões, independentemente do Nível de

Domínio apresentado no Bloco 1. Entretanto foi notável que os entrevistados menos experientes apresentavam a tendência de associar eventos específicos do *Scrum* às suas respostas, atribuindo mais exemplos práticos do que os mais experientes, os quais por outra via buscavam relativizar as respostas e entender melhor as definições propostas nas perguntas.

Outro fator interessante apontado pelas entrevistas foi a unanimidade das opiniões acerca da essencialidade e benefícios do método. Durante a modelagem do roteiro de entrevistas, foi previsto que haveriam características negativas atribuídas aos processos do método, sua filosofia ou quaisquer elementos que o Guia *Scrum* propõe. Todavia, foi constatado que todos os respondentes acreditam no *framework* e suas eventuais complicações decorrem exclusivamente da incapacidade das equipes.

Estes fatores humanos foram amplamente discutidos nos Blocos 3 e 4, onde são vistos em maioria as dificuldades gerenciais e na relação com o cliente. Podemos associar este fato aos 14 anti-padrões definidos no tópico 2.4, onde, com exceção apenas do anti-padrão 1, todos estão associados a problemas gerenciais e/ou com o cliente. Desta forma, as entrevistas corroboram com a proposta de Eloranta, Koskimies e Mikkonen (2016). Vale ressaltar que todos os anti-padrões em questão estão associados às atribuições dos papéis de *Product Owner* e *Scrum Master*, o que segue de acordo com as opiniões encontradas no Bloco 4, onde apresentaram-se diversas dúvidas sobre as definições das funções de ambos e diferentes cenários relatados em experiências passadas dos entrevistados.

Além disso, também foi constatada a semelhança de respostas deste trabalho com os resultados do *15th State of Agile Report* (2021). O levantamento do relatório sobre as principais barreiras para adotar e escalonar as práticas *Agile* nas empresas contempla todas as dificuldades de implantação do *Scrum* apontadas nas entrevistas, e além destas, todos os benefícios do método listados neste trabalho também foram contemplados no levantamento de principais áreas da empresa beneficiadas pela implementação do *Agile* e nas razões mais importantes para se adotar métodos ágeis nos times e organizações.

## 4.2 RESULTADOS DA PESQUISA-AÇÃO

Antes de descrever a evolução do *Scrum* na empresa sob estudo, vale ressaltar a divisão conceitual das análises realizadas nessa seção. Os tópicos 4.2.1 e 4.2.2 referem-se à descrição da organização e seus padrões de trabalho. A seção 4.2.3 apresenta algumas mudanças que ocorreram na empresa com vistas a melhor suportar o método *Scrum*. Já a seção 4.2.4 analisa

os eventos entre os meses de julho e dezembro de 2021, listada no Apêndice A, analisando sua relação com os padrões do *Scrum*.

#### 4.2.1 PROJETOS E EQUIPES

Ao longo do período de acompanhamento da empresa, excluindo pequenas demandas de suporte e assessoria, os seguintes projetos da Empresa A, apresentados no Quadro 6, estavam em desenvolvimento. São 9 projetos ao total, sendo que 4 deles, em destaque azul no Quadro 6, foram acompanhados de perto, além disso, o mesmo quadro apresenta a letra identificadora de cada projeto, a descrição destes, os cargos e quantidade de integrantes na equipe *Scrum* do projeto e por fim a data aproximada de início de operação.

Dos projetos em que se encontra um potencial de análise e possíveis resultados atrativos ao objetivo deste trabalho e seção, destacam-se os projetos B, E, F e G, os quais o autor participou de forma mais ativa no desenvolvimento e presenciou de perto a evolução de seus contextos. Vale ressaltar que diversos colaboradores atuam em múltiplos projetos paralelamente. Também é interessante pontuar que o projeto B retrata a Empresa B, descrita na seção 3.1.1 e abordada na seção 4.1. Além disso, as equipes híbridas pontuadas, retratam times compostos em parte por colaboradores da Empresa A em tempo integral no projeto e também por profissionais das empresas clientes.

Quadro 6: Projetos da Empresa A

Projeto	Descrição	Equipe	Iniciou há
A	Sistema de intranet para busca de dados.	1 Desenvolvedor 1 Analista de Req. 1 Designer 2 Testadores 1 Gestor de Proj.	2 semanas
B	Plataforma internacional de <i>gamification</i> de ações de colaboradores em empresas.	Equipe Híbrida: 4 Desenvolvedores 1 Testador 1 <i>Product Owner</i> 1 CTO	5 anos
C	Sistema de denúncias em organizações.	1 Desenvolvedor 1 Analista de Req. 1 Testador 1 Gestor de Proj.	2 anos
D	<i>Software</i> internacional de mentorias.	Equipe Híbrida: 3 Desenvolvedores 1 Testador	2 anos
E	Plataforma internacional de votações.	1 Desenvolvedor 1 Analista de Req.	2 meses

		2 Testadores 1 Gestor de Proj.	
F	Sistema de apoio à adequação de leis	2 Desenvolvedores 1 Testador 1 Gestor de Proj.	10 meses e encerrou há 3 meses
G	Plataforma internacional de serviços legais	2 Desenvolvedores 1 Analista de Req. 1 Testador	3 anos
H	Aplicativo de transações financeiras	1 Desenvolvedor 1 Analista de Req. 1 Testador	6 meses
I	Aplicativo de apoio a estudos	1 Desenvolvedor 1 Analista de Req. 1 Testador	2 meses

Fonte: Autor

A seguir, a evolução dos projetos acompanhados é descrita de forma resumida.

## PROJETO B

Este projeto é o primeiro e assim o mais antigo da Empresa A. Considerada uma empresa “irmã”, este cliente compartilha de recursos e mão-de-obra com a Empresa A, formando atualmente uma equipe híbrida, com um desenvolvedor, um testador e o autor deste trabalho como função de apoio no projeto. Ao longo dos anos, o produto deste cliente evoluiu de forma não planejada e com muita rotatividade, resultando hoje em grandes esforços para implementar novas funcionalidades e entender o que as equipes antigas construíram em termos de código de programação. Na prática, a maior evidência de necessidade de melhorias eram a *Sprints* com entregas incompletas, que necessitavam se estender por muitos dias subsequentes.

Durante o período de estudo, o passo mais importante dado pela equipe foi perceber a necessidade de serem mais ágeis, e assim muito mais essencial do que os processos sendo evoluídos, foi a mudança de postura do *CFO* e líder da equipe de desenvolvimento, o qual deixou propositalmente de tomar decisões sozinho e focar em problemas que tomam muito tempo e trazem poucos resultados. Além disso, a chegada do *Product Owner* trouxe padronização aos processos técnicos, reforço na filosofia ágil e estrutura para os eventos *Scrum*, além da divisão de sobrecarga de definições de *features* e requisitos. Desta forma, no fim do período de estudo deste trabalho, já se enxerga nitidamente a evolução na capacidade de entrega das *Sprints*, decorrente de comunicação mais efetiva e melhor entendimento dos requisitos, a níveis técnicos e de negócio.

## PROJETO E

Apresentando dificuldades logo no início das conversas de definição entre cliente e Empresa A, este projeto testou o quão ágil e adaptável este time *Scrum* conseguia ser. Foram diversos obstáculos iniciais, que já lançavam sobre o curto prazo de 12 semanas para a primeira entrega um ar de preocupação na equipe. Não era previsto o trabalho de um analista de requisitos para refinar os detalhes, o gestor e os desenvolvedores estavam sobrecarregados com outros projetos e havia a barreira da diferença de idiomas entre time e cliente. Além disso, já nas semanas iniciais ocorreu a troca de gestor do projeto, onde o autor deste trabalho assumiu este papel sem experiência prévia, o desenvolvedor recém chegado à empresa para assumir o papel neste projeto foi dispensado de suas funções e o prazo foi reduzido de 12 para 8 semanas.

As saídas encontradas para conseguir realizar uma entrega de qualidade e dentro do prazo foram atribuir o papel de desenvolvedor ao *CEO* da Empresa A, utilizar de horas adicionais e concisas do Analista de Requisitos para refinar detalhes não especificados pelo cliente e focar na comunicação com o mesmo, revendo e priorizando diversas vezes o escopo do projeto, de acordo com a capacidade de entrega da equipe a cada semana. Durante as *Sprint Retrospectives* ficou claro o progresso na adaptação da equipe aos fatores externos ao trabalho, de forma que as preocupações dos membros foram diminuindo e a satisfação do cliente aumentando. Hoje, o desenvolvimento está dentro do prazo e a definição do que será entregue alinhada com o cliente, considerando-se assim um projeto promissor.

## PROJETO F

Este projeto é considerado como um remanescente das más práticas ágeis aplicadas antigamente pela Empresa A. No início do período de estudo, o projeto já era considerado como fadado ao fracasso, devido à falta de definição de requisitos iniciais e conseqüente falta de entendimento do time sobre o que deveria ser construído, falta de precisão em estimativas gerais para a primeira entrega, gradativa falta de confiança e alinhamento com cliente, e *Sprints* com duração de mais de 2 meses.

Diante deste contexto, o projeto resultou em entregas parciais fora das expectativas do cliente e por fim em uma entrega final focada em realizar correções e finalizar, com meses de atraso e incrementos não necessários, o mínimo entregável, que fora acordado nos primeiros encontros comerciais com o cliente.

## PROJETO G

Representando o segundo cliente de mais longa data, o Projeto G representa uma grande história de sucesso na Empresa A, de forma que o cliente foi lentamente educado à cultura ágil e hoje está intimamente ligado ao trabalho do time *Scrum*, do qual seus membros trabalham integralmente no projeto. Apesar de também existir a barreira da língua e grande diferença de fuso horários, mantém-se uma comunicação diária com o cliente, fator crítico de sucesso nas *Sprints*.

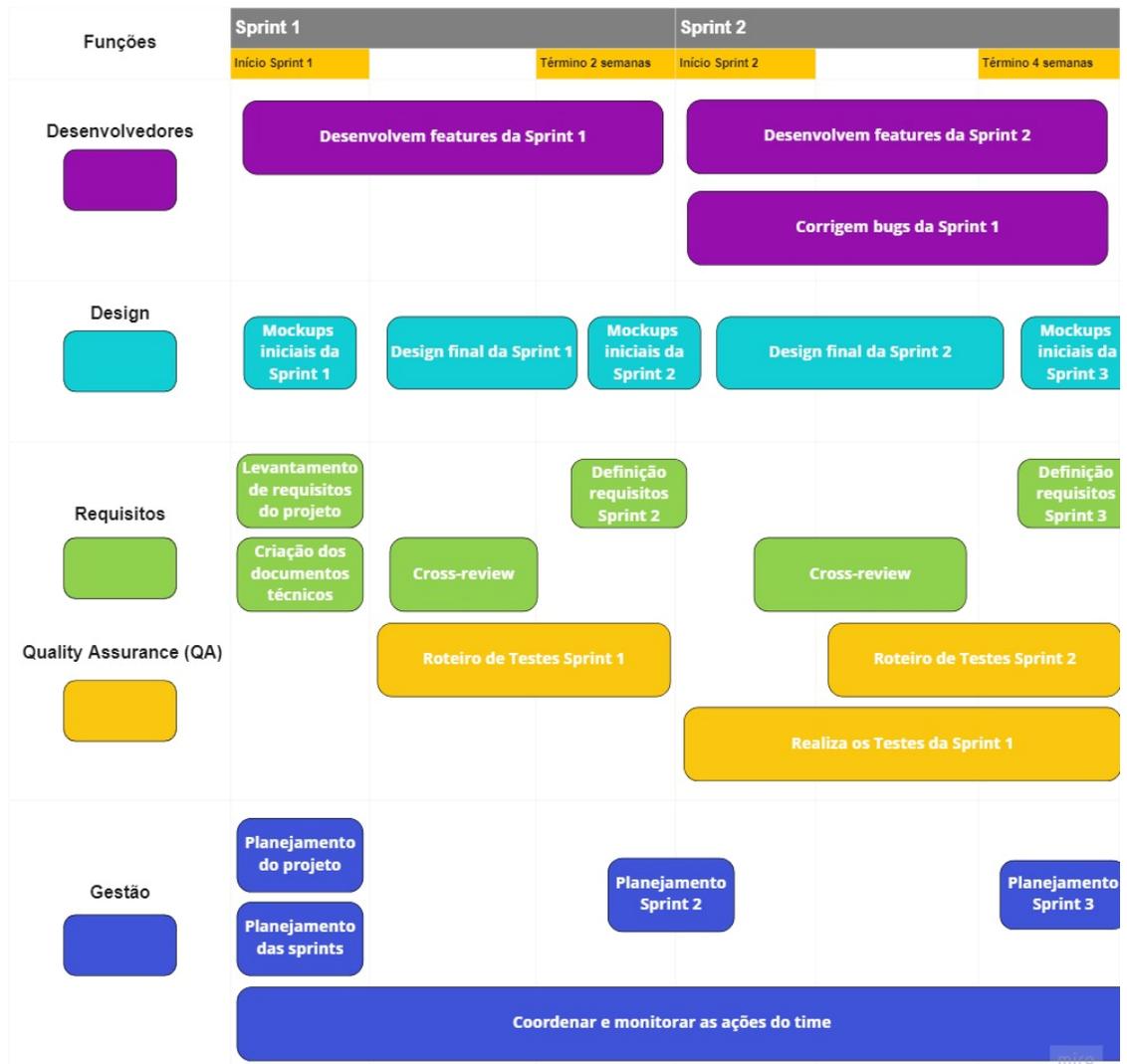
No decorrer da evolução dos processos ágeis na equipe do Projeto G, as principais adaptações ao *Scrum* podem ser vistas na divisão de *Sprints* realizadas pela equipe, de tal forma que existem 3 *Sprints* simultâneas, uma focada na definição de requisitos, outra no desenvolvimento e outra nos testes. Tal adaptação ocorreu de forma natural, sem noção de que isto é uma forma de desvio do método (anti-padrões 13, ao se medir o tempo de entrega das tarefas, e 14), baseada, porém, principalmente na ótima sinergia dos envolvidos. Além disso, também guiada pela ótima comunicação do time, outra consequência e desvio dos padrões definidos pelo Guia *Scrum* é a simplificação e encurtamento dos tempos dos eventos *Scrum*.

### 4.2.2 MÉTODO DE GESTÃO ADOTADO PELA EMPRESA A

Tendo em vista que nenhum projeto da Empresa A segue exatamente os mesmos processos, pois estes acabam adaptando-se à necessidade e contexto, existe, contudo, um padrão que os gestores buscam implementar nas *Sprints*. Além de sempre buscarem realizar todos os eventos ágeis, o seguinte esquema (Figura 9) representa a linha do tempo das entregas padrão de cada membro do time de desenvolvimento, lembrando que os papéis de *Product Owner* e *Scrum Master* não fazem parte deste padrão, pois procura-se suprir suas funções através do Gerente de Projetos (o que não é recomendado pelo Guia *Scrum*). Também é válido ressaltar que o esquema não inclui as tarefas para estimativas e nem tarefas operacionais de revisão e submissão de códigos, e também que o termo *Cross-review* retrata revisões das tarefas dos

desenvolvedores, e o termo *Mockup* representações gráficas e instruções para o que deverá ser desenvolvido.

Figura 9: Funções padrão para cada função nas *Sprints*.



Fonte: Autor.

Ainda que este modelo tenha sido elaborado de acordo com os projetos de sucesso e com a maneira convencional que a Empresa A vem trabalhando com o *Scrum*, pode-se pontuar alguns fatores críticos.

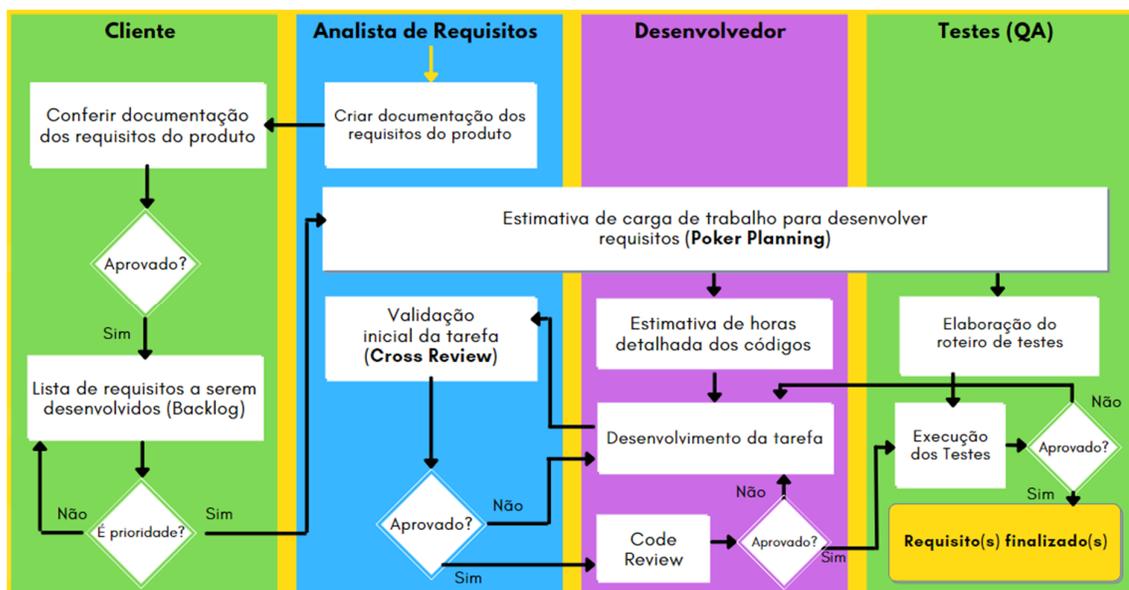
Destacam-se como possíveis gargalos produtivos as definições iniciais de requisitos, design e planejamento do projeto, tendo em vista que múltiplos fatores externos influenciam estas etapas e a provável indefinição logo no início de uma *Sprint* de desenvolvimento é um grande impeditivo para qualquer tarefa do time, onde pode-se em casos de progresso sob certa

liberdade criativa do desenvolvedor, ocorrer retrabalho, dado que será preciso ajustar e alterar características do produto após a confirmação e definição final dos requisitos iniciais. Atualmente, estes pontos levantados são alvos de discussões de melhorias, onde a empresa planeja preparar as conversas de definições iniciais antes do início do desenvolvimento e assim que o projeto possuir um planejamento mais acurado baseado nestas informações, definir um prazo também mais preciso.

Outro ponto de atenção neste modelo utilizado de funções na *Sprint*, são as correções de *bugs* das tarefas dos desenvolvedores ocorrendo apenas na *Sprint* subsequente, o que caracteriza a descrição do anti-padrão 14, de maneira que o que foi produzido na *Sprint* não são de fato entregáveis, pois não foram finalizados. Os gerentes de projetos já têm ciência disto e vêm buscando definir as cargas de trabalho da *Sprint* baseando-se em entregar no mesmo *timebox* as histórias de usuário e suas tarefas de desenvolvimento, testes e correções.

A seguir, está representado na Figura 10 o fluxo simplificado da produção de um requisito, dividido entre os quatros principais papéis na Empresa A que agregam valor direto a este produto. Vale ressaltar que o papel do gerente do projeto não está inserido pois este coordena e supervisiona todas estas etapas representadas, e também o papel dos designers não está representado pois em geral são requisitados sob demanda e em etapas iniciais de levantamento de requisitos.

Figura 10: Fluxo padrão de requisitos em uma Sprint.



Fonte: Autor.

Diferente do fluxo na Figura 9, a ordem e a obrigatoriedade dos processos na Figura 10 é inalterável (salvo raríssimas exceções). Este modelo é fruto de anos de experiências e testes e apresenta-se estável e adequado para atingir quaisquer níveis de ritmo e eficiência necessários nos projetos. Entretanto, na prática são diversos fatores que influenciam não na ausência de algumas etapas, mas no atraso destas.

A comunicação na equipe, apesar de coordenada pelo Gerente de Projetos, deve permanecer clara e eficiente entre os 4 papéis, inclusive na ausência do Gerente. Caso isto não ocorra, torna-se extremamente provável o entendimento parcial de requisitos e *status* das tarefas e como consequência final, atrasos.

Caso ocorram atrasos ou apenas comunicação não eficaz em quaisquer etapas, principalmente entre o Analista de Requisitos e o Cliente, aumentam as chances de um *backlog* do produto desordenado (Anti-padrão 5), falta de conhecimento sobre possíveis problemas e assim a falha na melhoria contínua (Anti-padrão 10), *Sprints* variáveis, muito longas ou deixar os testes para a *Sprint* seguinte (Anti-padrões 12, 13 e 14), times semi-funcionais (Anti-padrão 8) e interrupções futuras do cliente à medida que as expectativas se desvirtuam (Anti-padrão 9). Além disso, o retrabalho gerado ao longo desta *Sprint* tende a consumir horas preciosas de desenvolvimento, as quais então são priorizadas sobre os eventos *Scrum*, em principal a *Sprint Retrospective* e *Sprint Review* ao fim deste *timebox* (Anti-padrão 10), gerando assim *Loops* de *feedbacks* muito longos ou não existentes (Anti-padrão 4).

### 4.2.3 PRINCIPAIS AÇÕES DE SUPORTE AO MÉTODO

Durante o período de observação, destacam-se entre as principais mudanças que auxiliaram no apoio, desempenho e na melhoria geral da implementação do *Scrum*, as seguintes:

**Coleta e controle preciso das horas investidas pelos colaboradores:** O primeiro grande passo dado em direção às melhorias de gestão de projetos foi a criação e conferência diária de planilhas onde importam-se os dados da plataforma de gerenciamento de projetos e marcação de horas. Desta forma, foram reduzidos além de grandes erros de contabilidade ao final do mês, os erros de precisão quanto a estimativas e planejamentos dos projetos. Além disso, com a conseguinte implementação do *software Eazy Bi*, tornou-se ainda mais rápido e fácil acompanhar de perto o trabalho realizado pelos colaboradores e prever possíveis desvios da programação de desenvolvimento e extrair informações relevantes ao cliente, atribuindo assim

aos projetos características de previsibilidade e ajudando a garantir dois dos pilares do *Scrum*, transparência e inspeção.

**Mudança de ferramenta de gestão de projetos:** Por meio da migração do *software* de gerenciamento de tarefas *Teamwork*, para o *Jira Software*, foram adquiridas diversas novas formas de acompanhar o progresso do trabalho realizado. Após cerca de 1 mês de estudo, preparação e período de testes para migração dos times e desligamento do *Teamwork*, a iniciativa foi um sucesso, com o apoio da área de projetos, os times logo se adaptaram à ferramenta e obtiveram grandes resultados, afirmando em consenso que em parte, estes devem-se à facilitação da comunicação e transparência das tarefas entre os integrantes do time. Além disso, o tempo da área de projetos despendido com demandas operacionais de gestão de tarefas reduziu bruscamente, à medida que a plataforma oferece grande liberdade com automações, extração e análise de dados.

**Foco nos requisitos:** Iniciando com a contratação de Analistas de Requisitos, seguindo com a adaptação destes nos Times *Scrum* e a padronização de seus processos, a qualidade das entregas passou a aumentar através do melhor entendimento dos desenvolvedores sobre suas tarefas e dos testes mais eficazes e seguros, através de documentações mais específicas e alinhadas com as expectativas dos clientes. Em seguida, com a definição da métrica de retrabalho dos projetos e o acompanhamento destes indicadores, novas ações de melhorias foram tomadas, como padronizar os documentos de requisitos entre diferentes times e realizar revisões destes documentos com outros membros envolvidos na comunicação com o cliente (*Cross Review* de documentação). Com isto, os níveis de retrabalho, que se caracterizam geralmente por *bugs* ou alterações imprevistas no desenvolvimento, reduziram consideravelmente, de 10% em média de setembro a outubro para 4,5% de novembro a dezembro.

#### 4.2.4 ANÁLISE E DISCUSSÃO

A fim de sistematizar os principais acontecimentos relacionados ao *Scrum* na Empresa A de maneira cronológica, organizou-se uma lista dos eventos ocorridos de junho a dezembro de 2021 que mais impactaram o método de trabalho dos Times *Scrum*, disponível no Apêndice A, a lista dos eventos e datas ocorridos de junho a dezembro de 2021 que mais impactaram o método de trabalho dos Times *Scrum*.

Avaliando estes eventos e seus diferentes contextos na Empresa A, foi possível identificar um padrão evolutivo muito consistente. Ainda que a demanda por melhorias e a

execução destas mudanças tenham surgido de forma espontânea e sem um planejamento muito adequado, o sucesso dos incrementos nos processos de projetos da empresa pôde facilmente ser visualizado na satisfação dos colaboradores e clientes, além das entregas mais consistentes e redução do indicador de retrabalho pela metade.

Para atingir estes resultados satisfatórios foi preciso, porém, inicialmente construir uma estrutura estável de comunicação e registro do trabalho, para que os gestores pudessem desprender gradativamente seus esforços do operacional, conferindo horas investidas pelas equipes e erros de cobrança no financeiro, para o gerencial, coordenando de fato as equipes à nível tático. Em seguida, com a estrutura necessária para garantir a ausência de erros críticos nos relatórios de cobrança dos clientes e acompanhamento de horas do projeto, iniciaram-se os esforços de melhorias na ferramenta de gestão de projetos. Com a implementação planejada do *Jira* e as infinitas possibilidades de personalização na plataforma, as ideias de melhorias de processos internas aos projetos foram se concretizando.

Através de diversos testes em diferentes projetos, notou-se como cada equipe tem níveis de adaptação “ágil” diferentes, baseados nos tipos de produto e nível de maturidade do time. Os projetos mais recentes serviram de ambientes de intensa troca de informações visando melhoria contínua, e a praticabilidade das decisões tomadas foi amplamente maior nestes casos, onde já nos Projetos B e G por exemplo, de longa data e equipes antigas, as decisões eram tomadas com extrema cautela e sob constantes revisões.

Nestes projetos os quais o cliente atua como *Product Owner* e os processos atuais foram produtos de anos de construção consistente, à medida que eram sanados os principais problemas e dores levantados pelos integrantes das equipes, as *Sprints Retrospectives* passaram a alternar de momentos de discussão de poucos pontos específicos, que afetavam claramente o progresso das *Sprints*, para investigação e priorização de diversas ideias com potencial de aumento de eficiência e qualidade das entregas, o que caracteriza o aumento de maturidade da equipe, ao passo que os pontos mais críticos foram de fato tratados e resolvidos.

Já nos projetos novos e com grande potencial de adaptabilidade, os esforços eram direcionados principalmente à adequação da equipe ao ritmo de demanda do cliente, o qual torna-se desproporcionalmente intenso em seu início, durante a fase de levantamento dos requisitos gerais do produto. Ainda que muito se tenha conquistado em melhorias dos requisitos levantados ao longo do projeto e que exista um padrão desejado entre os projetos da Empresa A, nestes casos em fase inicial de desenvolvimento, a adequação brusca e repentina acaba por causar disrupções diretas e indiretas do cliente (anti-padrão 9) no trabalho da equipe, de modo

que o gerente do projeto, analista de requisitos e designer acabam sobrecarregados, enquanto por outra via, os desenvolvedores e time de QA tornam-se ociosos, à espera de definições do que produzir.

Desta forma, é preciso direcionar o foco das melhorias de processos na área de requisitos do produto, tendo em vista que foi uma das aquisições mais recentes da empresa. Pensando em termos de anti-padrão, como os vistos no capítulo 2, pode-se definir como um anti-padrão ou desvio não necessariamente do *Scrum*, mas das etapas de planejamento do projeto e que afetam, portanto, diretamente na performance dos times *Scrum*, a falta de definições de requisitos durante os acordos comerciais em pré-desenvolvimento e a ausência de um período devidamente formalizado de ajustes e preparação do time para a *Sprint* 1, o que foi abordado como uma das estratégias dos tópicos 2.4 e 2.5. Ainda que o *Scrum* não preveja esta etapa e esta ainda pareça contraditória sob a ótica “ágil”, pois adiciona-se processos e horas em planejamento, arma-se uma armadilha ao posterga-la para o início da primeira *Sprint*, mesmo que conforme comentado na seção 2.1, o conhecimento na curva de aprendizado deve ser mínimo neste momento, porém não tão pequeno a ponto de atrasar as definições e repasse de tarefas bem detalhadas, e gerar assim um retrabalho evitável.

Já ao analisar a adequação da empresa sob o principal aviso encontrado nas páginas iniciais do Guia *Scrum* (seção 2.2), que em resumo tratam a prática de remover elementos do Guia tidos como padrão do *Scrum* como algo que descaracteriza o próprio *framework*, e já o contrário, quando acrescentam-se processos complementares e que agregam aos tidos como padrão, como práticas bem-vindas, pode-se atestar que a Empresa A possui a tendência de remover alguns eventos do *Scrum* de forma esporádica e acrescentar diversos outros de maneira única em cada projeto.

Como o principal desvio do método, destaca-se o hábito de priorizar horas investidas em tarefas que agregam valor direto ao produto ao invés dos eventos *Scrum*, de forma que as *Sprint Reviews* não são realizadas e quando a entrega parece comprometida pelo tempo restante até o prazo de entrega, as *Sprint Retrospectives* e *Plannings* são canceladas. Além disso, quanto a importância dada para cada um dos eventos *Scrum*, a empresa segue o mesmo padrão encontrado no *15th State of Agile Report (2021)*, seguindo respectivamente em ordem de priorização, *Daily Standups*, *Sprint Retrospective*, *Sprint Planning* e *Sprint Review*. Esta situação tende assim a resultar nos anti-padrões 4, 5, 6 e 10, onde enalta-se como principal causa destas situações a falta de alinhamento de expectativas com o cliente ou *Product Owner*,

à causa de que estes não enxergam o progresso da equipe de forma transparente e não estão devidamente educados ao modelo de trabalho do Time *Scrum* e à própria metodologia.

Além disso, a desconformidade dos cargos dos integrantes dos projetos com os papéis *Scrum* tende a trazer dificuldades de limites de funções e atribuições. Apesar dos desenvolvedores, sob a definição do Guia, estarem muito bem alinhados com o que é visto na prática na empresa, o papel do *Product Owner* é dividido entre múltiplas pessoas e de diferentes maneiras e proporções em cada projeto. Em geral, as atribuições do *P.O.* dividem-se entre o cliente, gerente do projeto e analista de requisitos. As ideias de *features* e o que deve ser construído partem do cliente, o qual não está inserido no time *Scrum* (anti-padrão 2 e 9), portanto para que o *backlog* esteja de acordo com a capacidade do time nas *Sprints* e até com o planejamento de desenvolvimento do produto, o gerente de projetos tenta planejar, coordenar e gerenciar as tarefas do time através de alinhamentos constantes e de certa forma exaustivos com o cliente. Por fim, o analista de requisitos toma a frente nas tarefas mais operacionais, relacionadas à documentação e revisão dos requisitos de cada *feature* a ser estimada e desenvolvida pelo time.

Finalmente, através do Projeto E no mês de dezembro, foi possível verificar o quão adaptável e maduro o time *Scrum* deste projeto (formado por membros de outros projetos em paralelo) provaria ser e através da experiência adquirida com sucessos e dificuldades passadas, o time venceu grandes desafios relacionados a todas as problemáticas descritas no tópico 2.1. Porém vale ressaltar a difícil capacidade de discernimento entre o que de fato é uma adaptação e o que é apenas incapacidade do time em seguir os padrões de sucesso conhecidos.

Ainda que o time tenha deixado de seguir os processos exigidos pelo Guia *Scrum*, uma significativa característica válida de se realçar é a escolha consciente de suprimir certos eventos e processos em prol da agilidade. A escolha consciente aqui, é definida pela ciência do desvio de teor teórico do *framework* adaptado e dos impactos destas escolhas. Desta forma, os cortes realizados em horas investidas em certas áreas consideradas importantes para a melhoria contínua, foram substituídos por valor agregado diretamente ao produto, sendo assim, justificadamente, um desvio do método *Scrum*. E ainda, pelo fato de se entender a importância e perigo destes desvios, o time tende a recuperar seus hábitos ágeis à medida que o projeto se distancia das fases iniciais de levantamento de requisitos e estabiliza seu ritmo de desenvolvimento. Assim, apesar da contrariedade de se poupar esforços em tempo que agrega na qualidade geral das entregas a fim de melhorar o próprio produto, é prevista na equipe

madura, o retorno acelerado aos padrões de qualidade normais, onde a partir daí serão buscados novos e não previstos incrementos aos processos.

### 4.3 CAUSAS DIRETAS DE ANTI-PADRÕES E DESVIOS NO SCRUM

Através de todas as informações coletadas nas entrevistas e no período de acompanhamento dos projetos de desenvolvimento de *software*, foi possível elencar as principais causas diretas que tendem a levar os Times *Scrum* a efetuarem os anti-padrões descritos na seção 2.4.

A seguir, encontram-se no Quadro 7 todas as causas identificadas nas entrevistas e na pesquisa-ação e no Quadro 8 suas relações com os anti-padrões, onde os termos anti-padrão e desvio foram substituídos por AP e D respectivamente, seguidos do número identificador do anti-padrão na seção 2.4. Vale ressaltar que nas colunas da pesquisa-ação dos dois quadros não foram duplicadas causas já levantadas nas entrevistas, também estão indicados na Quadro 7 o grau de impacto ao projeto destas situações e ao fim do Quadro 8 está o desvio apresentado como um anti-padrão extra na seção 4.2.4.

Quadro 7: Lista de causas diretas ao anti-padrões e desvios do Scrum.

Causas identificadas nas entrevistas	Impacto	Causas identificadas na pesquisa-ação	Impacto
Falta de especialização.	Grave	Agregar maior valor para o cliente.	Positivo
Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Grave	Otimização de horas (Projeto B e F).	Positivo
Priorizar horas de desenvolvimento.	Médio	Comunicação não efetiva.	Grave
Dúvidas sobre funções do papel.	Médio	D1: Falta definição de requisitos no pré-desenvolvimento	Grave
Sempre adaptar o método à realidade.	Médio	Falta de planejamento do produto.	Grave
Produto instável ou não estruturado.	Médio	Requisitos não definidos/entendidos.	Grave
AP2 e cliente não educado ao processo.	Médio	AP11 e Estimativas ruins.	Médio
Pouco tempo de experiência.	Leve	Falta de alinhamento de expectativas.	Médio
Ferramenta(s) inadequada(s).	Leve	Falta de padronização nos processos.	Médio
Acredita-se ser a melhor opção.	Leve	Falta mentalidade ágil no time.	Médio
		Sobrecarga do Gerente de Projetos.	Médio
		Cliente no papel de <i>Product Owner</i> (AP2).	Leve
		Falta de <i>P.O.</i> na Empresa A.	Leve
		Concentração de conhecimento.	Leve
		Gerente de Projetos no papel de <i>P.O.</i>	Leve
		Estimar apenas tarefas dos desenvolvedores.	Leve

Fonte: Autor

Quadro 8: Anti-padrões e desvios do Scrum e suas causas diretas.

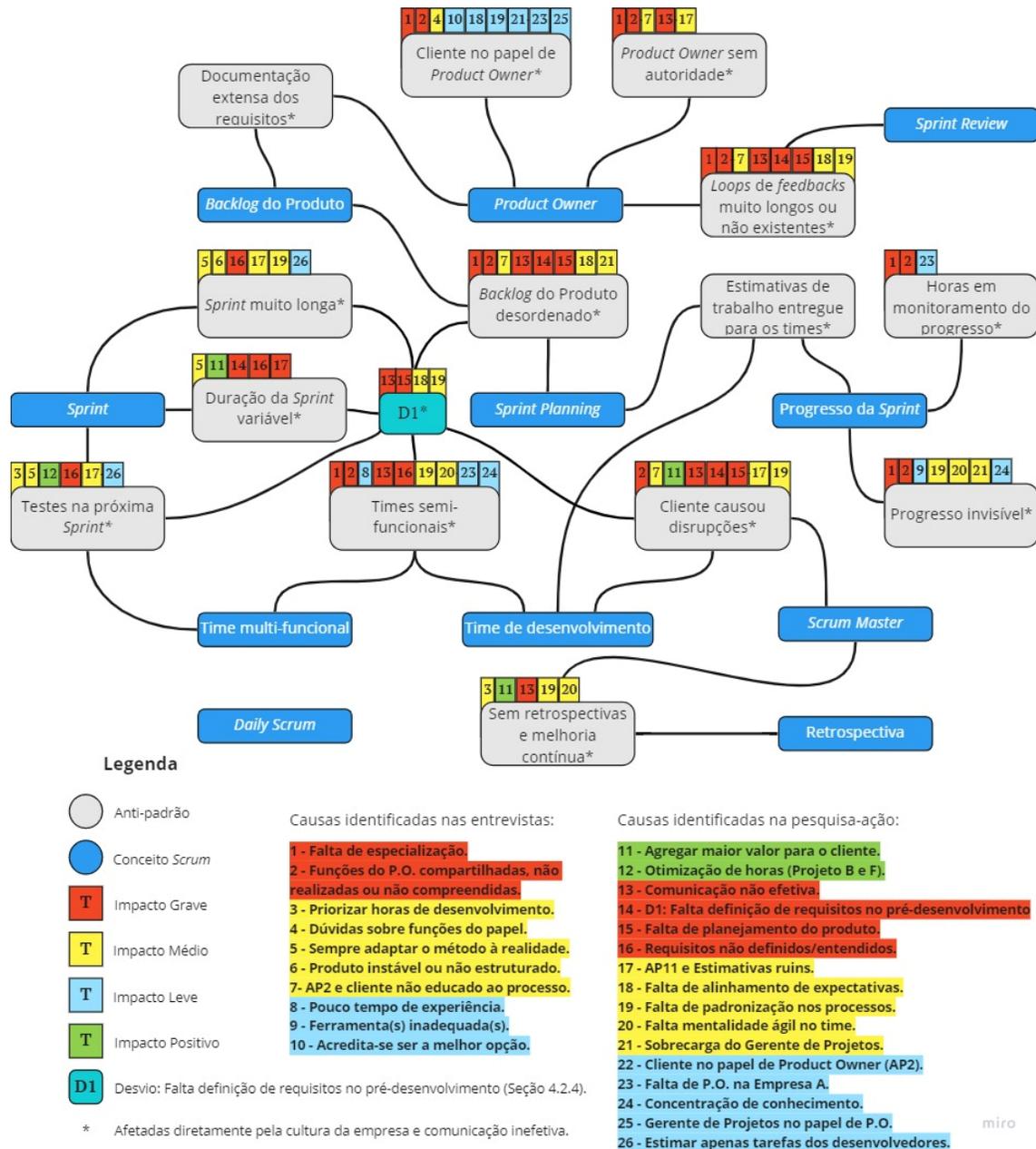
Anti-padrões (AP) e desvios (D) do Scrum	Causas diretas identificadas	
	Entrevistas	Pesquisa-ação
<b>AP1: Documentação extensa de requisitos.</b>		
<b>AP2: Cliente no papel de <i>Product Owner</i></b>	Acredita-se ser a melhor opção.	Falta de <i>P.O.</i> na Empresa A.
	Dúvidas sobre funções do papel.	Gerente de Projetos no papel de <i>P.O.</i>
	Falta de especialização.	Falta de alinhamento de expectativas.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta de padronização nos processos.
		Sobrecarga do Gerente de Projetos.
<b>AP3: <i>Product Owner</i> sem autoridade</b>	Falta de especialização.	AP2.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Comunicação não efetiva.
		AP11 e Estimativas ruins.
<b>AP4: Loops de feedbacks muito longos ou não existentes</b>	AP2 e cliente não educado ao processo.	Falta mentalidade ágil no time.
	Priorizar horas de desenvolvimento.	Falta de padronização nos processos.
<b>AP5: <i>Backlog</i> do Produto desordenado</b>	AP2 e cliente não educado ao processo.	Falta de alinhamento de expectativas.
	Falta de especialização.	Comunicação não efetiva.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta de planejamento do produto.
		Sobrecarga do Gerente de Projetos.
	D1.	
<b>AP6: Estimativas de trabalho entregue para os times</b>		
<b>AP7: Horas em monitoramento do progresso</b>	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta de <i>P.O.</i> na Empresa A.
	Falta de especialização.	
<b>AP8: Times semi-funcionais</b>	Pouco tempo de experiência.	Concentração de conhecimento.
	Falta de especialização.	Requisitos não definidos/entendidos.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta de <i>P.O.</i> na Empresa A.
		Comunicação não efetiva.
		Falta mentalidade ágil no time.
		Falta de padronização nos processos.
<b>AP9: Cliente causou interrupções</b>	AP2 e cliente não educado ao processo.	Concentração de conhecimento.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta de planejamento do produto.
		Comunicação não efetiva.
		Falta de padronização nos processos.
		D1.
		AP11 e Estimativas ruins.
		Agregar maior valor para o cliente.

<b>AP10: Sem retrospectivas e melhoria contínua</b>	Priorizar horas de desenvolvimento.	Falta mentalidade ágil no time.
		Falta de padronização nos processos.
		Comunicação não efetiva.
		Agregar maior valor para o cliente.
<b>AP11: Progresso invisível</b>	Falta de especialização.	Concentração de conhecimento.
	Funções do <i>P.O.</i> compartilhadas, não realizadas ou não compreendidas.	Falta mentalidade ágil no time.
	Ferramenta(s) inadequada(s).	Falta de padronização nos processos.
		Sobrecarga do Gerente de Projetos.
<b>AP12: Duração da <i>Sprint</i> variável</b>	Sempre adaptar o método à realidade.	Requisitos não definidos/entendidos.
		D1.
		AP11 e Estimativas ruins.
		Agregar maior valor para o cliente.
<b>AP13: <i>Sprint</i> muito longa</b>	Sempre adaptar o método à realidade.	Requisitos não definidos/entendidos.
	Produto instável ou não estruturado.	Falta de padronização nos processos.
		AP11 e Estimativas ruins.
		Estimar apenas tarefas dos desenvolvedores.
<b>AP14: Testes na próxima <i>Sprint</i></b>	Sempre adaptar o método à realidade.	Requisitos não definidos/entendidos.
	Priorizar horas de desenvolvimento.	AP11 e Estimativas ruins.
		Otimização de horas (Projeto B e F)
		Estimar apenas tarefas dos desenvolvedores.
<b>D1: Falta definição de requisitos no pré-desenvolvimento (Seção 4.2.4).</b>		Falta de planejamento do produto.
		Falta de alinhamento de expectativas.
		Falta de padronização nos processos.
		Comunicação não efetiva.

Fonte: Autor

A partir das causas diretas elencadas e o grau de impacto nos projetos, é possível visualizar através da Figura 11, como as causas raízes estão intrinsicamente interligadas com os anti-padrões e são de fato causadores de diversos problemas na aplicação da metodologia ágil. Cabe destacar que apenas os anti-padrões 1 e 6 não foram identificados no estudo e também que duas causas e fatores influenciadores foram identificados como impactantes em todos os anti-padrões, “Cultura da empresa” e “Comunicação Inefetiva”. Além disso, foi incluído no esquema da Figura 11 o desvio “Falta definição de requisitos no pré-desenvolvimento” e suas causas e relação com 6 outros anti-padrões.

Figura 11: Esquema de relacionamento de causas, anti-padrões e conceitos Scrum.



Fonte: Autor

Em conclusão, infere-se que das 26 causas identificadas, 24 destas impactam negativamente os projetos *Scrum*, sendo 6 delas com impacto grave, 10 médio, e 8 leve. Por outra via, as causas 11 e 12 mostraram-se como justificativas plausíveis de anti-padrões, à medida que são escolhas conscientes e que de fato trazem benefícios diretos ao projeto e à equipe.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve o objetivo de demonstrar os principais fatores críticos e impeditivos à aplicação do método ágil *Scrum* de gerenciamento de projetos em empresas de desenvolvimento de software, identificando através da literatura, entrevistas e pesquisa-ação em empresa e projetos do setor, qual a percepção dos gestores e equipes dos projetos em projetos *Scrum* e quais são de fato as causas diretas de sucesso ou fracasso ao adotar a metodologia.

Após o levantamento dos anti-padrões e embasamento teórico para contribuir nas discussões e análises das etapas seguintes, foram realizadas as entrevistas com 8 colaboradores de diferentes cargos e níveis de experiência da indústria de tecnologia. Assim, foi possível identificar padrões relevantes entre os respondentes e identificar 10 causas diretas dos anti-padrões do *Scrum*, também identificados no referencial teórico.

Em seguida, por meio da pesquisa-ação na Empresa A, ao analisar inicialmente o modelo de gestão da organização, já foi possível destacar diferentes características entre os projetos e seus contextos, que causavam impactos diretos ao *Scrum*. Além disso, vale ressaltar as ações de suporte efetuadas pela empresa, que ajudaram a estabelecer definitivamente um ambiente de melhoria contínua da metodologia ágil. Por fim, por meio da cronologia de eventos (Apêndice A), permitiu-se levantar todas as informações complementares necessárias para analisar e discutir o contexto geral da empresa e seus hábitos quanto à adoção do *Scrum*.

Desta forma, complementando as causas de anti-padrões identificadas pelas entrevistas, a pesquisa-ação proveu como base para identificar mais 26 diferentes causas. Também foi elencado o nível de impacto, entre leve, médio, grave e positivo, das 26 causas totais nos projetos da Empresa A, além de representar as relações com cada um dos 14 anti-padrões e 11 conceitos *Scrum*.

Em adição aos anti-padrões e causas identificadas ao longo do trabalho, vale ressaltar a definição de um novo anti-padrão e também causa direta de outros, atribuída à falta de definição de requisitos no pré-desenvolvimento, a qual somente tende a ocasionar problemas nas entregas, ao invés da aparente agilidade adquirida ao dividir este planejamento ao longo das *Sprints*.

Tendo em vista todos os aspectos dos projetos, dos fatores humanos e da adoção do *Scrum* analisados neste trabalho e a natureza iterativa da pesquisa-ação, pode-se concluir que a

Empresa A se encontra agora em novos patamares de maturidade ágil e consciência dos próximos passos necessários em escalabilidade das equipes e implementação do *Scrum*.

É preciso reforçar, porém, que este estudo foi aplicado em um ambiente específico, de uma empresa pequena e seus projetos, clientes e empresas parceiras. Desta forma, é necessária cautela ao atribuir as características destes resultados à outras organizações e modelos de negócio. Ainda que seja válida a comparação e inferência dos produtos deste trabalho a empresas de desenvolvimento de *software*, os resultados aqui apresentados levaram em consideração projetos com mão-de-obra terceirizada e também projetos com produtos nativos, que tendem a ter pontos críticos diferentes, à medida que o prazo e tipo de cliente atuam de formas diferentes nas decisões do projeto. Assim, o leitor que procurar concluir e comparar os resultados finais fora deste trabalho, deve atentar-se a estas características do seu próprio ambiente de análise.

## 5.1 TRABALHOS FUTUROS

De acordo com a recomendação ao final da seção anterior, torna-se necessária a avaliação inicial do novo ambiente de estudo em relação à semelhança das características da Empresa A. Caso seja verdadeira a adequação, este trabalho pode servir como referência na identificação de anti-padrões do *Scrum* e situações de atenção em projetos de desenvolvimento de *software*, além de prover opiniões e discussões de diversos tópicos relacionados ao *Scrum*.

Por fim, conforme constatado neste trabalho o consenso dos entrevistados em atribuir a culpa da inefetividade apenas aos fatores humanos, um interessante produto de pesquisa seria o aprofundamento destes fatores, identificando e classificando padrões de equipes, assim como soluções não só para anti-padrões, mas para conflitos caracterizados como causas indiretas de retrabalho e desvios do método.

## REFERÊNCIAS

- Digital.ai. **15th State of Agile Report**. Disponível em: <<https://digital.ai/resource-center/analyst-reports/state-of-agile-report>>. Acesso em: 18 mar. 2022.
- AL-SAQQA, S.; SAWALHA, S.; ABDELNABI, H. Agile software development: Methodologies and trends. **International Journal of Interactive Mobile Technologies**, v. 14, n. 11, p. 246–270, 2020.
- BECK, K. et al. **Manifesto for Agile Software Development Twelve Principles of Agile Software**. Disponível em: <<http://www.agilemanifesto.org>>. Acesso em: 18 mar. 2022.
- COHEN, D.; LINDVALL, M.; COSTA, P. **DACS State-of-the-Art/Practice Report Agile Software Development**.
- COSTA, E. P.; POLITANO, P. R.; PEREIRA, N. A. Exemplo de aplicação do método de Pesquisa-ação para a solução de um problema de sistema de informação em uma empresa produtora de cana-de-açúcar. **Gestao e Producao**, v. 21, n. 4, p. 895–905, 1 out. 2014.
- DOMINGUES, F. **Metodologia Ágil: A arte de fazer o dobro em metade do tempo**. Disponível em: <<https://www.inovalab.net.br/post/metodologia-ágil-tudo-que-você-precisa-saber-antes-de-começar-a-usar>>. Acesso em: 17 out. 2021.
- ELORANTA, V. P.; KOSKIMIES, K.; MIKKONEN, T. Exploring ScrumBut - An empirical study of Scrum anti-patterns. **Information and Software Technology**, v. 74, p. 194–203, 1 jun. 2016.
- ENGEL, G. I. Pesquisa-ação. In: **Educar**. v. n. 16p. 181–191.
- ESPINHA, R. **Sprint: o que é e como fazer essa etapa do SCRUM**. Disponível em: <<https://artia.com/blog/sprint/>>. Acesso em: 23 out. 2021.

HASSANI-ALAOUI, S.; CAMERON, A.-F.; GIANNELIA, T. **“We Use Scrum, but ...”:** **Agile Modifications and Project Success.** Proceedings of the 53rd Hawaii International Conference on System Sciences. **Anais...**2020.

HEIKKILA, V. T.; PAASIVAARA, M.; LASSENIUS, C. **ScrumBut, but does it matter? A mixed-method study of the planning process of a multi-team scrum organization.** International Symposium on Empirical Software Engineering and Measurement. **Anais...**2013.

HRON, M.; OBWEGESER, N. **Scrum in practice: an overview of Scrum adaptations.** Proceedings of the 51stHawaii International Conference on System Sciences. **Anais...**2018. Disponível em: <<http://hdl.handle.net/10125/50568>>

KUSSUNGA, F.; RIBEIRO, P. **Proposal of a Visual Environment to Support Scrum.** Procedia Computer Science. **Anais...**Elsevier B.V., 2019.

LEI, H. et al. A statistical analysis of the effects of Scrum and Kanban on software development projects. **Robotics and Computer-Integrated Manufacturing**, v. 43, p. 59–67, 1 fev. 2017.

MORANDINI, M. et al. **Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams****Computer Science Review**Elsevier Ireland Ltd, , 1 fev. 2021.

SALAS, E.; SIMS, D. E.; SHAWN BURKE, C. **Is there A “big five” in teamwork?****Small Group Research**, out. 2005.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game.** [s.l: s.n.].

TAM, C. et al. The factors influencing the success of on-going agile software development projects. **International Journal of Project Management**, v. 38, n. 3, p. 165–176, 1 abr. 2020.

## APÊNDICE A – Cronologia da evolução do *Scrum* na Empresa A

Neste tópico foram elencados os principais eventos relacionados direta e indiretamente à execução do *Scrum* nos projetos da Empresa A.

### JULHO 2021

<b>01/07</b> - Efetivação do novo gestor de projetos. Momento de turbulência com clientes antigos e reestruturação de processos.
<b>07/07</b> - Projeto H é iniciado.
<b>09/07</b> – Gestor planeja entrega final do Projeto F, porém novas decisões não agradam cliente e a relação já abalada apenas piora a partir daqui.
<b>15/07</b> - Criação de planilha de controle de horas dos funcionários e dos projetos para compreender as dores destes projetos através de dados.
<b>19/07</b> – Criação de bonificações para funcionários que marcam as horas nos projetos corretamente.
<b>20/07</b> – Criação de atualizações periódicas sobre horas investidas nos projetos para os clientes.
<b>22/07</b> - Incentivo da empresa para cursos em gestão de projetos.
<b>31/07</b> - Novo modelo de coleta de dados para revisão e cobrança dos clientes.

### AGOSTO 2021

<b>02/08</b> – Ferramenta nova para reunião de estimativas de tarefas ( <i>Planning Poker</i> ).
<b>11/08</b> – Ferramenta nova de <i>Business Intelligence</i> para extração e análise de dados. Reorganização da gestão de tarefas internas da empresa.
<b>20/08</b> – Mais de 1 mês após as conversas de levantamento de requisitos do Projeto H, as tarefas são estimadas para desenvolvimento na <i>Sprint 1</i> .
<b>18/08</b> - Decisão de mudar ferramenta de gestão para o <i>Jira</i> fica mais concreta, justificativas e esforços necessários são elencados e preparação do ambiente iniciada.
<b>25/08</b> – Início de reuniões semanais da área de projetos.
<b>29/08</b> – Projeto B, sob gestão do cliente, apresenta dificuldades com <i>Jira</i> . CTO centraliza a gestão e tem dificuldades com a navegação e visualização na ferramenta. Não hierarquizam as tarefas e têm <i>Sprint Plannings</i> muito longas. Começaram a utilizar nova ferramenta para as <i>Sprints Retrospectives</i> , incentivando diariamente seu uso nas <i>daillies</i> .
<b>31/08</b> – Projeto B procura otimizar o uso de estimativas por <i>Story Points</i> , começam a utilizar também a ferramenta <i>Planning Poker</i> e tamanhos de camiseta genéricos ao invés de horas estimadas (anti-padrão 7) nas tarefas. Além disso é necessário chegar num consenso para determinar a estimativa de uma tarefa.

### SETEMBRO 2021

<b>02/09</b> - São criadas bonificações e iniciativas de redução de retrabalho nos times de desenvolvimento.
<b>03/09</b> - Retrospectivas devem agora sempre terminar com planos de ação. Projeto F conseguiu passar de <i>Sprints</i> mal planejadas de 2 a 3 meses (anti-padrão 12 e 13), para <i>Sprints</i> padrão de 2

semanas, devido à inclusão de uma analista de requisitos e assim melhores definições dos requisitos. Empresa A também começa a medir todos os projetos por tamanhos de camiseta ao invés de horas.
<b>08/09</b> - Dificuldades no Projeto B em manter alimentação diárias de sugestões de melhorias.
<b>09/09</b> – Início da migração de todos os projetos para o <i>Jira</i> . Grande esforço necessário devido à personalização para cada projeto, principalmente em seus workflows (etapas das tarefas).
<b>16/09</b> - Transparência do indicador de retrabalho, mostrando o acompanhamento toda sexta-feira.
<b>21/09</b> – Projeto B começa a se tornar um pouco mais ágil e estão procurando por <i>P.O.</i> Ainda não conseguem terminar as tarefas de desenvolvimento propostas para as <i>Sprints</i> (Anti-padrão 14).
<b>22/09</b> – Projeto G oficializa a utilização de 3 <i>Sprints</i> simultâneas, a fim realizar com calma e clareza uma <i>Sprint</i> de requisitos, uma de desenvolvimento e outra de testes (anti-padrão 14). Torna-se unânime a necessidade de melhor documentação dos testes, tanto do QA quanto dos desenvolvedores ao terminarem uma tarefa.
<b>24/09</b> – Primeiro projeto teste gerido no <i>Jira</i> (Projeto G): Time aprova e considera muito melhor que a plataforma antiga ( <i>Teamwork</i> ), Analista de Requisitos passa a utilizar um <i>board</i> de requisitos e podendo assim separar o fluxo de documentação antes do desenvolvimento.
<b>28/09</b> – <i>Sprint Retrospective</i> no Projeto B ainda não analisa questões de ritmo de entrega da <i>Sprint</i> e acabam fazendo isso na <i>Sprint Planning</i> , tornando-a onerosa. Líder da equipe, o <i>CTO</i> entende cada vez mais seu papel, que deve ser ágil, e o impacto na motivação da equipe foi visível durante a retrospectiva.
<b>29/09</b> – <i>Sprint Planning</i> no Projeto B, após a última retrospectiva: Time cheio de ideias, foram adicionadas as etapas de <i>Cross Review</i> e <i>Code Review</i> no ciclo de desenvolvimento, os quais visam reduzir erros e retrabalho.

## OUTUBRO 2021

<b>04/10</b> - Migração geral nos projetos da Empresa A para o <i>Jira</i> , muitas ideias são levantadas, como <i>boards</i> e <i>workflows</i> personalizados, melhores descrições de tarefas, e métodos de extração e elaboração de relatórios gerenciais. Além disso, através de análise de dados, notou-se a carga horária excessiva de desenvolvimento no Projeto G, que já ocorria há 1 ano e o time e cliente não tinham noção.
<b>05/10</b> - Time de <i>business</i> do cliente no Projeto B está em maior contato com o time <i>Scrum</i> , a fim de melhorar a comunicação e entendimento das entregas (sem ocorrer disrupções, anti-padrão 9).
<b>13/10</b> – Testador sai da Empresa A e do Projeto B, impactando capacidade das entregas.
<b>19/10</b> – Após um ano de grande rotatividade na equipe, o cliente do Projeto B está terminando os processos de contratação do <i>P.O.</i> , com a qual o time deposita grandes esperanças para implementação de melhorias. O grande gargalo produtivo está no papel do <i>CTO</i> , principalmente no <i>Cross Review</i> , onde com a ausência dele as tarefas ficam paradas e sem andamento. Foi decidido que novas mudanças e melhorias só devem ser discutidas e implementadas com a chegada do <i>P.O.</i>
<b>22/10</b> - <i>Sprint</i> sem <i>CTO</i> na equipe foi ineficaz, com muitos bloqueios e indefinições, além das tarefas ficaram represadas no <i>Cross Review</i> (anti-padrão 8).
<b>25/10</b> - Início do Projeto I e conversas comerciais para iniciar Projeto E, as quais tinham o intuito de refinar o <i>backlog</i> inicial de um <i>MVP</i> (Versão mínima do produto) e se prolongaram o suficiente para prejudicar o prazo de entrega do mesmo. Assim iniciaram-se discussões sobre etapas pré-desenvolvimento nos projetos. Além disso, com a chegada do <i>P.O.</i> no Projeto B, diversas ideias foram levantadas, como utilizar termos de aceite ou definição de pronto nas tarefas e descrevê-las de forma que possam ser compreendidas por pessoas não técnicas e de fora do time.
<b>28/10</b> – Ações planejadas pelo <i>P.O.</i> no Projeto B: Trazer QA para início da definição de requisitos, mudar visualização de tarefas do nível de <i>features</i> para histórias de usuário, agregando valor nas entregas dos desenvolvedores e facilitando o entendimento deles, reduzindo assim a demanda e necessidade de mudanças no <i>Cross Review</i> .

## NOVEMBRO 2021

<b>01/11</b> – Projeto F é suspenso e encerrado sob contexto insatisfatório para Empresa A e cliente.
<b>05/11</b> – Nota-se a necessidade de definir todas as tarefas em entregas que caibam na <i>Sprint</i> , visando restar no <i>board</i> de acompanhamento do <i>backlog</i> da <i>Sprint</i> apenas tarefas em pendentes ou finalizadas.
<b>08/11</b> - Foi iniciado a implementação das seguintes ideias no Projeto B: Tarefas de desenvolvimento que não levem mais que um dia para ser feito <i>Cross Review</i> , Responsáveis do time pela entrega de cada história do usuário e <i>dashboard</i> de acompanhamento nas reuniões <i>Daily</i> das <i>Sprints</i> .
<b>10/11</b> – Define-se que time de QA deve testar histórias do usuário como um todo, ou seja, só iniciar o teste assim que todas as tarefas de desenvolvimento destas histórias estiverem prontas.
<b>12/11</b> – Implementadas reuniões de <i>checkpoint</i> das histórias com os seus respectivos responsáveis.
<b>17/11</b> – Definido que apenas o time de QA deve estimar tarefas de QA e somente quem for fazer os requisitos deve estimar a documentação, ainda que isso aconteça em conjunto na mesma reunião.
<b>21/11</b> – Projeto B: Em contraste com meses atrás, ocorreu uma <i>Retrospective</i> com muitas opiniões, mas com falta de priorização e controle do tempo na reunião, não resultando assim em bons planos de ação.
<b>29/11</b> – Início do Projeto E: Autor deste trabalho atua como Gerente do Projeto, e novo colaborador na Empresa A entra no time de desenvolvimento do projeto.

## DEZEMBRO 2021

<b>03/12</b> – Durante <i>Sprint Retrospective</i> no Projeto B, o time escolheu conscientemente realizar o anti-padrão 14, realizando testes na <i>Sprint</i> seguinte, baseando-se no Projeto G.
<b>04/12</b> – Projeto E: Longas reuniões de alinhamento e definições de requisitos durante a <i>Sprint 1</i> , pois não foram definidas com clareza nas reuniões comerciais com o cliente. Atrasando e sobrecarregando as tarefas de desenvolvimento.
<b>07/12</b> – Colaborador recém chegado à Empresa A e Projeto E apresenta grandes dificuldades de adaptação e causou fortes disrupções no processo <i>Scrum</i> , removendo diversas etapas das tarefas.
<b>09/12</b> - Colaborador foi dispensado da empresa e CEO tornou-se desenvolvedor do projeto.
<b>15/12</b> - Diversas adaptações aos processos foram feitas no Projeto E para tornar o desenvolvimento mais ágil, como requisitos menos específicos, exceção do CEO nas <i>daillies</i> , participando assim de forma assíncrona, exclusão do <i>Code Review</i> e testes menos rigorosos.
<b>22/12</b> - Notou-se que as horas de gestão do Projeto E foram muito superiores aos outros, devido aos requisitos menos detalhados e a diferença de idiomas com o cliente. O gerente do projeto precisou realizar muito mais horas de estudo e alinhamento dos requisitos, realizando a ponte comunicativa entre empresa cliente e time de desenvolvimento, encarecendo o valor cobrado do cliente.
<b>31/12</b> - Time adaptou-se ao ritmo diferente deste do Projeto E, conseguindo recuperar o tempo perdido nas duas primeiras semanas com o colaborador dispensado, retornando o desenvolvimento dentro do prazo das entregas.