

# DESAFIOS DA APLICAÇÃO DO ÁGIL ESCALADO EM PROJETOS DE SOFTWARE: ESTUDO DE CASO EM UMA ORGANIZAÇÃO FINANCEIRA

## CHALLENGES OF AGILE SCALED APPLICATION IN SOFTWARE PROJECTS



**Leonardo Augusto Feitosa**

Especialista em Engenharia de Projetos em Sistemas de Informação  
Faculdade Carlos Drummond de Andrade – FCDA.  
São Paulo, SP – Brasil.  
[leoredhat@gmail.com](mailto:leoredhat@gmail.com)



**Wagner Solivan Ferreira**

Mestre em Administração  
Universidade Nove de Julho – UNINOVE.  
São Paulo, SP – Brasil.  
[wgsolivan@gmail.com](mailto:wgsolivan@gmail.com)

### Resumo

Este artigo tem por objetivo apresentar os desafios da aplicação do ágil escalado em projetos de *software* caracterizados por alta complexidade e longa duração, por meio da realização de um estudo de caso em uma organização financeira do Brasil. A pesquisa possui abordagem qualitativa e natureza descritiva, com a unidade de pesquisa caracterizada por um projeto de desenvolvimento de *software* de alta complexidade em uma empresa do setor financeiro. Os resultados desta pesquisa indicaram que apesar da organização já desenvolver seus projetos de *software* em conjunto da abordagem de uma metodologia ágil escalada, ainda enfrenta dificuldades relacionadas ao desalinhamento entre a metodologia, a complexidade e a duração do projeto. Logo, a presente pesquisa contribui para a prática, apresentando possíveis ações mitigadoras para a implementação de uma metodologia ágil em escala, em detrimento das adversidades das organizações. Desafios secundários foram evidenciados em consequência da inclusão de trabalho não planejado, oriundo de agências reguladoras ao escopo do projeto, desta forma caracterizando-se como a maior dificuldade encontrada pela organização durante a implementação da metodologia ágil escalada ao seu projeto.

**Palavra-chave:** Metodologia Ágil. Ágil Escalado. Projeto de Software. Gestão de projetos. Software.

### Abstract

This article aims to present the challenges of applying agile scaled in software projects characterized by high complexity and long duration, by conducting a case study in a financial organization in Brazil. The research has a qualitative approach and a descriptive nature, with the research unit characterized by a highly complex software development project in a company in the financial sector. The results of this research indicated that despite the organization already developing its software projects together with the approach of a scaled agile methodology, it still faces difficulties related to the misalignment between the methodology, the complexity and the duration of the project. Therefore, this research contributes to the practice, presenting possible mitigating actions for the implementation of an agile scale methodology to the detriment of the adversities of the organizations. Secondary challenges were highlighted as a result of the inclusion of unplanned work from regulatory agencies within the scope of the project, thus being characterized as the greatest difficulty encountered by the organization during the implementation of the agile methodology scaled to its project.

**Keywords:** Agile Methodology. Scaled Agile. Software Project. Project management. Software.

*Cite como - American Psychological Association (APA)*

Feitosa, L. A., & Ferreira, W. S. (2021, Ed. Esp. jan./abr.). Desafios da aplicação do ágil escalado em projetos de software: estudo de caso em uma organização financeira. *Revista de Gestão e Projetos (GeP)*, 12(1), 195-221. <https://doi.org/10.5585/gep.v12i1.17825>.

## 1 Introdução

Projetos complexos possuem características comuns, como o alto volume de recursos, elevada quantidade de código, orçamento e colaboradores envolvidos, o que exigem das organizações maiores esforços para o seu gerenciamento (Abrar et al., 2019; Pitkänen, 2015), gerando uma dificuldade natural para a obtenção de níveis satisfatórios de qualidade e desempenho. Scheerer, Hildenbrand e Kude (2014) reiteram que o gerenciamento de tarefas e múltiplas equipes se torna uma adversidade para as organizações, que necessitam coordenar o trabalho destas equipes, que em muitos casos estão geograficamente distribuídas (Dikert, Paasivaara, & Lassenius, 2016).

Os métodos ágeis transformaram a maneira como *softwares* são desenvolvidos, enfatizando o envolvimento ativo do usuário final, equipes reduzidas, alta agilidade e tolerância à mudanças, entrega evolutiva e procedimentos informais (Dingsoeyr, Falessi, & Power, 2019; Khalid, Butt, Jamal, & Gochhait, 2020; Silva & Lovato, 2016). Esta flexibilidade inspirou grandes organizações a aplicarem os métodos ágeis em seus complexos projetos almejando alcançar altos níveis de desempenho, contudo, os métodos ágeis como *Scrum* (Chen, Chen, Lien, &

Huang, 2019; Schwaber & Sutherland, 2017), *Extreme Programming* (XP) (Chen et al., 2019; Wood, Michaelides, & Thomson, 2013), *Lean Software Development* (Poppendieck & Cusumano, 2012), *Adaptive Software Development* (Highsmith, 2000), entre outros, foram projetados originalmente para equipes pequenas e centralizadas (Dikert et al., 2016; Uludag, Kleehaus, Caprano, & Matthes, 2018).

Em contrapartida, projetos de larga escala exigem um volume maior de recursos, seja no tamanho da equipe, tamanho do código, orçamento ou tamanho do projeto (Abrar et al., 2019), gerando assim uma dificuldade natural para o atingimento de níveis satisfatórios de qualidade e desempenho. Diante deste cenário, a inclusão de métodos ágeis em escala apresenta desafios decorrentes da interligação entre vários programas e da atuação de dezenas de equipes trabalhando em um só produto (Pitkänen, 2015). Além disto, o gerenciamento de tarefas e de múltiplas equipes se torna uma adversidade para a organização, que necessita coordenar o trabalho desenvolvido (Scheerer et al., 2014), muitas vezes por equipes que estão distribuídas em diversas localizações geográficas (Dikert et al., 2016).

Em busca de solucionar questões como a dificuldade de aplicação da metodologia ágil

em sua essência, questões relacionadas à multiplicidade de equipes e sua dispersão geográfica, muitas organizações sustentaram-se na utilização de métodos ágeis em larga escala como o *Scrum @ Scale* (Sutherland, 2020), *Scaled Agile Framework* (SAFe) (Knaster & Leffingwell, 2019), *Nexus* (Schwaber, 2018), *Large-Scale Scrum* (LeSS) (Larman, 2010), entre outros. Além disso, tais métodos são suportados por combinações de várias técnicas e procedimentos do *Agile* e do *Lean*, que permitem a otimização do processo de desenvolvimento de *software* em grandes escalas (Alqudah & Razali, 2016). Logo, entende-se que tais métodos podem contribuir no aumento do desempenho operacional ao mesmo tempo em que os valores ágeis são incorporados aos valores da organização (Iivari & Iivari, 2011).

Dado que uma organização pode ter parte de sua equipe praticando o ágil e parte utilizando técnicas e ferramentas tradicionais de gerenciamento de projetos no início de sua jornada de escalada do ágil, ela deve ter em mente que escalar as práticas ágeis não é uma tarefa fácil, bem como não acontecerá em um curto espaço de tempo (Penha, Silva, & Russo, 2020). Diante deste contexto, o presente artigo tem por objetivo apresentar os desafios da aplicação do ágil escalado em projetos de *software* caracterizados por alta complexidade e longa duração, isto sendo feito por meio de um estudo de caso em uma organização financeira do Brasil.

A relevância desta pesquisa é justificada pela ausência de literatura sobre a abordagem dos desafios e dificuldades encontradas por organizações do setor financeiro instaladas no Brasil que optam pela adoção de metodologias ágeis escaladas em seus projetos de desenvolvimento de *software*. Desse modo, esta pesquisa considera o estudo de caso de uma organização multinacional do mercado financeiro que atua a décadas com desenvolvimento de *software*, se beneficia dos métodos ágeis a anos e que considerou a implementação do *SAFe* (*Scaled Agile Framework*) como metodologia ágil de escala em seus projetos de tecnologia em busca da maximização dos benefícios propostos inicialmente pelas metodologias ágeis.

No que tange ao método utilizado para a elaboração desta pesquisa, realizou-se a coleta e análise de documentação eletrônica sobre o projeto, disponibilizada em rede interna de dados da organização (Yin, 2001), e efetuou-se a observação do andamento do projeto por parte dos pesquisadores, atuantes junto ao projeto no papel de consultores do mesmo.

Em síntese, o presente trabalho está estruturado da seguinte maneira: a seção inicial, composta pela introdução; a segunda seção constituída pelo referencial teórico necessário para a sua fundamentação; a terceira seção, tratando dos procedimentos metodológicos para a elaboração desta pesquisa; a quarta seção explica e caracteriza o caso a ser estudado; a quinta seção é destinada

a análise e discussão do caso; a sexta seção expõe as considerações e contribuições.

## 2 Referencial teórico

Esta seção apresenta os principais conceitos explorados. Primeiramente são apresentados os pressupostos da metodologia ágil de gerenciamento de projetos. Logo em seguida são discutidas metodologias ágeis escaladas e na sequência são apresentados alguns desafios para escalar a metodologia ágil.

### 2.1 Métodos Ágeis

A área de gerenciamento de projetos de desenvolvimento de *software* tem como alguns de seus diversos desafios, compreender de forma clara quais são os reais problemas dos seus clientes e entregar a eles soluções que atendam de forma efetiva aos seus propósitos. Como consequência destas dificuldades, não são raras as situações em projetos de engenharia de *software* onde apenas ao final do projeto, percebe-se que aquilo que foi desenvolvido não é exatamente o que o solicitante esperava (Silva & Lovato, 2016).

Assim, em fevereiro de 2001, o encontro de um grupo composto por dezessete pessoas, simpatizantes com a necessidade de uma alternativa aos processos de desenvolvimento de *software* orientados por documentação, deu origem ao Manifesto Ágil de Desenvolvimento de *Software* (Beck et al., 2001). Criando um conjunto de processos e

técnicas que destacavam o alto grau de valorização nas interações entre equipes e clientes, além da rápida absorção de mudanças identificadas e entendidas como necessárias para manter a geração de valores dos artefatos resultantes destas interações (Khalid et al., 2020; Silva & Lovato, 2016).

Muitos métodos já existiam antes mesmo do Manifesto ágil, como *Extreme Programming* (XP) (Chen et al., 2019; Wood et al., 2013), *Lean Software Development* (Poppendieck & Cusumano, 2012), *Scrum* (Chen et al., 2019; Schwaber & Sutherland, 2017), entre outros, que contribuíram para a criação do Manifesto Ágil. O *Extreme Programming* (XP), segundo Beck (1999), é indicado para projetos de desenvolvimento terceirizado ou interno de sistemas de pequeno e médio porte onde os requisitos são vagos e propensos a mudar, explorando a redução do custo de mudança de *software* desenvolvendo aos poucos todas as atividades relacionadas, ao invés de planejar, analisar e projetar sua totalidade para um futuro longínquo.

Quanto à otimização do processo de criação de *software*, Poppendieck (2012) discorre sobre o *Lean Software Development* (LSD), reforçando para o fato de se interpretar o lean como um conjunto de princípios ao invés de um conjunto de práticas, assim possibilitando a aplicação dos conceitos enxutos aos processos de desenvolvimento de *software* e assim agregando melhorias tanto no processo quanto de qualidade.

O *Scrum*, segundo Cervone (2011), é construído com base em três componentes principais: funções, processos e artefatos. O *Scrum* está fundamentado em teorias empíricas de controle de processo, ou empirismo, empregando uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos, concentrando-se em como os membros da equipe devem funcionar para produzir um produto de maneira flexível em um ambiente de constante mudança (Schwaber & Sutherland, 2017).

O *Scrum Master* é o responsável por promulgar os valores e práticas do *Scrum*, remover os impedimentos e maximizar o valor criado pelo time de desenvolvimento (Bass, 2016; Schwaber & Sutherland, 2017), enquanto o *Product Owner* é o responsável por maximizar o valor do produto por meio da ordenação e priorização dos itens a serem desenvolvidos, garantindo que o *backlog* de produto esteja visível, transparente e claro para todos (Bass, 2016; Schwaber & Sutherland, 2017). Por fim, o *Development Team* é um conjunto de profissionais multifuncionais que determinam no âmbito técnico como o produto será concebido e realizam entregas do incremento deste produto ao final de cada *Sprint* (Bass, 2016; Schwaber & Sutherland, 2017; Silva & Lovato, 2016).

O *Product Backlog* é a lista ordenada e priorizada dos itens entendidos como necessários para o produto, com os itens mais prioritários com maior nível de detalhamento e estimativa de tempo mais precisa, sendo a

única origem de quaisquer requisitos e mudanças a serem feitos no produto ao longo do desenvolvimento do projeto (Schwaber & Sutherland, 2017; Silva & Lovato, 2016). O *Sprint Backlog* é o conjunto de itens do *Product Backlog* selecionados para o *Sprint*, tornando todo o trabalho do *Development Team* visível, direcionando para o atingimento da meta da *Sprint* (Sutherland, 2010). O incremento é a soma de todos os itens do *Product Backlog* completados durante uma *Sprint*, caracterizados por funcionalidades, melhorias ou correções (Silva & Lovato, 2016; Sutherland, 2010).

A *Sprint Planning* é a reunião de definição do que pode ser entregue e em como o trabalho será realizado (Schwaber & Sutherland, 2017), com o *Product Owner* apresentando ao *Development Team* o *backlog* para a *Sprint*, junto dos critérios de aceite, possibilitando ao time de desenvolvimento a estimativa do esforço para realizar o trabalho (Silva & Lovato, 2016). A *Daily Scrum Meeting* é um evento diário de quinze minutos destinado ao planejamento do trabalho do *Development Team* pelas próximas 24 horas e inspecionar o progresso da entrega total do *backlog* da *Sprint* e ao atingimento de seu objetivo (Schwaber & Sutherland, 2017). A *Sprint Review* é uma reunião não obrigatória, realizada no final da *Sprint* com o objetivo de inspecionar tanto o incremento gerado e adaptá-lo caso se faça necessário, quanto o próprio time *Scrum* (Schwaber & Sutherland, 2017).

### 2.3 Metodologias Ágeis Escaladas

De acordo com Boehm e Turner (2005), os benefícios a partir da utilização de metodologias ágeis são obtidos com maior facilidade com determinados tipos de projetos em contextos específicos. Estudos de projetos fora daquilo que é considerado como contexto ideal demonstraram que estes projetos são demasiado problemáticos (Dikert et al., 2016). Uma vez que, em projetos pequenos e independentes, as práticas ágeis são menos onerosas e de maior sintonia com as necessidades crescentes da indústria de *software*, além de lidar melhor com mudanças contínuas, frustram-se diante da dificuldade de sua ampliação e integração em organizações tradicionais de desenvolvimento de *software* (Boehm & Turner, 2005).

Um problema específico na aplicação do Ágil em projetos maiores e mais complexos está relacionado a coordenação de equipes, envolvendo preocupações na interface com outras unidades organizacionais, como recursos humanos, marketing, vendas e gerenciamento de produtos (Paasivaara & Lassenius, 2016). Ao estudar os diferentes aspectos da aplicação de metodologias ágeis em larga escala, o indicativo comum é que as pessoas usem o pensamento ágil para resolver problemas em diferentes disciplinas, como arquitetura, design, marketing, gerenciamento de portfólio ou gerenciamento de programas (Laanti, 2014).

Dingsøyr, Falessi e Power (2019) afirmam que a adoção de práticas ágeis em

escala podem levar as organizações a desenvolverem sistemas otimizados, de forma disruptiva e diretamente associados à cultura e ao ecossistema de atuação, desde que seja obtido suporte por parte da alta gestão, aliado a um time de desenvolvimento altamente motivado e confiante nos princípios, valores e práticas ágeis que sustentam a maneira como os resultados são produzidos.

Alqudah e Razali (2016) acrescentam que diante da ampliação dos desafios de se adotar métodos ágeis para desenvolver *softwares* considerando a necessidade de se dimensionar as metodologias e práticas ágeis, as empresas de grande porte optam por empregar o *Disciplined Agile Development* (DAD), LeSS, SAFe, Spotify, Nexus e RAGE, conseqüentemente sendo consideradas as principais metodologias ágeis de escalonamento adotadas em grandes organizações.

A estrutura do DAD permite integrar diversas metodologias utilizando um conjunto de princípios enxutos e práticas ágeis, a fim de preencher as lacunas durante o ciclo de vida do projeto, como equipes geograficamente dispersas (Paasivaara & Lassenius, 2016; Stojanov, Turetken, & Trienekens, 2015). Enquanto a prática do Less é aplicar o *framework Scrum* em múltiplas equipes (Stojanov et al., 2015). Segundo os autores Alqudah e Razali (2016) e Paasivaara e Lassenius (2016), a estrutura LeSS pode trabalhar com até 10 times Scrum, sendo representado cada equipe por dois membros,

junto com um único Product Owner para definir os itens do backlog do produto.

A metodologia do Spotify trabalha com múltiplas equipes auto-organizáveis, sendo que cada equipe (Squad) utiliza sua própria metodologia ágil, e cada Product Owner tem a responsabilidade por priorizar o trabalho realizado pela equipe e colaboram para manter um documento de script de alto nível mostrando para onde o Spotify como um todo está indo (Alqudah & Razali, 2016). O Nexus por sua vez está focado no desenvolvimento e manutenção de produto em escala e projeto de desenvolvimento de *software*. A sua estrutura é composta por funções, eventos e técnicas semelhante ao *framework Scrum*, porém, utilizando mais de um time *Scrum* em um único *product backlog* (Alqudah & Razali, 2016). Enquanto o RAGE outra metodologia do ágil escalado que utiliza o *framework Scrum* para gerenciar a tomada de decisão rápida em qualquer processo, ou seja, sua flexibilidade otimiza os processos utilizando conjunto de elementos como métricas, artefatos, funções e cerimônias (Alqudah & Razali, 2016).

### 2.3.1 Scaled Agile Framework (SAFe)

O *Scaled Agile Framework* (SAFe) baseia-se nos princípios de pensamento do *Lean*, mas não cobre todos os recursos dos aspectos de agilidade (Laanti, 2014), sendo adotado por um grande número de organizações para escalar o ágil em projetos de desenvolvimento de *software* (Putta,

Paasivaara, & Lassenius, 2018). Knaster e Leffingwell (2019) reiteram de que o *Scaled Agile Framework* é uma estrutura escalável e configurável que ajuda as organizações a entregar sistemas em um prazo de entrega sustentável, mais curto e com melhor qualidade e valor possíveis, sincronizando alinhamento, colaboração e entrega para várias equipes ágeis.

Os primeiros a adotar o *SAFe* relataram melhorias significativas em termos de produtividade e de qualidade, visto que a melhoria de produtividade e qualidade são preocupações fundamentais em qualquer organização (Laanti, 2014). No que tange aos benefícios do *SAFe*, cabe ressaltar alguns destaques como o fato de ser altamente prescritivo, fornecendo uma estrutura que viabiliza e facilita a transição para um *framework* ágil (Bass, 2016) e fornecer maior produtividade e consequente retorno do investimento (Leffingwell, 2011).

De acordo com Stojanov, Turetken e Trienekens (2015), o *SAFe* visa incorporar as práticas dos princípios ágeis e enxutos no nível organizacional, integrando os corpos de trabalho existentes no *Scrum* e *XP*, composto por uma estrutura dividida nos níveis de equipe, programa e portfólio, com limites arbitrários que servem de modelo para a abstração do escopo e da escala entre os níveis. Laanti (2014) ressalta que quando os métodos ágeis são utilizados em outras disciplinas organizacionais, além do desenvolvimento de *software*, é comum confiar em outros

princípios que são compatíveis com os princípios ágeis.

Sugerido para organizações de médio e grande porte, é considerado um dos *frameworks* de escalada ágil mais utilizado mundialmente, com alto índice de aceitação (Penha et al., 2020), e tem como valores essenciais de alinhamento, qualidade integrada, transparência e execução do programa, representando suas crenças fundamentais, consideradas a chave para a sua eficiência (*Core Values - Scaled Agile Framework*, n.d.).

Leffingwell, Yakyma, Knaster, Jemilo e Oren (2017) ratificam que o SAFe suporta tanto projetos que atuam com um número modesto de equipes ágeis, quanto projetos de grandes dimensões que comumente exigem centenas de profissionais para desenvolvê-lo e mantê-los. Brenner e Wunder (2015) dividem o SAFe em três níveis: o nível de portfólio, o nível de programa e o nível de equipe. Os níveis de programa e equipe, trabalham juntos em um contingente denominado *Release Train* (conjunto de times onde somados, possui entre 50 e 124 pessoas) atuante em um mesmo fluxo de valor. Este fluxo de valor deve efetivamente agregar valor ao negócio da organização e todas as equipes que compõem a *Release Train* devem possuir dependências entre si para conduzir a entrega de valor (Brenner & Wunder, 2015).

O nível de equipe deve ser composto por equipes ágeis contendo 7 pessoas, até duas delas devem ser responsáveis pela definição,

construção e teste das estórias de usuários em uma série de iterações e liberações. Laanti (2014) observa que, o nível de equipe pode trabalhar utilizando o método *Scrum*, o método Kanban, ou mesmo a combinação de ambos, reforçando para o fato de que as práticas do SAFe neste nível são fortemente compatíveis com os princípios ágeis, apesar de não serem suficientes para dizer como organizar com mais eficiência os níveis de portfólio e de programa.

#### 2.4 Desafios em torno das metodologias ágeis escaladas

Rigby, Sutherland e Noble (2018) estudaram o escalonamento ágil em centenas de empresas, incluindo pequenas empresas que administram toda a sua organização com métodos ágeis, empresas que nasceram ágeis e se tornaram ainda mais ágeis a medida que cresceram, e empresas que estiveram em transição de hierarquias tradicionais para ágeis. Os autores observaram histórias de sucesso, mas também de fracasso, com empresas que não obtiveram os resultados financeiros esperados por investidores ativos ou mesmo pelo conselho de suas respectivas organizações.

Em seus estudos, Boehm e Turner (2005) dividem as barreiras para o escalonamento ágil em três áreas: conflitos de processo de desenvolvimento, conflitos de processos de negócios e conflitos de pessoas, considerando esta última, como crítica para os gerentes de *software* de grandes organizações

em trazer abordagens ágeis para suportar seus projetos.

Segundo o 14th Annual State of Agile Report (2020), as três principais respostas citadas como desafios para a adoção e dimensionamento das práticas ágeis indicam que a cultura organizacional permanece como um obstáculo para o sucesso em muitas empresas, destacando com resultado de 48% a resistência às mudanças por parte das organizações, seguido por 46% referentes à falta de participação das lideranças organizacionais e 45% relacionados à inconsistência de práticas e processos entre os times.

Em seu trabalho, Penha et al. (2020) levantam pontos importantes durante a jornada das organizações para escalar as práticas ágeis, a considerar a necessidade de mudança cultural da organização, ações motivadoras para a adoção do ágil em todos os níveis organizacionais e a desmistificação sobre a ilusão de que isto inevitavelmente acarretará em mudanças na estrutura da organização.

De maneira complementar, Conboy (2019) identificou nove desafios de maior impacto para a implementação de ágil em larga escala nas organizações – definição de conceitos e termos, comparação e contraste de estruturas, prontidão e apetite por mudanças, balanceamento entre estrutura organizacional e *frameworks*, implementação *top-down versus bottom-up*, enfoque excessivo da adesão à estrutura em relação ao seu valor, falta de utilização em detrimento da falta de

evidências, preservar a autonomia dos desenvolvedores e o desalinhamento entre os processos e as estruturas do cliente.

Rigby, Sutherland e Noble (2018) reitera de que as empresas podem ser ágeis e escalar esta agilidade de forma eficaz, gerando benefícios substanciais, desde que de forma realista, pois nem todas as funções precisam ser organizadas em equipes ágeis, inclusive reforçando de que os métodos ágeis são considerados inadequados para certas atividades. Devido seu alto nível de prescrição, Ambler e Lines (2016) enfatizam que o *SAFe* pode restringir o desenvolvimento contínuo das equipes, enquanto Stojanov, Turetken e Trienekens (2015) ressaltam que o *SAFe* não está dentro das expectativas do *Agile*.

### 3 Procedimentos metodológicos

Este trabalho tem abordagem qualitativa e natureza descritiva, com a unidade de pesquisa caracterizada por um projeto de desenvolvimento de *software* de alta complexidade em uma empresa do setor financeiro. A utilização da abordagem qualitativa permite ao pesquisador coletar, interpretar e compreender o fenômeno estudado a partir da perspectiva e das ações dos indivíduos, grupos e organizações em seu ambiente ou contexto social (Creswell, 2010; Godoy, 1995; Neves, 1996). O estudo de caso permite a investigação da complexidade, tanto em situações problemáticas quanto bem-sucedidas, quando o pesquisador tem pouco controle sobre os acontecimentos, fornecendo

conhecimento amplo e detalhado mediante estudos aprofundados (Yin, 2001), considerando-se a coleta e análise de dados e a descrição das características de indivíduos, grupo, organizações e até comunidades a fim de atender a um estudo de natureza descritiva (Gil, 2009).

### 3.1 A empresa

A organização considerada para este estudo é caracterizada por ser uma das maiores empresas financeiras do Brasil, com início de suas operações na década de 1940, expandindo suas operações em nível internacional no início da década de 1980, tornando-se em 2008 uma das maiores instituições financeiras do Brasil e uma das 20 maiores do mundo em valor de mercado. O projeto destinou-se à ampliação das operações financeiras transacionadas via internet banking, considerando a criação de novas funcionalidades e a melhoria de funções já existentes, buscando proporcionar maiores níveis de iteração entre seus clientes e menores taxas de indisponibilidade sistêmica.

A motivação para a seleção da organização e do projeto se concentrou na possibilidade de efetuar de perto o acompanhamento e a observação de todo o andamento do projeto *in loco* desde o seu início com a atuação dos pesquisadores no papel de consultores do projeto, permitindo uma análise aprofundada do seu desenvolvimento, das ações e atitudes dos envolvidos e da programação e execução das

dinâmicas e cerimônias relacionadas à metodologia ágil escalada abordada, da documentação prévia existente e da documentação construída ao longo do desenvolvimento do projeto, bem como de fatores que não necessariamente estão documentados na base de conhecimento da organização.

### 3.2 Coleta e análise de dados

A documentação utilizada para a auxiliar a análise do estudo de caso objeto desta pesquisa consistiu a partir da coleta de dados primários, informações de documentações digitais disponibilizadas pela organização em sua rede interna de dados, do acompanhamento e observação da evolução do processo de desenvolvimento do *software* proposto pela organização. Logo, pretende-se obter informações e evidências de materiais que ainda não passaram pela análise de um pesquisador, ou que podem ser reexaminados (Marconi & Lakatos, 2003; Martins & Theóphilo, 2009).

A atuação dos pesquisadores como parte integrante das equipes do projeto na posição de consultores viabilizou a observação do projeto antes, durante e após o seu desenvolvimento, além de auxiliar o processo de obtenção das permissões para o acesso à toda documentação eletrônica referente ao projeto e sobre as disposições metodológicas da empresa de como seus colaboradores devem atuar em relação aos projetos, dispostas pela

ferramenta *Atlassian Confluence*, hospedada em rede interna e utilizada para organizar a base de conhecimento da organização, predispondo para um estudo de caso longitudinal.

A documentação digital constituiu-se de informações expressas por documentos explicativos sobre os conceitos das metodologias ágeis e de seu escalonamento, descrição de particularidades sobre as metodologias considerando a estratégia organizacional, discriminação de papéis e responsabilidades dentro da organização e de seus projetos, informações técnicas sobre os projetos e informações relacionadas às equipes e sua iteração, detalhados por meio das cerimônias de retrospectiva e revisão das *sprints* dos projetos.

Por fim, tais informações foram adquiridas durante o período compreendido entre o primeiro semestre de 2019 e segundo semestre de 2020, em adição ao conhecimento oriundo da observação e acompanhamento do processo de desenvolvimento do projeto pelos pesquisadores atuando no papel de consultores do mesmo, considerado como o caso único e objeto deste estudo.

A análise dos dados deu-se a partir da coleta de evidências compostas por documentação eletrônica disposta na rede interna de dados da empresa e disponibilizadas pela ferramenta *Atlassian Confluence*, considerada como repositório e base de conhecimento da organização conforme sugerido por Yin (2001), em adição à análise

oriunda da observação dos pesquisadores, por meio de sua atuação no projeto no papel de consultores. A base de conhecimento tecnológico da organização, utilizada pelos pesquisadores reuniu documentos digitais relacionados aos seus projetos desde o ano de 2015, enquanto a coleta por parte dos pesquisadores ocorreu entre o início de 2019 e final de 2020.

#### 4 Apresentação e análise dos resultados

Esta seção apresenta as informações relativas ao projeto e características que ajudam a compreender o fenômeno da escalar ágil. São apresentadas as ações de seleção, customização e utilização da metodologia ágil escalada para o projeto, bem como os resultados do projeto analisado que demonstram os desafios de escalar a metodologia ágil.

##### 4.1 O Projeto

O estudo de caso analisado nesta pesquisa é um projeto de desenvolvimento de *software* de alta complexidade e longa duração, com início em janeiro de 2020 e previsão de término das entregas planejadas em dezembro de 2020. O projeto foi desenvolvido na região sudeste do Brasil, por uma organização do setor financeiro. Este projeto foi considerado de alta complexidade devido ao seu alto valor orçamentário, grande quantidade de profissionais envolvidos e a possibilidade de inclusões de demandas

regulatórias que não estariam no escopo inicial do projeto e que poderiam desembocar em penalidades para a organização em forma de multas com valores em torno dos milhares de reais.

Segundo Schwaber (2004) o desenvolvimento de *software* é um empreendimento complexo e tal complexidade é algo impossível de ser ignorado. Baccarini (1997) complementa que a complexidade em projetos consiste em diversas partes inter-relacionadas e variadas, podendo ser operacionalizadas em termos de diferenciação e interdependência, aplicando-se a qualquer dimensão do projeto, relevantes para o seu processo de gerenciamento.

O projeto até o momento desta pesquisa contou com a colaboração de 98

profissionais, número este considerado insuficiente pela organização para dar vazão a quantidade de demandas e com alta probabilidade de ampliação. Também foram apuradas as horas utilizadas por todas as equipes envolvidas diretamente no projeto, totalizando mais de 250 mil horas de projeto. Também foi considerada a inclusão de diversas demandas regulatórias, o que influenciou para que a empresa centralizasse a maior parte destas demandas em uma única equipe de cada *release train*. A seguir é possível verificar a Tabela 1, que apresenta a quantidade de *features* desenvolvidas ao longo dos três primeiros trimestres de 2020.

**Tabela 1.** Features desenvolvidas

| Features/Trimestre | 1 Trimestre | 2 Trimestre | 3 Trimestre |
|--------------------|-------------|-------------|-------------|
| Features Business  | 36          | 86          | 105         |
| Features Enabler   | 10          | 25          | 41          |
| Total              | 46          | 111         | 146         |

Fonte: Elaborado pelos autores.

De acordo com a Tabela 1, percebe-se o crescimento considerável de incrementos desenvolvidos para atender às necessidades do projeto, reforçando a necessidade de ampliação da quantidade de equipes envolvidas para prover entregas dentro de prazos anteriormente estimados. Cabe afirmar que os números informados não compreendem demandas regulatórias, oriundas de adequações exigidas

por parte de órgãos reguladores nacionais e internacionais, tampouco medidas impostas pelo governo brasileiro que podem surgir e são tratadas com prioridade máxima devido às altas multas impostas em caso de atraso.

O projeto teve como escopo a otimização e o incremento de funcionalidades existentes do sistema, considerando a esfera de contas, tarifação e derivados, além da criação

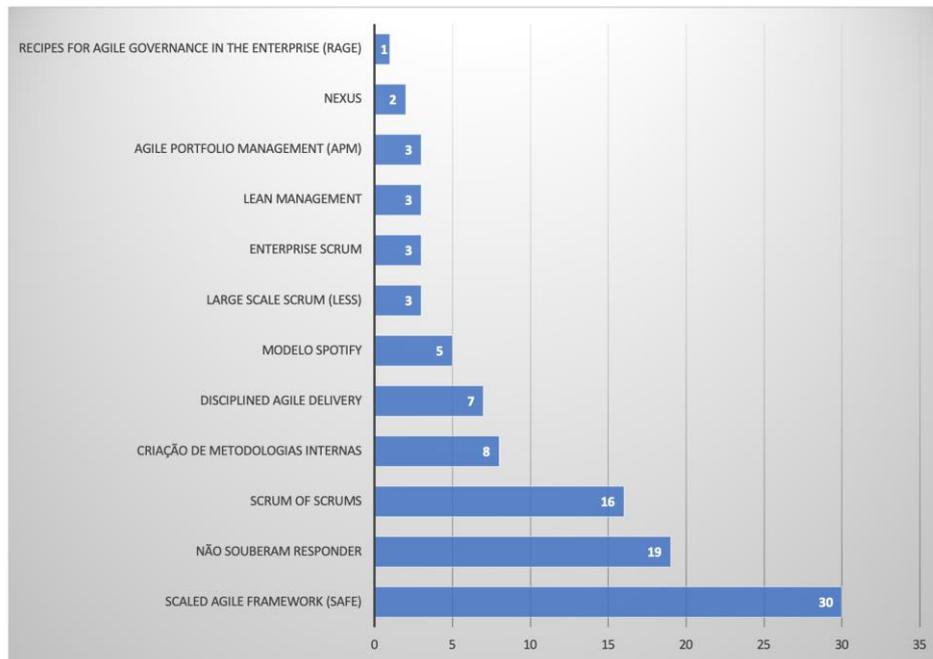
de novas funcionalidades que proporcionem melhorias na interação entre os próprios colaboradores da organização, seus clientes e acionistas.

#### 4.2 Seleção, Customização e Utilização da Metodologia Ágil Escalada para o Projeto

Previamente ao desenvolvimento do projeto aqui analisado como estudo de caso, a organização passou inicialmente pelo processo de transformação ágil, dando início aos novos projetos baseando-se na abordagem *Scrum*, que de acordo com 14th Annual State of Agile Report (Digita.ai, 2020) é responsável por 58% do resultado de sua pesquisa sobre qual metodologia ágil comumente utilizada pelas organizações. A princípio, os projetos iniciais utilizando a abordagem ágil foram aqueles considerados de curto espaço de tempo e com baixa complexidade, considerados como projetos de *benchmarking* para validação e verificação das viabilidades técnica, tecnológica e de processos para sua posterior implementação nos demais projetos da organização. Em seguida, deu-se o processo de disseminação da metodologia pela organização, alcançando desta forma projetos maiores em níveis de escopo e de complexidade.

Diante do dinamismo tecnológico e do ambiente da organização, caracterizado por constantes mudanças em seus produtos e do alto nível de integração entre eles, a abordagem *Scrum* em sua essência se tornou insuficiente para prover níveis competitivos de aumento da produtividade, do valor das entregas para os clientes, de redução de riscos e de aumento da qualidade em seus projetos de tecnologia, levando a organização a buscar uma alternativa viável e conhecida no mercado.

Este cenário direcionou a organização na adoção do SAFe devido a sua popularidade entre as organizações desenvolvedoras de *software*. Essa situação corrobora o resultado do 14th Annual State of Agile Report (Digita.ai, 2020), o qual reitera que 30% dos respondentes utilizam o SAFe como metodologia ágil escalada, conforme apresentado na Figura 1. Putta et al. (2018) expressam que 28% dos entrevistados consideram o SAFe como um *framework* proeminente em relação a outras estruturas como *Scrum of Scrum* e LeSS, apresentando melhorias de produtividade e qualidade no processo (Laanti, 2014). Além de possibilitar a introdução de novas práticas e conceitos que se integram às práticas ágeis básicas e em escala (Stojanov et al., 2015).



**Figura 1.** Principais metodologias ágeis escaladas pelas organizações.

**Fonte:** Adaptado de Digita.ai (2020).

Outras metodologias ágeis oferecem soluções que geralmente se limitam a reestruturar a metodologia *Scrum*, combinando-a com metodologias tradicionais de desenvolvimento de *software* para a resolução de problemas (Alqudah & Razali, 2016), como o *Scrum of Scrums*, com 16% de resultados, DAD, LeSS, *Enterprise Scrum* e *Lean Management* representadas com 4% consecutivamente e em processo de ascensão, Nexus e *Agile Portfolio Management* (APM) com 3% dos resultados cada.

O *SAFe* é tido como uma metodologia ágil de escala baseada em desenvolvimento

ágil, desenvolvimento enxuto de produtos, pensamento sistêmico e *DevOps*, o que o torna amplo, profundo e escalável (Knaster & Leffingwell, 2019). Sua aplicação visa à melhoria dos quesitos anteriormente mencionados, o que permite a empresa voltar a tornar-se competitiva no mercado financeiro por meio do seu *software*. A dinâmica dos processos e as iterações entre os envolvidos nos diversos papéis dentro da metodologia *SAFe* está disposta conforme a Figura 2.

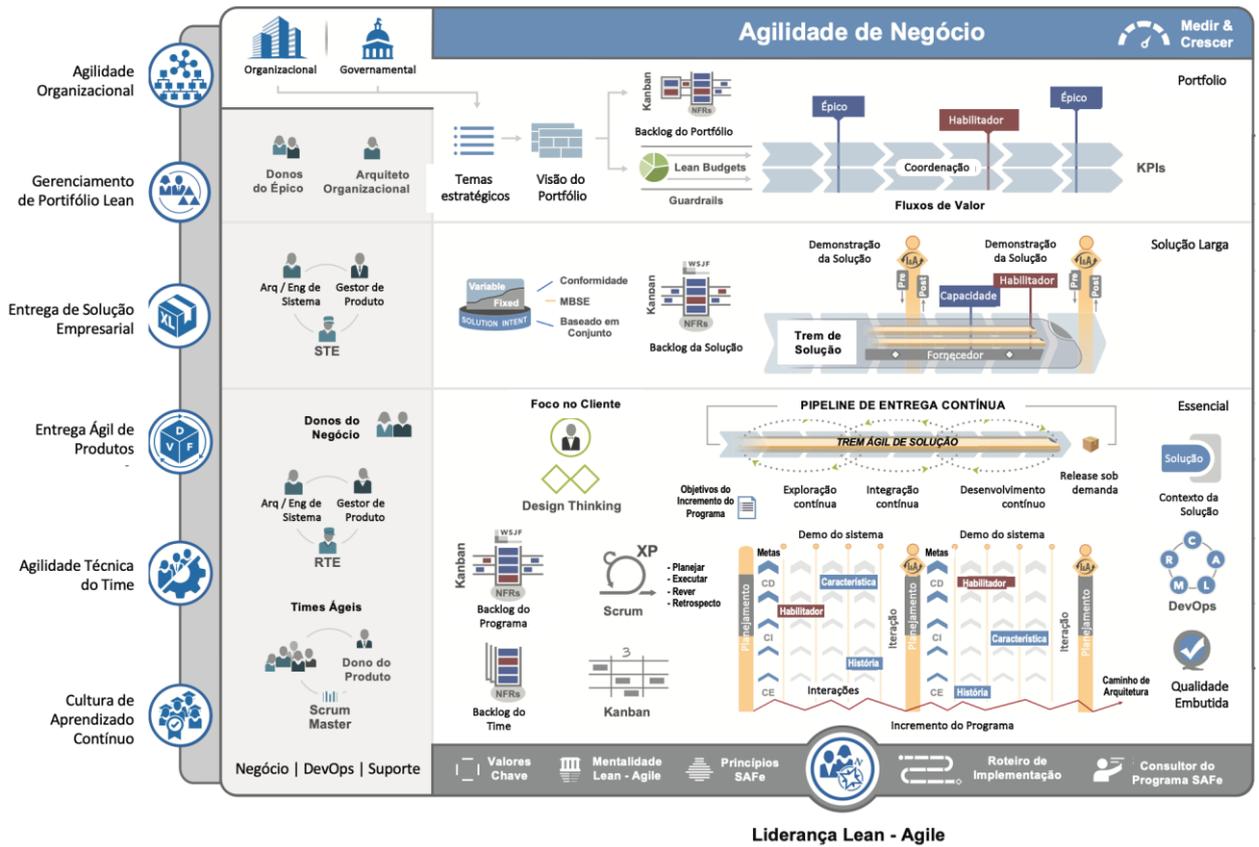
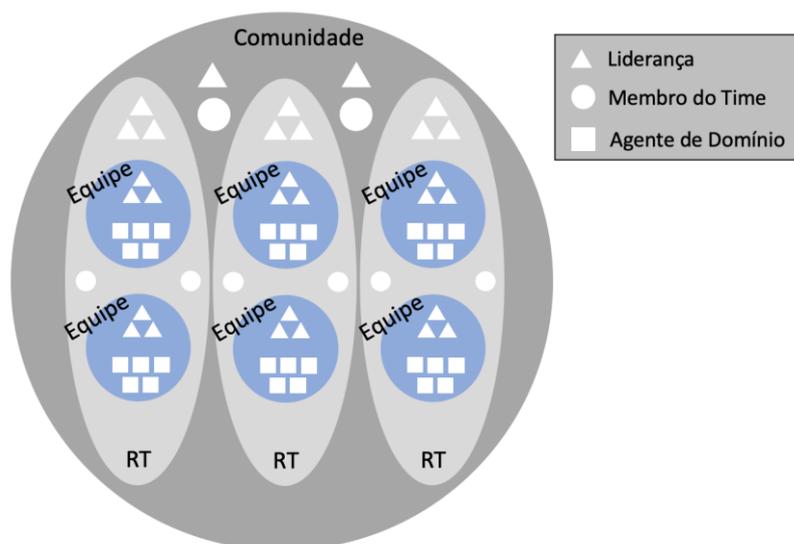


Figura 2. Scaled Agile Framework.

Fonte: Adaptado de SAFe 5.0 Framework - SAFe Big Picture (n.d.).

A organização utilizou-se do conceito estrutural de comunidades, considerada como parte da transformação de uma organização tradicional para uma organização enxuta e ágil (Paasivaara & Lassenius, 2014), além de

possibilitar a replicação das mesmas características dos níveis mais elementares da sua estrutura em níveis mais abrangentes. A Figura 3 demonstra a estrutura de equipes de desenvolvimento aplicada pela organização.



**Figura 3.** Estrutura das comunidades de desenvolvimento de software  
**Fonte:** Elaborado pelos autores.

O nível mais elementar de uma *Release Train* é um time composto por liderança e membros, e um conjunto de times podem ser agrupados em uma comunidade ou em uma *Release Train*. Cada *Release Train* é composta por 5 a 12 times ágeis, um mínimo de 50 e um máximo de 125 integrantes e inclui as regras e a infraestrutura necessária para entregar soluções totalmente testadas e funcionais (Leffingwell et al., 2017).

A Comunidade observada nesta pesquisa estava disposta em 4 *Release Trains* e cada uma delas compreendia entre 5 e 9 equipes de desenvolvimento. A Tabela 2 apresenta a configuração de cada *release train* do projeto e sua evolução no sentido de ampliação de equipes para aumentar a vazão de entregas.

**Tabela 2.** Quantidade de equipes por Release Train

| RT/Trimestre           | 1 Trimestre | 2 Trimestre | 3 Trimestre |
|------------------------|-------------|-------------|-------------|
| RT Contas e Tarifas    | 3 equipes   | 4 equipes   | 4 equipes   |
| RT Movimentação        | 5 equipes   | 8 equipes   | 9 equipes   |
| RT Onboarding          | 7 equipes   | 7 equipes   | 8 equipes   |
| RT Tarifas e Rotativos | 5 equipes   | 7 equipes   | 8 equipes   |

**Fonte:** Elaborado pelos autores.

No contexto organizacional, cada *Release Train* dispõe de uma liderança que é composta por três profissionais – o líder de produto, o líder técnico e o líder operacional. Dependendo da complexidade do assunto tratado pela comunidade são considerados mais três profissionais – o Agende de Domínio, responsável pela coesão tática entre as equipes no nível de *Release Train* e o Especialista de Domínio, responsável pela coesão estratégica entre as equipes no nível de comunidade e, por último, o *Business Owner* responsável pelos resultados em nível de negócio tanto da *Release Train* quanto da comunidade, além de atuar como direcionador estratégico.

Os times também seguem algumas das características padrões para times ágeis, conforme proposto por SCRUMstudy (2017), sendo multidisciplinares e autônomos, pequenos, independentes e estáveis. Os times envolvidos no projeto eram compostos igualmente pelo seguinte conjunto de profissionais: um *Product Owner*, representado pela área de negócio ou até pelos representantes da gestão de portfólio, um líder técnico, responsável por fornecer informações de priorização considerando o ponto de vista técnico ou arquitetural comumente representado por um engenheiro de sistemas, um líder de equipe, encarregado por manter a harmonia da equipe e de seu trabalho e, por último, os *team members*, grupo composto por analistas, programadores, testadores e

quaisquer papéis necessários para o atingimento do propósito da equipe.

Até o momento do desenvolvimento desta pesquisa, a comunidade estudada totalizava 98 profissionais, dos quais 33 foram alocados ao projeto como recursos terceirizados, com o auxílio de consultorias parceiras da organização. Para o projeto, considera-se a grande probabilidade de aumento do número de colaboradores e a consequente criação de novas equipes de desenvolvimento seguindo a estrutura proposta pela organização, com apoio das consultorias parceiras para a obtenção dos recursos com maior nível de aderência em relação aos conhecimentos técnicos necessários para a realização das funções, à metodologia ágil escalada utilizada no projeto e aos processos internos da organização.

#### 4.2 Achados

Esta pesquisa teve por objetivo apresentar os desafios da aplicação do ágil escalado em projetos de *software* caracterizados por alta complexidade e longa duração, por meio da realização de um estudo de caso em uma organização financeira do Brasil. Amparando-se na análise da base de conhecimento da organização, composta por documentação digital disponibilizada em rede interna corporativa e no acompanhamento e observação do processo de desenvolvimento do projeto pelos pesquisadores assumindo o papel de consultores do projeto, os resultados indicaram que apesar da organização já

executar seus projetos de *software* em conjunto da abordagem de uma metodologia ágil escalada, ainda enfrenta dificuldades relacionadas ao desalinhamento entre a metodologia, a complexidade e a duração do projeto.

A existência de desafios em decorrência do processo de escalonamento da metodologia ágil dentro da organização confirmou-se ao verificar a dificuldade de efetuar o planejamento integrado das entregas dos times de desenvolvimento. A dificuldade em reorganizar as entregas quando considerada a entrada de uma demanda regulatória, que não foi prevista anteriormente no planejamento e nos refinamentos dos respectivos *backlogs* das equipes. A redução da qualidade dos entregáveis, motivada pelo desalinhamento dos papéis e das responsabilidades atribuídas aos colaboradores, colaborando para que a organização buscasse por ações mitigadoras para este e outros desafios, desta forma cooperando para a proposta desta pesquisa.

Em detrimento de questões envolvendo a integridade física e a saúde dos colaboradores, viu-se a necessidade de manter as operações remotamente. Qualquer organização que tenha suas equipes de desenvolvimento geograficamente distribuídas enfrenta desafios significativos com o desenvolvimento ágil (Kalenda, Hyna, & Rossi, 2018; Paasivaara & Lassenius, 2016).

Estas dificuldades foram observadas como o baixo envolvimento, ou ausência de membros dos times durante as cerimônias de reunião diária, reunião de planejamento e mesmo a reunião de demonstração dos entregáveis. Dikert et al. (2016) discorrem sobre a consequente redução da sensação de proximidade quando surgem necessidades de telecomunicação e a dificuldade na organização de reuniões frequentes, o que corrobora para o fato de que o desenvolvimento ágil enfatiza a comunicação constante e o espírito de equipe, fatores negativamente influenciados pelo ambiente distribuído (Kalenda et al., 2018).

A Tabela 3 mostra as taxas de ausência e ociosidade dos participantes de acordo com as principais cerimônias. Para a amostragem de ausências, considerou-se a não participação dos colaboradores que não estavam dentro das seguintes características – férias, afastamento por questões de saúde, folga por compensação de horas e quaisquer tipos de licenças com base nos aceites de *invites* transmitidos via correspondência digital disparada pelo *software Microsoft Outlook*. Para a amostragem da ociosidade, foram considerados os profissionais que adentravam as reuniões e ficavam com seu status em “modo de espera”, o que significava que o colaborador estava em outra chamada paralela, desta forma, configurando apenas uma presença simbólica.

**Tabela 3.** Percentuais de presença, abstenção e ociosidade dos colaboradores nas cerimônias

| <b>Cerimônia</b>           | <b>Presença</b> | <b>Ausência</b> | <b>Ociosidade</b> |
|----------------------------|-----------------|-----------------|-------------------|
| <i>Daily Scrum Meeting</i> | 77%             | 23%             | 5%                |
| <i>Sprint Planning</i>     | 89%             | 11%             | 8%                |
| <i>Release Planning</i>    | 92%             | 8%              | 18%               |

Fonte: Elaborado pelos autores.

Outra dificuldade percebida está relacionada ao planejamento integrado dos times, visto que alguns destes times estavam envolvidos constantemente com demandas regulatórias. Apesar deste tipo de demanda fora de escopo já ser algo conhecido e metodologias ágeis como o próprio *Scrum* buscarem contorná-las por meio de seções periódicas de refinamento, a organização ainda encontra dificuldades em lidar com elas, estas demandas de caráter emergencial normalmente são incorporadas aos projetos e tidas com alto nível de criticidade, urgência e impacto, dificultando o seu planejamento prévio, iteração com os demais times e interligação com as demais funcionalidades do sistema.

A organização optou pela utilização do *framework* SAFe, porém considerou a mudança de nomes relacionados aos papéis desempenhados, mas não de suas responsabilidades, como o *System and Solution Architect/Engineering* que é o responsável pela definição e comunicação da visão técnica e arquitetônica compartilhada na *Release Train*. Esta posição foi renomeada para *Technical Lead*, com funções adicionais como apoiar o *Product Owner* na priorização de *features* de qualidade e de risco, evolução e melhorias de plataforma. O papel do *Scrum Master* sofreu a

maior carga de modificações, pois além da mudança de nomenclatura, agora chamado de *Team Lead*. A sua responsabilidade como gestor foi ampliada de uma única equipe para múltiplas equipes de desenvolvimento, além de acompanhar os indicadores de qualidade e riscos, acompanhar as entregas das equipes, agregar visão de qualidade e também apoiar o *Product Owner* na priorização de *backlog*.

Estas adaptações efetuadas pela organização corroboram para a afirmação de Rigby et al. (2018) de que o aumento do ágil escalado dentro da organização traz valores e princípios ágeis para as operações de negócios e funções de suporte, mesmo que muitas atividades de rotina permaneçam. Aqui fazemos a ressalva de que estas modificações de nomenclatura relacionadas aos papéis e responsabilidades dentro da organização e da metodologia não possibilitaram evidenciar melhorias no processo organizacional.

Em relação aos colaboradores, notou-se o agravamento dos níveis de colaboração, inicialmente provocado pela não identificação dos colaboradores com seus novos papéis e responsabilidades e posteriormente pela desmotivação a partir do entendimento de que seus novos papéis estão recebendo maior grau de responsabilidades quando considerados os

frameworks tradicionais de escala do ágil, além da falta de reflexo deste acúmulo de responsabilidades em seus salários.

No contexto capacitação dos colaboradores, a organização disponibilizou treinamentos internos para os colaboradores do projeto em uma ferramenta de *e-learning* própria como forma de nivelar os conhecimentos acerca das novidades envolvendo os novos processos, papéis e responsabilidades. Contudo, devido a indisponibilidade da ferramenta para os colaboradores terceirizados, percebeu-se um efeito inverso ao proposto pela ferramenta, pois o alinhamento não atingia todos os colaboradores em todos os níveis operacionais do projeto. O impacto deste desalinhamento foi minimizado pela organização ao disponibilizar posteriormente para os demais colaboradores terceirizados os treinamentos de forma parcial dentro da ferramenta *Atlassian Confluence*, que é um ambiente acessível tanto para colaboradores internos quanto terceiros.

Problemas com a garantia e perda de qualidade no código em consequência de responsabilidades adicionais, pressão sobre as equipes ou até mesmo uma má definição do conceito de “pronto” são algumas das barreiras enfrentadas pelas organizações ao escalar o ágil (Kalenda et al., 2018). Neste estudo corroboramos para tal afirmação ao verificar a queda de qualidade e atrasos das entregas, provocados pela entrada de demandas regulatórias que não tinham sido previamente planejadas no projeto, resultando em *lead time*

das entregas muitas vezes superior a 100 dias. Este cenário no contexto de projetos vai ao encontro aos achados de Nuottila, Aaltonen e Kujala (2016) de que o maior desafio para as organizações estaria relacionado à integração de práticas ágeis aos processos existentes no ambiente do projeto. Portanto, pode-se observar o considerável desgaste dos integrantes das equipes quanto ao processo de integração dos artefatos desenvolvidos em uma única solução.

Em relação a gestão visual do projeto, verificou-se uma considerável redução da gestão, uma vez que inicialmente os times atuavam no mesmo ambiente físico, utilizando quadros brancos espalhados pelos setores com *post-its* representando a imagem atual do andamento de cada parte do projeto, facilitando sua visualização por qualquer colaborador sem a necessidade de interrupção das atividades de outros colaboradores para qualquer tipo de alinhamento. A partir do momento em que a continuidade do projeto ocorreu de forma remota, a gestão visual só pode ser viabilizada pela utilização do *Microsoft Whiteboard*, única ferramenta homologada pela organização para atuar com dados sensíveis, e de acordo com os próprios colaboradores, uma ferramenta muito aquém das necessidades do projeto e em comparação a outras ferramentas disponíveis no mercado.

Quanto à documentação do projeto, notou-se uma deficiência por parte das equipes de desenvolvimento em abordar não apenas as questões estritamente técnicas envolvendo codificação como também decisões e definições

em níveis mais altos alcançando até a estratégia organizacional. Conboy e Carroll (2019) ressaltam sobre a importância de se criar evidências para apoiar a estrutura ágil em grande escala e proporcionar a transformação organizacional, além da necessidade de identificar e contextualizar problemas para então poder oferecer orientações e estabelecer melhores práticas.

Boehm e Turner (2005) apontam para a necessidade das organizações em aprender e acomodar as questões relacionadas a recursos humanos, descrições de cargos, habilidades necessárias e recompensas individuais, uma vez que os departamentos de recursos humanos e seus procedimentos tendem a atrapalhar o empoderamento das pessoas em busca de abordagens não tradicionais. Com a mudança de nomenclatura dos cargos para as funções a serem desempenhadas no projeto, foi possível perceber o desconforto por parte dos colaboradores, mediante a inconclusividade sobre a relação entre o cargo ocupado e as demandas que o profissional efetivamente desenvolve. Isto foi observado tanto em posições operacionais quanto em posições gerenciais e de liderança, envolvendo profissionais anteriormente classificados como gerentes e coordenadores, que passaram a compor a liderança operacional, de produtos e técnica, bem como se tornarem responsáveis por uma série de atividades anteriormente não lhes eram designadas.

Ainda cabe ressaltar que mesmo considerando a adoção de uma metodologia ágil

escalada, a organização ainda atua com um modelo de recompensa, não centrado na equipe, o que é considerado por Dikert et al. (2016) como sendo o departamento de recursos humanos trabalhando contra a adoção do ágil uma vez que as recompensas são individuais e não coletivas. Essa situação é considerada problemática por estar atrelada ao desempenho individual e agindo contra o pensamento centrado na equipe, o que refletiu negativamente nos indicadores organizacionais relacionados ao *turnover* das equipes, possibilitando verificar que mais de 90% destas equipes tinham índices variando entre 10% e 20% por *sprint* do projeto.

As questões observadas na organização sobre recompensas e incerteza sobre a relação cargo *versus* atividades desempenhadas e a falta de possibilidade de planejamento com demandas regulatórias foram apontadas pelos próprios membros das equipes como fatores contribuintes para o desgaste tanto individual quanto coletivo dos colaboradores. Portanto, a data de entrada em vigor de uma alteração ou implementação de lei, estabelece um prazo para o projeto, o que conflita diretamente com os métodos ágeis (Nuottila et al., 2016).

Devido ao alto nível de complexidade do projeto, observou-se a necessidade de ampliação das equipes envolvidas. Em detrimento desta necessidade, novas equipes foram constituídas repentinamente, forçando a organização a buscar por novos profissionais no mercado, o que constituía uma tarefa complicada, uma vez que aqueles perfis

considerados ideais pela organização para desempenhar as atividades necessárias eram escassos ou preenchiam apenas parcialmente a lacuna de especificidades teóricas e mesmo técnicas necessárias, reforçando a afirmação de Putta et al. (2018) sobre o desafio e a dificuldade de se encontrar os profissionais certos e com a experiência necessária.

A organização mostrou-se relativamente rígida em relação à adoção de ferramentas mais comuns disponíveis no mercado para auxiliar nas atividades de gestão do projeto, concentrando-se na utilização do *Atlassian JIRA* para efetuar o acompanhamento do andamento das atividades das equipes relacionadas ao projeto em conjunto ao *Atlassian Confluence* como base de conhecimento organizacional, responsável por armazenar toda a documentação do projeto, diretrizes ligadas às customizações do *framework SAFe* e até documentação ligada às cerimônias e dinâmicas efetuadas. Diante deste contexto, muitos colaboradores buscavam a otimização de suas atividades considerando a utilização de ferramentas externas e não homologadas, como Miro, Mural, Kudobox, SurveyMonkey e FunRetro, por proporcionarem maiores níveis de colaboração e interatividade em relação às ferramentas propostas pela organização.

Durante a cerimônia de planejamento integrado, os times interagem apenas com as demandas passíveis de análise, dimensionamento de tempo e esforço e, durante os ciclos de desenvolvimento ou *Sprints*,

quando surge uma demanda regulatória os times se organizam para poder dar vazão tanto à demanda regulatória quanto às demandas previamente planejadas para aquela *Sprint*, o que desemboca em maior esforço por parte de todos os integrantes do time, desde a equipe de desenvolvimento e testes até as lideranças envolvidas direta e indiretamente ao processo. Tal esforço é refletido em maiores cargas de trabalho, com os times atuando além da carga de 40 horas semanais, estendendo as jornadas aos finais de semana e feriados para conseguir efetuar tanto a entrega da demanda regulatória quanto das demandas anteriormente planejadas.

Como consequência direta deste processo, observou-se a estafa dos times, fator que influenciou diretamente os indicadores da empresa relacionados a estabilidade dos times e seus índices de *turnover*. Para estes indicadores, a organização considera como regra que, a partir da base de colaboradores da equipe, verifica-se mensalmente a sua composição contabilizando entradas, saídas e o tamanho da equipe. Em seguida é aplicada a fórmula  $0,5 * \frac{[SOMA(ENTRADAS) + SOMA(SAÍDAS)]}{[POSIÇÕES OCUPADAS DURANTE O PERÍODO]}$ . Como valores aceitáveis pela organização, tal indicador não deve ultrapassar 25% de *turnover*, o que já é considerado um valor relativamente elevado e, ainda assim as equipes não conseguem manter seus índices inferiores ao máximo permitido.

As descobertas adquiridas nesta pesquisa contribuem para a análise de possíveis ações mitigadoras para a implementação de

uma metodologia ágil escalada, em detrimento das dificuldades encontradas durante o seu processo de adoção em projetos complexos de desenvolvimento de *software*.

## 6 Considerações finais

O objetivo desta pesquisa foi apresentar os desafios da aplicação do ágil escalado em projetos de *software* caracterizados por alta complexidade e longa duração, por meio da realização de um estudo de caso em uma organização financeira do Brasil. O achado de maior impacto para esta pesquisa está diretamente relacionado à dificuldade no gerenciamento e administração de demandas regulatórias, impostas pelo governo ou até mesmo por órgãos reguladores nacionais e internacionais. Estas demandas são variáveis em todos sentidos, aumentando consideravelmente a complexidade do projeto e terminando por gerar outros desafios evidenciados neste estudo. Tais demandas apesar de monitoradas pela organização, são comumente caracterizadas pelo alto impacto financeiro, uma vez que o atraso de suas entregas é responsável pela geração de multas milionárias para a organização.

Outro ponto importante sobre estas demandas, está na sua inclusão ao escopo do projeto de forma brusca, sem dimensionamento do esforço para sua conclusão e devido ao seu alto nível de prioridade, consequentemente acarretando paralisação parcial do escopo que já está em andamento e fatalmente gerando

atrasos nas entregas das demandas planejadas em *sprints* e *releases*.

Assim, como resultado final, esta pesquisa contribui para a prática, trazendo à luz a necessidade de análise de possíveis ações mitigadoras para a implementação de uma metodologia ágil escalada, em detrimento das dificuldades encontradas. Entre as dificuldades levantadas por esta pesquisa, a recorrência dos colaboradores pela utilização de ferramentas não homologadas pela organização em virtude das limitações encontradas nas poucas ferramentas homologadas, o desalinhamento relacionado pela falta de clareza no entendimento de papéis e responsabilidades dos colaboradores quando efetuadas mudanças de nomenclatura dos papéis e a consequente fragilização do sistema de recompensas, a inclusão de trabalho não planejado ao escopo do projeto por questões regulatórias de órgãos nacionais e internacionais se mostrou como o maior desafio da organização. Este trabalho não planejado culmina em vários outros problemas para o projeto, como a paralisação do que está em desenvolvimento para priorizar este trabalho não planejado, o consequente atraso das demandas previamente planejadas, maior desgaste das equipes de desenvolvimento devido à maior carga de atividades e necessidade de trabalhar horas extras, resultando em índices altos de *turnover* na organização.

Esta pesquisa possui algumas limitações, como limitar-se em analisar um único projeto de desenvolvimento de *software*,

desconsiderando outros projetos de mesmas características a nível comparativo. Outro ponto de limitação desta pesquisa consistiu em desconsiderar projetos curtos ou de baixa complexidade e projetos oriundos de outros tipos de atividades fora da temática de desenvolvimento de *software* destinado a atender a área financeira. Este trabalho propõem para pesquisas futuras estudos de maior profundidade que contemplem a análise de um conjunto de projetos complexos correlatos, ou não à área financeira, corroborando para o aumento da abrangência desta pesquisa, viabilizando a obtenção de novos achados e novas constatações.

### Referências

- Abrar, M. F., Khan, M. S., Ali, S., Ali, U., Majeed, M. F., Ali, A., Amin, B., & Rasheed, N. (2019). Motivators for Large-Scale Agile Adoption from Management Perspective: A Systematic Literature Review. *IEEE Access*, 7, 22660–22674. <https://doi.org/10.1109/ACCESS.2019.2896212>
- Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 828–837. <https://doi.org/10.18517/ijaseit.6.6.1374>
- Ambler, S. W., & Lines, M. (2016). The Disciplined Agile Process Decision Framework. In *Software Quality. The Future of Systems- and Software Development* (Vol. 238, Issue August 2015, pp. 3–14). [https://doi.org/10.1007/978-3-319-27033-3\\_1](https://doi.org/10.1007/978-3-319-27033-3_1)
- Baccarini, D. (1997). *10.The Logical Framework Method for Defining Project Successbaccarini1999*.
- Bass, J. M. (2016). Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75, 1–16. <https://doi.org/10.1016/j.infsof.2016.03.001>
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70–77. <https://doi.org/10.1109/2.796139>
- Beck, K., Beedle, M., Bennekum, A. Van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. The Agile Alliance. <http://agilemanifesto.org/>
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30–39. <https://doi.org/10.1109/MS.2005.129>
- Brenner, R., & Wunder, S. (2015). Scaled Agile Framework: Presentation and real world example. *2015 IEEE 8th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2015 - Proceedings*, 4–5. <https://doi.org/10.1109/ICSTW.2015.7107411>
- Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems and Services*, 27(1), 18–22. <https://doi.org/10.1108/10650751111106528>
- Chen, P. S., Chen, G. Y. H., Lien, S. F., & Huang, W. T. (2019). Using Scrum and unified modelling language to analyze and design an automatic course scheduling system. *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A*, 42(6), 534–543. <https://doi.org/10.1080/02533839.2019.1613930>

- Conboy, K., & Carroll, N. (2019). Implementing Large-Scale Agile Frameworks: Challenges and Recommendations. *IEEE Software*, 36(2), 44–50. <https://doi.org/10.1109/MS.2018.2884865>
- Core Values - Scaled Agile Framework. (n.d.). Retrieved October 30, 2020, from <https://www.scaledagileframework.com/safe-core-values/>
- Creswell, J. W. (2010). Creswell, J.W. projeto de pesquisa- método qualitativo, quantitativo e misto. Tradução de Luciana de oliveira da rocha. 2 Ed. Porto Alegre-artmed, 2007. In *Artmed*. <http://ir.obihiro.ac.jp/dspace/handle/10322/3933>
- Digita.ai. (2020). *14th annual State of Agile report*.
- Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87–108. <https://doi.org/10.1016/j.jss.2016.06.013>
- Dingsoeyr, T., Falessi, D., & Power, K. (2019). Agile Development at Scale: The Next Frontier. *IEEE Software*, 36(2), 30–38. <https://doi.org/10.1109/MS.2018.2884884>
- Gil, A. C. (2009). *Estudo de Caso*.
- Godoy, A. S. (1995). Introdução à pesquisa qualitativa e suas possibilidades. *Revista de Administração de Empresas*, 35(2), 57–63.
- Highsmith, J. A. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House Publishing.
- Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, 53(5), 509–520. <https://doi.org/10.1016/j.infsof.2010.10.008>
- Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, 30(10), 1–24. <https://doi.org/10.1002/smr.1954>
- Khalid, A., Butt, S. A., Jamal, T., & Gochhait, S. (2020). Agile Scrum Issues at Large-Scale Distributed Projects: Scrum Project Development at Large. *International Journal of Software Innovation*, 8(2), 85–94. <https://doi.org/10.4018/IJSI.2020040106>
- Knaster, R., & Leffingwell, D. (2019). *SAFe 4.5 Distilled* (R. Knaster & D. Leffingwell (Eds.)). Addison-Wesley.
- Laanti, M. (2014). Characteristics and principles of scaled agile. *Lecture Notes in Business Information Processing*, 199, 9–20. [https://doi.org/10.1007/978-3-319-14358-3\\_2](https://doi.org/10.1007/978-3-319-14358-3_2)
- Larman, C. (2010). Practices for Scaling Lean & Agile Development. In *Addison-Wesley*. <http://ptgmedia.pearsoncmg.com/images/9780321636409/samplepages/0321636406.pdf>
- Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and Enterprise*. Addison-Wesley.
- Leffingwell, D., Yakyma, A., Knaster, R., Jemilo, D., & Oren, I. (2017). *SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering*. [https://books.google.com.co/books?id=BPADAAAQBAJ&dq=safe+4.0+Reference+Guide:+Scaled+Agile+Framework@+for+Lean+Software+and+Systems+Engineering&source=gbs\\_navlinks\\_s](https://books.google.com.co/books?id=BPADAAAQBAJ&dq=safe+4.0+Reference+Guide:+Scaled+Agile+Framework@+for+Lean+Software+and+Systems+Engineering&source=gbs_navlinks_s)
- Marconi, M., & Lakatos, E. (2003). Fundamentos de metodologia científica. In *Editora Atlas S. A.* <https://doi.org/10.1590/S1517-97022003000100005>
- Martins, G. A., & Theóphilo, C. R. (2009). *Metodologia da Investigação Científica*

- para Ciências Sociais Aplicadas - 2ª Ed.
- Neves, J. L. (1996). Pesquisa Qualitativa - Características, Usos e Possibilidades. *Caderno de Pesquisas Em Administração*, 1(3), 5.
- Nuottila, J., Aaltonen, K., & Kujala, J. (2016). Challenges of adopting agile methods in a public organization. *International Journal of Information Systems and Project Management*, 4(3), 65–85. <https://doi.org/10.12821/ijispm040304>
- Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization - Case Ericsson. *Information and Software Technology*, 56(12), 1556–1577. <https://doi.org/10.1016/j.infsof.2014.06.008>
- Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations - A research proposal and a pilot study. *ACM International Conference Proceeding Series*, 24-May-201. <https://doi.org/10.1145/2962695.2962704>
- Penha, R., Silva, L. F. da, & Russo, R. D. F. S. M. (2020). Escalando as práticas ágeis. *Revista de Gestão e Projetos*, 11(2), 1–11. <https://doi.org/10.5585/gep.v11i2.18133>
- Pitkänen, A. (2015). *Agile Transformation: A case study*. 90 + 4. <http://urn.fi/URN:NBN:fi:aalto-201512165681>
- Poppendieck, M., & Cusumano, M. A. (2012). Lean Software Development: A Tutorial. *IEEE Software*, 29(5), 26–32. <https://doi.org/10.1109/MS.2012.107>
- Putta, A., Paasivaara, M., & Lassenius, C. (2018). Adopting Scaled Agile Framework (SAFe): A multivocal literature review. *ACM International Conference Proceeding Series*, Part F1477. <https://doi.org/10.1145/3234152.3234164>
- Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review*, May-June 2, 1–15.
- SAFe 5.0 Framework - SAFe Big Picture. (n.d.). Retrieved October 30, 2020, from <https://www.scaledagileframework.com/>
- Scheerer, A., Hildenbrand, T., & Kude, T. (2014). Coordination in large-scale agile software development: A multiteam systems perspective. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 4780–4788. <https://doi.org/10.1109/HICSS.2014.587>
- Schwaber, K. (2004). Agile Project Management with Scrum. In *Microsoft Press*.
- Schwaber, K. (2018). *Guia do Nexus - O Guia Definitivo para escalar o Scrum com oNexus: As Regras do Jogo*. 15.
- Schwaber, K., & Sutherland, J. (2017). *Guia do Scrum - Um guia definitivo para o Scrum: As regras do Jogo*. 21. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>
- SCRUMstudy. (2017). *Um Guia para o Conhecimento em Scrum (Guia SBOK™)* (3rd ed.).
- Silva, E. C. da, & Lovato, L. A. (2016). Framework Scrum: Eficiência em Projetos de Software. *Revista de Gestão e Projetos*, 07(02), 01–15. <https://doi.org/10.5585/gep.v7i2.330>
- Stojanov, I., Turetken, O., & Trienekens, J. J. M. (2015). A Maturity Model for Scaling Agile Development. *Proceedings - 41st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2015*, 446–453. <https://doi.org/10.1109/SEAA.2015.29>
- Sutherland, J. (2010). *Scrum Handbook*. Scrum Training Institute.
- Sutherland, J. (2020). *The Definitive Guide to the Scrum@Scale Framework*. 19.
- Uludag, O., Kleehaus, M., Caprano, C., & Matthes, F. (2018). Identifying and structuring challenges in large-scale agile

development based on a structured literature review. *Proceedings - 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference, EDOC 2018*, 191–197. <https://doi.org/10.1109/EDOC.2018.00032>

Wood, S., Michaelides, G., & Thomson, C. (2013). Successful extreme programming: Fidelity to the methodology or good teamworking? *Information and Software Technology*, 55(4), 660–672. <https://doi.org/10.1016/j.infsof.2012.10.002>

Yin, R. K. (2001). *Estudo de Caso - Planejamento e Métodos*.