UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUAN LIMA DA SILVA

UM ESTUDO SOBRE DESVIOS NO FRAMEWORK DE DESENVOLVIMENTO DE SOFTWARE SCRUM

São Leopoldo
2020

UM ESTUDO SOBRE DESVIOS NO FRAMEWORK DE DESENVOLVIMENTO DE SOFTWARE SCRUM

Luan Lima da Silva *

Cristiano Bonato Both **

Resumo: O desenvolvimento ágil de software utilizando o *framework* Scrum popularizou-se no mercado facilitando respostas a mudanças e trazendo maior produtividade para as equipes. Apesar da grande adesão ao *framework* nota-se que muitas empresas e equipes estão aderindo apenas algumas práticas, ou também adaptando as sugestões propostas pela ferramenta, tendo assim, consequências não esperadas. Existem poucos estudos na literatura sobre as consequências que essas alterações ou não adesões completas no *framework* acarreta. Portanto, esta pesquisa tem como objetivo em um estudo de caso analisar os desvios cometidos em uma equipe de desenvolvimento que utiliza o *framework* Scrum, classificando-os com suas respectivas áreas do Scrum, trazendo as consequências que os mesmos podem acarretar, e apontando sugestões para a equipe. Para alcançar esse objetivo, foi realizada uma pesquisa com agentes envolvidos no desenvolvimento de software em uma equipe de uma empresa multinacional situada no Vale dos Sinos, no Rio Grande do Sul, que utilizam Scrum, apontando as consequências de seus desvios e propondo sugestões de melhorias. Assim, alertando equipes que pretendem adotar e que já adotaram o *framework* Scrum sobre consequências de escolhas em relação a aplicação do *framework.*

Palavras-chave: Desenvolvimento de Software. Melhoria Contínua. Metodologias Ágeis. Scrum.

1 INTRODUÇÃO

A adoção das metodologias de desenvolvimento ágeis vem tomando a indústria de desenvolvimento. A metodologia ágil mais comum e utilizada atualmente é o Scrum. (DINGSØYR *et al.*, 2012). Entretanto, conforme apontado por Eloranta, Koskimies e Mikkonen (2016), quando verificada a aplicação do Scrum nas organizações, apenas algumas práticas desta ferramenta são adotadas. Levando então a deterioração, ou, não total aproveitamento de alguns processos dentro da equipe de desenvolvimento, consequentemente, encaminhando a equipe para desvios das práticas propostas pela ferramenta.

Tendo em vista o estudo de Eloranta, Koskimies e Mikkonen (2016), exemplificando casos de equipes que não adotam completamente o Scrum e as consequências destas adaptações propõem-se a seguinte questão: *Existem desvios nos processos de Scrum? Os desvios dos processos de Scrum estão causando falhas nas equipes de desenvolvimento de software?*

* Graduando em Análise e Desenvolvimento de Sistemas pela Unisinos. Email: luan_lima_da_silva@hotmail.com
** Professor Orientador, Doutor em Ciência da Computação. Email: cbboth@unisinos.br

Esta pesquisa guia-se com o objetivo de identificar desvios na aplicação do *framework* Scrum em um estudo de caso de uma equipe que gerencia diversos projetos de software em uma empresa multinacional, classificá-los com suas causas e inferir possíveis consequências. A fim de atingir o objetivo geral desta pesquisa foram elaborados os seguintes objetivos específicos, sendo primeiramente verificar a existência de desvios na aplicabilidade do Scrum. Caso verificada e constatada a existência de desvios, se identifica e estruturam-se os mesmos. E por fim, inferir possíveis consequências dos desvios, propondo recomendações para as adaptações.

O foco dessa pesquisa está na identificação e apresentação dos desvios do Scrum com base em dados empíricos. Essencialmente, o objetivo é esclarecer o conceito relativamente difuso do Scrum. (MARTINS; DE MENEZES, 2016). Identificando e explorando os desvios reais do Scrum em seu contexto. Como citado por Janes e Succi (2012), deve-se entender as consequências e reais aplicações de novas ferramentas e tendências na área de tecnologia, validando seus pontos altos e baixos.

Pode-se ver a utilidade deste trabalho principalmente em duas direções. Primeiro, expor para a equipe onde se ocorre o presente estudo de caso sobre os desvios que estão cometendo, a fim de alertá-los sobre perigos das escolhas e inferir possíveis melhorias para o mesmo. O estudo também se vê necessário para empresas e equipes que estão começando a usar o Scrum, devem estar cientes de desvios que podem parecer razoáveis, mas que podem ser realmente prejudiciais. A empresa deve entender o contexto e os problemas normalmente envolvidos em tais desvios, como exemplo de não realização de cerimonias, times maiores do que sugeridos, pessoas exercendo mais de um papel dentro dos projetos, entre outros fatores. E por segundo ponto nas utilidades desta pesquisa, entender os desvios bem motivados do padrão, o guia do Scrum fornece informações para refinar a metodologia para atender aos objetivos das empresas.

O presente estudo inicia pela fundamentação teórica, apresentando conceitos como: Metodologias ágeis e Scrum. Em seguida conta com uma revisão bibliográfica, apresentando artigos relacionados que são usados como base para a pesquisa em questão. Após encontra-se a seção de materiais e métodos onde se descreve a metodologia utilizada por esta presente pesquisa. Seguido pela discussão dos resultados, e por fim, a conclusão do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção destina-se a apresentação do referencial teórico necessário para o entendimento deste trabalho. São apresentados os principais conceitos abordados nesta pesquisa bem como os trabalhos relacionados.

2.1 Metodologias Ágeis

O *movimento ágil* da indústria de software teve seu início oficialmente demarcado pela assinatura do Manifesto Ágil no ano de 2001. (BECKET *et al*., c2001). Neste encontro, estavam reunidos especialistas e influentes da área de desenvolvimento de software, onde cada um contava com suas peculiaridades de cada esfera do desenvolvimento de software onde atuavam. O objetivo desse encontro foi buscar melhorias no desempenho de projetos de software. (MARTINS; MENEZES, 2016). O grupo buscava alternativas para as metodologias de desenvolvimento da época, que eram consideradas engessadas, por possuírem fortes regulamentações – como no modelo cascata para desenvolvimento de software, assim caracterizado pelo grupo como burocrático e lento comparado com a forma como os engenheiros desempenhavam seu trabalho. Conforme Becket *et al.* (c2001), a reunião deu origem ao Manifesto Ágil para desenvolvimento de software, baseados em quatro valores, como descrito no manifesto:

a) indivíduos e interações, mais que processos e ferramentas;
b) software em funcionamento, mais que documentação abrangente;
c) colaboração com o cliente, mais que negociação de contrato;
d) responder a mudanças, mais que seguir um plano.

Além do Manifesto Ágil, o grupo introduziu um conjunto de doze princípios. Alguns exemplos dos princípios são: Foco no cliente, simplicidade, auto-organização, interação, conversa cara-a-cara, dentre outros. (BECKET *et al*., c2001). A natureza iterativa do pensamento Ágil permite interações constantes com *stakeholders,* correções e mudanças ao decorrer do projeto, e atualizar os requerimentos do projeto conforme necessário. Times auto gerenciáveis são unidades básicas para realizar entregas efetivas e eliminar desperdícios e ineficiências na execução de projetos. (ABRAHAMSSON; CONBOY; WANG, 2009). Como uma das bases também do manifesto ágil se destaca para o presente estudo o objetivo de rápida

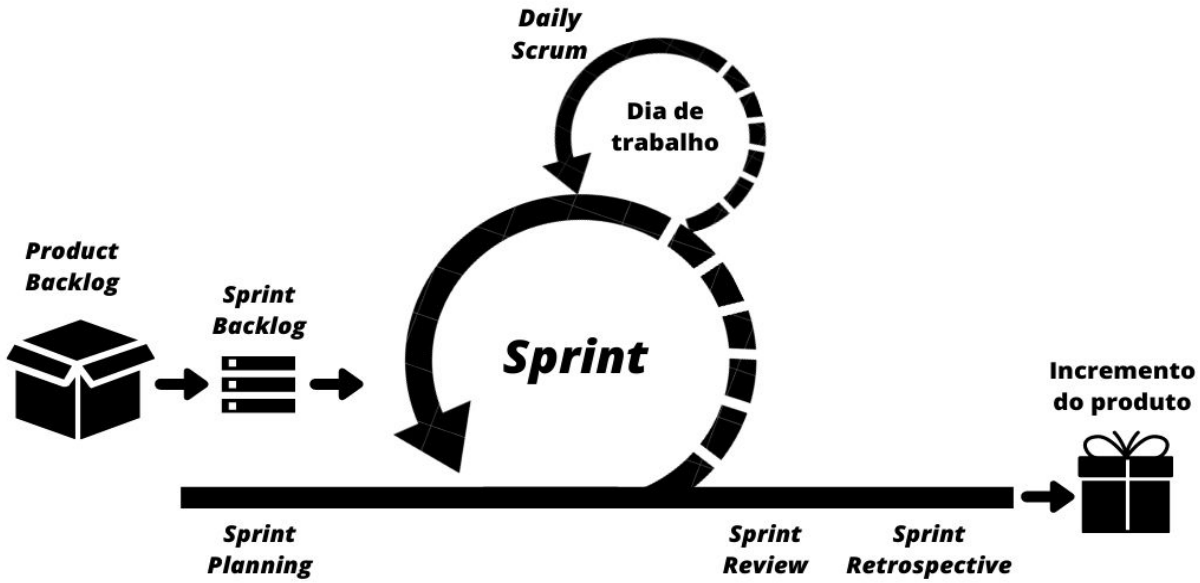
resposta a mudanças e adaptações que a equipe tem de tomar para atingir seus objetivos e os de seus *stakeholders.* Sendo neste ponto que se encontra a problemática do presente estudo, quando essas mudanças se tornam um empecilho para o time, ao invés de uma melhoria.

2.2 Scrum

Por conta do Manifesto Ágil, *frameworks* e metodologias de desenvolvimento de software ganharam espaço no mercado na época, e o Scrum foi um deles. (MARTINS; DE MENEZES, 2016). O Scrum é uma das metodologias de desenvolvimento Ágil mais conhecida atualmente. (DINGSØYR *et al*., 2012). O *framework* Scrum de desenvolvimento de software foi originalmente proposta por Schwaber (1997), na qual se baseia no *product and development methodology* descrito por Takeuchi e Nonaka (1986). O Scrum traz uma proposta de autoridade de tomada de decisão para o nível operacional dos projetos, assim tornando os times auto gerenciáveis. (ANNOSI *et al*., 2016).

Segundo o Guia do Scrum, Schwaber e Sutherland (2013), o Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. Conforme Schwaber e Sutherland (2013), existem três pilares que apoiam a implementação do processo: transparência, inspeção e adaptação. O Scrum propõe que o time, o *Scrum Team*, deve ser composto pelos seguintes papéis principais: *Product Owner, Scrum Master e Development Team*. Além dos integrantes o Scrum propõe o uso de alguns artefatos, como: *Product Backlog*, *Sprint Backlog*, etc. Além disso, propõe alguns eventos importantes, sendo os principais: *Sprint Planning, Sprint, Daily Scrum, Sprint Review e Sprint Retrospective.* (MARTINS; DE MENEZES, 2016). Na Figura 1 ilustra-se como acontecem e se relacionam as cerimônias e artefatos do Scrum.

Figura 1 - Metodologia Scrum



Fonte: Elaborado pelo autor.

No *framework* proposto pelo Scrum, os requerimentos levantados são primeiramente armazenados no *Product Backlog,* no formato que é chamado de *Product Backlog Items.* Esses itens são implementados em *sprints*, ou iterações cujo tamanho não deve ser maior do que poucas semanas, onde se recomenda *sprints* de duas a quatro semanas, para se ter o *feedback* constante das entregas e alinhamento de expectativas. Para cada *sprint*, alguns *Product Backlog Items* são selecionados e refinados em uma lista de tarefas chamada de *Sprint Backlog,* que é consistida pelos *Sprint Backlogs Items.* Juntamente, essas tarefas formam um plano de implementação para os selecionados *Product Backlog Items,* provendo que é possível criar estimativas de trabalho mais assertivas para todos esses itens. (CHO, 2008). Depois do trabalho de refinamento, os *Sprint Backlog Items*, são implementados durante a execução da *sprint.* O *Development Team* time que executará a *sprint*, tem autonomia para definir as melhores ferramentas, técnicas e melhores formas de executar a *sprint* e desenvolver seus *Sprint Backlog Items.* (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

Após o término de cada *sprint*, uma parte do sistema, deve ser entregue e disponível ao cliente. Para aprender com cada *sprint*, é realizada a cerimônia de *Sprint Retrospective* para o time considerar como as maneiras de trabalhar podem ser melhoradas. (ELORANTA; KOSKIMIES; MIKKONEN, 2016). Como citado anteriormente, apenas três papéis são definidos no Scrum: o *Product Owner* que é o responsável por gerenciar o *Product Backlog,* garantindo o mesmo em ordem de prioridade e consistência. O *Development Team* formado pelos desenvolvedores e são responsáveis por executar as *sprint.* E o papel de *Scrum Master*

existente para eliminar qualquer bloqueio ou problema que o *Development Team* possa ter, e garantir que os processos de Scrum ocorram da melhor maneira. (MARTINS; DE MENEZES, 2016).

Segundo Eloranta, Koskimies e Mikkonen (2016), apesar dos ideais apresentados formarem um *framework* ágil e coerente para executar tarefas de desenvolvimento, é comum que organizações sigam o Scrum parcialmente ou de forma que algumas práticas propostas são totalmente omitidas. Omitir algumas práticas do Scrum ou mudar essas práticas de forma que não estejam alinhadas com o Guia do Scrum é comumente chamada de *ScrumBut*. Algumas dessas mudanças estão sendo apontadas como uma fonte de perda de muitas vantagens do Scrum. No presente estudo essas mudanças, adaptações, ou não total adoção do Scrum são denominados desvios.

2.2.1 Desvios

Como o Scrum, assim como as metodologias ágeis, fomentam a melhoria contínua das equipes que o adotam, junto com essas melhorias podem acontecer algumas adaptações que inicialmente podem parecer razoáveis e válidas para o contexto da equipe, mas que posteriormente podem acarretar problemas para a equipe. (ELORANTA; KOSKIMIES; MIKKONEN, 2016).

Segundo Eloranta, Koskimies e Mikkonen (2016), os desvios podem acontecer também no caso de equipes que estão adotando a ferramenta Scrum, onde adotam só algumas práticas aconselhadas pelo Scrum, e desviam de outras práticas, podendo ter alguns efeitos colaterais. Observa-se que tais desvios podem acontecer tanto em equipes que recentemente adotaram, ou estão adotando Scrum, como em equipes que já utilizam o modelo há tempos, e decidem adapta-lo. Eloranta, Koskimies e Mikkonen (2016) classificaram alguns desvios como antipadrões para a metodologia de desenvolvimento Scrum em sua pesquisa denominada: *Exploring scrumBut - an empirical study of Scrum antipatterns*. A pesquisa foi feita na Finlândia com 18 representantes de 11 diferentes empresas. No Quadro 1 são apresentados alguns exemplos dos 14 antipadrões que foram classificados ao final da pesquisa dos autores.

Referente aos antipadrões descritos no Quadro 1, Eloranta, Koskimies e Mikkonen (2016) destacam o problema de *Product Owner* sem autoridade, já que esta pessoa é responsável por gerenciar o *Backlog* do Produto e decidir quais itens maximizam o valor do produto. E o contexto deste problema é uma compreensão insuficiente do papel de um *Product Owner*, pessoas adequadas não estão motivadas para usar o Scrum, falta de interesse em geral

pelo papel de *Product Owner*, em grandes organizações onde a responsabilidade por produtos e desenvolvimento pode ser fragmentada. Causando consequências como de o *Product Owner* não ter autoridade para decidir quais itens serão implementados e quais serão descartados. Não podendo realmente tomar decisões sobre o produto e, assim, a priorização perde seu significado, no final, tudo deve ser implementado.

Outro antipadrão mapeado na pesquisa de Eloranta, Koskimies e Mikkonen (2016) é referente ao tamanho das *sprints* variando, não sendo fixos. A recomendação do Guia do Scrum, Schwaber e Sutherland (2013), sugerem que as *sprints* durem de duas a quatro semanas, mas que o tamanho definido se mantenha até o final do projeto, ou mude o mínimo possível. O contexto onde foi mapeado este antipadrão identificado foi em projetos onde faltava disciplina dos envolvidos, e o cliente causando interrupções no time. Trazendo consequências do tipo sem visibilidade de progresso realizado, ficando mais desfocado e os possíveis problemas no desenvolvimento não aparecem. E a equipe sem compromisso com os objetivos da *sprint*.

E um terceiro antipadrão destacado nessa presente pesquisa, e um dos identificado no estudo de Eloranta, Koskimies e Mikkonen (2016), é sobre os clientes dos projetos causando interrupções na equipe. Schwaber e Sutherland (2013) recomendam que as equipes devem ser protegidas contra interrupções externas, para focar em seu trabalho principal, desenvolver o projeto. O contexto onde foi identificado esse antipadrão relaciona-se com o já citado de *Product Owner,* somando com o de *Scrum Master*, sem autoridade, deixando a equipe exposta a interferências externas. Acarretando consequências de nos casos de interrupção do fluxo de trabalho da equipe, onde reduz consequentemente a eficiência do trabalho realizado. Além de que novos recursos podem rastrear o produto sem o envolvimento do *Product Owner*, pois o cliente está dando novo trabalho diretamente à equipe de desenvolvimento.

Quadro 1 – Antipadrões identificados no estude de Eloranta *et al.* (2016)

Nome do antipadrão	**Recomendação do Scrum**	**Contexto**	**Consequência**
Product Owner sem autoridade	O *Product Owner* é responsável pelo *Backlog* do Produto	Compreensão insuficiente do papel de um *Product Owner*.	A priorização perde seu significado.
Tamanho da *sprint* variando	*Sprints* de duas a quatro semanas	Falta de disciplina, cliente causando interrupções no time.	Sem compromisso com os objetivos da *sprint*.
Cliente causando interrupções na equipe	As equipes não devem ter interrupções externas	*Product Owner* e *Scrum Master* sem autoridade	Reduz eficiência da equpe

Fonte: Eloranta, Koskimies e Mikkonen (2016, p. 195).

Nem sempre os desvios são de todo o mal, podendo tais desvios da metodologia e dos seus guias serem benéficos para a equipe, sendo bem justificáveis e com consequências positivas. No exemplo citado por Eloranta, Koskimies e Mikkonen (2016), onde um dos antipadrões traçados pelos pesquisadores é a sugestão do *Product Owner* da equipe não ser alguém do cliente, e sim um colega da empresa que está prestando o serviço de desenvolvimento, para evitar pressões e interrupções na equipe de desenvolvimento. Exemplos como esse é levado em conta para a presente pesquisa como recomendações para melhorias nos guias da ferramenta Scrum.

2.3 Trabalhos Relacionados

Nesta seção, encontram-se os filtros usados para busca dos trabalhos relacionados desta pesquisa, apresentação de dois trabalhos relacionados e um quadro comparativo entre os mesmos.

2.3.1 Revisão bibliográfica sistemática

Para o início deste trabalho, foi realizada a Revisão Bibliográfica Sistemática. Assim, permitindo uma maior familiaridade com o problema a ser estudado e o entendimento do estado da arte na área pesquisada sobre desvios na implementação das atividades propostas pelo *framework* Scrum. O objetivo da pesquisa tem o propósito de identificar lacunas no mapeamento de desvios na aplicação no *framework* Scrum em projetos de desenvolvimento de software na literatura.

Quadro 2 – *String* de busca

TITLE-ABS-KEY (scrum) AND TITLE-ABS-KEY (agile) AND (TITLE-ABS-KEY (deviation) OR TITLE-ABS-KEY (scrumbut) OR TITLE-ABS-KEY (anti-patterns) OR TITLE-ABS-KEY (dark)) AND (LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019) OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017) OR LIMIT-TO (PUBYEAR,2016))

Fonte: Elaborado pelo autor.

O objetivo da *string* de busca é filtrar pesquisas e documentos sobre artigos que citam desvios em aplicações do *framework* Scrum em projetos de desenvolvimento de *software*. Foi também usado termos como *dark, scrumbut, anti-patterns* como critério de busca na pesquisa,

pois, segundo Eloranta, Koskimies e Mikkonen (2016), são termos comuns na literatura para descrever desvios e adaptações na aplicação do *framework* Scrum. Uma delimitação de 4 anos foi aplicada para restringir os resultados às publicações mais recentes que traduzissem a situação de projetos e equipes que utilizam o *framework*. No dia 15 de Agosto de 2020 foram encontrados 15 publicações na ferramenta de pesquisas de artigos e estudos acadêmicos denominada *Scopus*. Entretanto, algumas referências usadas no presente estudo passaram da data limite estipulada de 4 anos para os artigos, por serem consideradas boas e importantes referências bibliográficas. Nas próximas subseções são apresentados dois artigos pelo Autor da presente pesquisa como forma de comparativo para com o atual estudo.

2.3.2 *Exploring ScrumBut - An Empirical Study of Scrum Anti-patterns*

O artigo de Eloranta, Koskimies e Mikkonen (2016), apresenta conceitos do *framework* de desenvolvimento de Software ágil Scrum. A pesquisa levanta questionamentos sobre empresas e equipes de desenvolvimento de software que subvertem alguns desses conceitos apresentados pela ferramenta Scrum, levando os autores a questionarem sobre o impacto dessas mudanças ou alterações nos processos. A pesquisa introduz seu objetivo de entender algumas mudanças recorrentes nos processos da metodologia, os impactos e qual área do *framework* Scrum é afetada por tal mudança.

Os autores denominam os desvios nas áreas da metodologia como antipadrões. Para melhor entender e conhecer esses antipadrões a pesquisa realizou uma pesquisa qualitativa com 18 representantes de times de desenvolvimento de software, de mais variados papéis dentro desses times, como: Desenvolvedores, *Product Owners*, *Testers*, Arquitetos de software, etc. Esses 18 representantes eram de 11 diferentes empresas e todas essas empresas estavam situadas na Finlândia.

O conteúdo da pesquisa aplicada foi sobre todas as áreas do Scrum, para poder categorizar os achados em quais princípios do Scrum estavam atrelados. Após aplicada a pesquisa, e analisar seus resultados, os autores da pesquisa puderam identificar 14 antipadrões na qual eram compostos pelo contexto do desvio, quais princípios básicos do Scrum foram infringidos, possíveis consequências do desvio encontrado e recomendações sobre como evitar o desvio previamente catalogado.

Apesar das limitações da aplicação da pesquisa, pelo fato de ter sido apenas aplicada na Finlândia, e com poucas equipes de desenvolvimento, trata-se de um incentivo para se identificar alguns erros ou práticas que se pode evitar, ou talvez, contornar, para a indústria de

desenvolvimento de software aprimorar seus processos. O artigo é uma das bases para a presente pesquisa pelo fato de já ter identificados algumas antipráticas do Scrum. Na presente pesquisa, para a construção do questionário, baseou-se no roteiro usado pelos autores Eloranta, Koskimies e Mikkonen (2016), porém aplicado em equipes brasileiras de desenvolvimento validando se o mesmo ocorre no contexto brasileiro.

2.3.3 *The Dark Side of Agile Software Development*

O artigo de Janes e Succi (2012), busca expressar a necessidade da indústria, e da comunidade acadêmica, no que se deve a buscar mais bases quantitativas sobre os ganhos das metodologias ágeis no desenvolvimento de software. Para desmistificar o uso das metodologias ágeis e entender suas reais aplicações e sua melhor forma de executar.

Ao decorrer da pesquisa os autores apontam algumas falhas de gurus das metodologias ágeis por valorizarem tanto as metodologias ágeis sem antes mostrarem dados quantitativos sobre seus reais ganhos. O artigo e seus questionamentos são muito convenientes, ainda hoje, pois na indústria de desenvolvimento de software as tecnologias e novos processos surgem de maneira muito rápida, e deve-se questionar sobre a real utilização e aplicabilidade dessas novas ferramentas. O que não deve ser diferente para as metodologias ágeis.

Esse artigo é usado como referência na presente pesquisa pelo fato dele possuir embasamento e questionamentos relativos a entender melhor as ferramentas, contemplando seus ganhos, seus pontos negativos, e deixar de lado a euforia de quando se começa a usar algo novo, e sim, fazer uma análise crítica dos mesmos.

2.3.4 Comparativo entre trabalhos relacionados

No Quadro 3 encontra-se um comparativo entre os trabalhos relacionados desta pesquisa. Os objetivos de Eloranta, Koskimies e Kikkonen (2016), são de obter dados empíricos sobre desvios das práticas recomendadas do Scrum em empresas que o usa para desenvolver software e classificar esses desvios como antipadrões. Já os objetivos de Janes e Succi (2012), diferentemente de uma abordagem prática, seguindo em linhas mais teórica é de alertar a comunidade sobre a necessidade de validar as reais vantagens e desvantagens sobre os frameworks de metodologias ágeis atuais, incluindo o Scrum.

Assim como diferentes objetivos, ambas as pesquisas também diferem em suas contribuições, onde Eloranta, Koskimies e Kikkonen (2016), contribuem mapeando 14

antipadrões para o uso da metodologia Scrum. Esses antipadrões identificados são compostos pelo contexto do desvio, quais princípios básicos do Scrum foram infringidos, possíveis consequências do desvio encontrado e recomendações sobre como evitar o desvio previamente catalogado. Enquanto Janes e Succi (2012), contribuem com dados apresentados no *Gartner's Hype Cycle*, exemplificando o pico de euforia que acontece normalmente na indústria de desenvolvimento de software em relação a novas ferramentas na área. Alertando sobre a validação antes do uso dessas novas ferramentas tendências da área.

Apesar de ambas pesquisas contribuírem de forma significativa, elas possuem suas limitações, no caso de Eloranta, Koskimies e Kikkonen (2016), a pesquisa foi feita apenas com 18 representantes de equipes de desenvolvimento da Finlândia, assim limitando muito o resultado. E na pesquisa de Janes e Succi (2012), os autores não possuem nenhuma aplicação prática de como validar uma nova tendência no mercado de desenvolvimento de software, apesar dos questionamentos levantados.

Quadro 3 – Resumo dos trabalhos relacionados

Trabalho	*Exploring scrumbut - an empirical study of scrum anti-patterns*	*The dark side of agile software development*
Autores e ano	Eloranta, Koskimies e Mikkonen (2016)	Janes e Succi (2012)
Objetivo	Mapear desvios e classificá-los como antipadrões	Validar reais vantagens e desvantagens sobre ferramentas da área de desenvolvimento de software
Contribuições	Mapeados 14 antipadrões para o uso da metodologia Scrum	Alertando a comunidade, baseado em dados, sobre o pico de euforia na indústria com novas ferramentas
Limitações	Apenas 18 participantes entrevistados	Nenhuma aplicação prática

Fonte: Elaborado pelo autor.

3 MATERIAIS E MÉTODOS

A presente pesquisa possui caráter exploratório de natureza aplicada, devido ao seu interesse na aplicação prática. (MARCONI; LAKATOS, 2012). Esta pesquisa classifica-se como qualitativa, pois busca coletar experiências, por meio de questionário, de indivíduos envolvidos no desenvolvimento de software utilizando a metodologia de desenvolvimento Scrum, para alcançar seus objetivos. A amostra da pesquisa conta com desenvolvedores e envolvidos no processo de desenvolvimento, que utilizam Scrum. Todos os integrantes da amostra da presente pesquisa são empregados da filial de uma empresa multinacional de

desenvolvimento de software situada no Vale do Rio dos Sinos, Rio Grande do Sul. Para questões de preservação, a mesma é mantida anônima durante todo presente estudo.

A metodologia que a pesquisa se guia é estudo do caso, contando com questionário para a coleta de dados. Já a análise dos dados coletados é feita utilizando a instrumento de análise de afinidade, onde cada informação coletada é encapsulada junto com alguma área da Scrum onde se relaciona. O questionário[1] criado por Eloranta, Koskimies e Mikkonen (2016), para o estudo denominado por eles de *Exploring ScrumBut - An empirical study of Scrum anti-patterns*, é usado como base para a construção do questionário para a presente pesquisa. São adaptadas as perguntas, e a estrutura, visando tornar o questionário desta pesquisa mais enxuto que de Eloranta *et al.* (2016).

3.1 Coleta e Análise de Dados

Como instrumento de pesquisa para coleta de dados, aplicou-se o questionário através de formulário eletrônico no *Microsoft Forms*. Para a construção do formulário foram pensadas em 5 seções que tratam de desenvolvimento de software utilizando o *framework* ágil Scrum, sendo as seções: conhecendo o contexto do perfil do respondente, organização do time, cerimônias do time, padrões e artefatos do time, e por último, uma seção referente aos *backlog items* do time. As mesmas seções não foram refletidas no questionário enviado aos respondentes para evitar qualquer viés que pudesse vir a acarretar ao questionário. Tomou-se cuidado para que as questões fossem claras, objetivas e endereçassem os objetivos da presente pesquisa.

O roteiro do questionário foi elaborado com base na fundamentação teórica deste trabalho e encontra-se na íntegra disponível plataforma de versionamento *GitHub*[2]. Garantindo assim, a disponibilidade e livre acesso para uma possível futura reprodutibilidade da presente pesquisa. Durante o pré-teste algumas perguntas que se apresentavam em formato de múltipla escolha foram reformatadas para escala de apreciação, de forma que fosse possível extrair uma informação mais relevante. Além disso, algumas novas questões opcionais foram incluídas ao questionário, para coletar informações mais precisas sobre possíveis respostas. Posteriormente, o questionário foi enviado á todos os 175 membros da equipe em questão no dia 07/10/2020 via aplicativo de mensagens instantânea interno da equipe chamado *Slack.* O questionário ficou disponível para durante 7 dias e obteve um total de 50 respostas.

[1] Questionário disponível em: http://www.elisanet.fi/weellu/scrum/interview_questions.html. Acesso em: 23 nov. 2019.
[2] Questionário encontra-se disponível em: https://github.com/luanlima1997/ScrumDeviations. Acesso em: 05 out. 2020.

Das 50 respostas coletadas pelo questionário, todos os participantes responderam que trabalham usando metodologia ágil em seu projeto. Assim como todos participantes confirmaram que trabalham com o *framework* de metodologia ágil Scrum, portanto, todas as respostas sendo válidas e úteis para a presente pesquisa. As repostas estão todas disponíveis na íntegra na plataforma *GitHub*, com todas as informações relativas aos participantes estando de forma anônima.

3.1.1 Questionário

Para a criação do questionário pensou-se em alguns grupos de perguntas, seções, para a melhor organização e embasamento teórico do questionário. Na plataforma *GitHub* também se encontra as perguntas descritas para cada seção. As seções do questionário baseiam-se nas seguintes fundamentações, *auxiliar na identificação do perfil dos respondentes*: Segundo Ebrahim, Ahmed e Taha (2019), Desenvolvimento de software pode envolver time de pessoas trabalhando de forma integrada de diferentes experiências, idades, vivências, entre outros fatores, todos com a intenção de atingir um objetivo em comum. Assim, é importante entender o contexto do individuo respondente, entendendo seu papel, este podendo ser mais de um eventualmente, desempenhado no time, bem como treinamentos feitos envolvendo o *framework* Scrum e a metodologia ágil.

No Scrum é fundamental que as equipes tenham significado multifuncional e que os membros da equipe possuem todo o conhecimento necessário para entregar um software funcional. Além disso, a equipe deve ser auto organizada de forma que cada membro possa escolher no que trabalhar e não haja ninguém atribuindo tarefas. (ELORANTA; KOSKIMIES; MIKKONEN, 2016). No questionário existe uma seção denominada *organização do time.*

Segundo Cho (2008), existem algumas cerimônias no processo Scrum incluindo o *Daily Scrum Meeting*, *Sprint Review Meeting* e a Reunião de planejamento do *sprint*. De forma geral todas as cerimônias do Scrum existem para facilitar as comunicações, identificar e remover impedimentos ao desenvolvimento, destacar e promover a tomada de decisão rápida e melhor transparência (visibilidade). Sempre seguindo e mirando nos pilares do Scrum que tem três pilares subjacentes a todas suas implementações: transparência (visibilidade), inspeção e adaptação. Sabendo da importância das cerimônias no *framework*, cria-se a seção de *cerimônias do time* no questionário.

No Scrum, cada membro da equipe é responsável pela completude de sua solução. Contudo, há um grande número de opções de metodologias para verificar completude. Isso

significa que um membro da equipe pode usar qualquer método de verificação. Sabendo disso, existem artefatos como o *Definition of Done*, onde todos os membros do time, juntos, definem o que é considerado pronto para o seu projeto. Assim, é necessário o time definir alguns padrões para seguir com o seu trabalho de forma uniforme. (POHL; HOL, 2015). Para o questionário, a seção padrões e artefatos do time contém questionamentos sobre essas áreas do *framework*.

Os *product backlogs*, ou *backlog items*, são funcionalidades divididas em listas, para assim, todo o time estar ciente dos próximos passos do projeto, e ao longo do caminho alterarem a lista com as prioridades do projeto, e a cada iteração diminuir essa lista. É de suma importância esses itens estarem sempre ordenados, organizados e visíveis para todos os membros do projeto. (MARTINS; DE MENEZES, 2016). Assim, sabendo-se a importância da organização do *backlog* de um projeto de software, que se guia pelo *framework* Scrum, cria-se a seção de *backlog items* no questionário, avaliando assim, esses quesitos na pesquisa.

4. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção são apresentados os resultados da pesquisa e uma análise com base nas referências encontradas na literatura, relacionando-as com os dados coletados, que representam o que de fato é a realidade da equipe em questão referente às práticas de Scrum no setor.

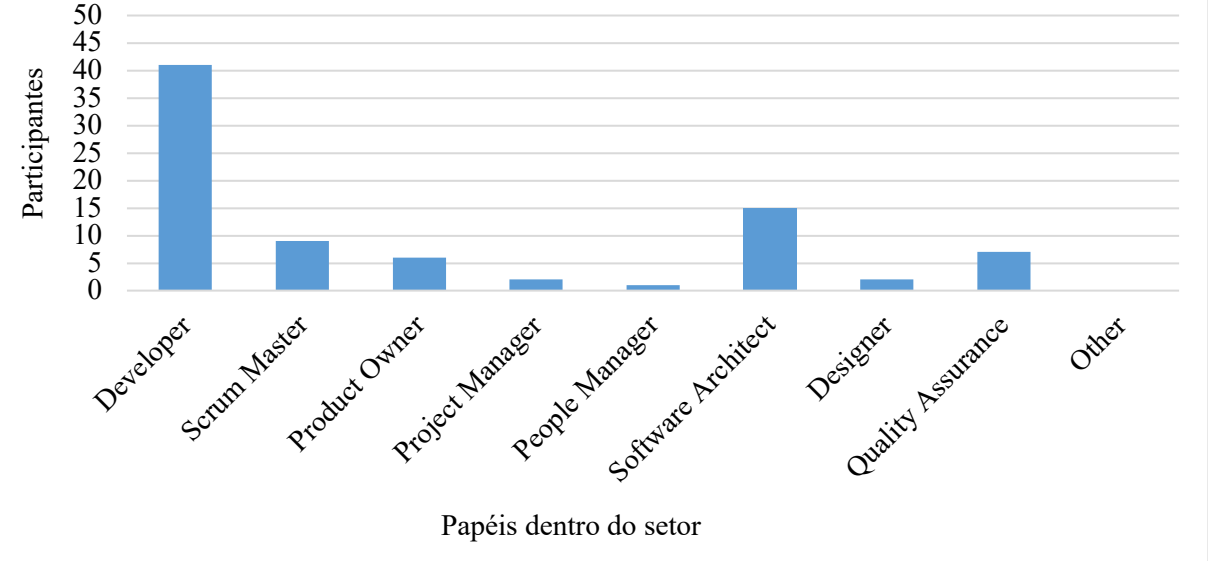
4.1 Perfil dos participantes

A parte inicial do questionário designava-se para tratar o perfil do participante, identificando o papel que o indivíduo exerce dentro de seu time e se essa pessoa já participou de algum treinamento oficial de Scrum. Sobre os papéis desempenhados pelos respondentes, 41 dos 50 participantes da pesquisa, exercem o papel de desenvolvedor dentro do seu time, como observa-se na Figura 1, podendo este estar acumulado com mais algum papel, ou não. Entretanto, um dado que chama atenção é quando o indivíduo exerce um papel diferente do que desenvolvedor dentro de seu projeto, esse papel, em quase todos os casos, é acumulado junto com a tarefa de desenvolvedor, esse é o caso de 25 respostas, onde junto com o papel de desenvolvedor a pessoa também atua em seu projeto como arquiteto de software, *scrum master, designer,* analista de qualidade, entre outros.

O Guia do Scrum, Schwaber e Sutherland (2013), menciona que não há problema de um indivíduo exercer mais de uma função dentro do seu time, desde que os papéis não sejam antagônicos entre eles, como no caso de *Product Owner* e *Scrum Master*, onde um papel pode

influenciar o outro. Devem ser pessoas diferentes assumindo esses papéis para evitar algum conflito de interesse em decisões do projeto. Entretanto, identifica-se 3 respostas diferentes onde a mesma pessoa exerce pelo menos essas duas tarefas dentro do time ao mesmo tempo. Logo, sugere-se que pessoas diferentes devam desempenhar essas funções dentro dos times.

Figura 2 - Papéis desempenhados pelos respondentes



Fonte: Elaborado pelo autor.

Já na pergunta 4 do questionário referente sobre o grau de instrução dos participantes, indagando especificamente se já participou de algum treinamento de Scrum, observa-se que apenas 60% dos respondentes, ou seja, 30 pessoas, confirma que já participou de algum treinamento oficial de Scrum. Olhando mais afundo nesses casos, onde o respondente não participou de algum treinamento de Scrum, percebe-se que existem indivíduos que mesmo sem alguma instrução oficial desempenham algum papel relevante no projeto, como *Product Owner*, duas respostas se encaixam nesse caso.

Segundo o Guia do Scrum, Schwaber e Sutherland (2013), o *Product Owner* é responsável por maximizar o valor do produto resultante do trabalho da equipe de desenvolvimento. A forma como isso é feito pode variar amplamente entre organizações, equipes e indivíduos. O *Product Owner* é a única pessoa responsável por gerenciar o *Product Backlog*. Portanto, para se ter sucesso no projeto, o *Product Owner* precisa ter uma compreensão concreta de todos os aspectos de gerenciamento do produto, visando sempre agregar ao produto.

Sugere-se que exista uma definição na empresa sobre os deveres de um *Product Owner* dentro dos projetos, pois, ainda segundo Schwaber e Sutherland (2013), a forma como os

Product Owners atuam pode variar amplamente entre diferentes organizações, equipes e indivíduos. Além da sugestão de definir deveres e expectativas das funções de que um *Product Owner* deve executar, sugere-se também que todos tenham algum treinamento oficial para todos que exercem a função, garantindo a uniformidade do trabalho, mesmo em diferentes projetos.

Finalizando a seção de conhecimento do perfil dos participantes, questiona-se, em resposta no formato de texto livre, sobre a experiência em geral com o *framework*. Por esta ser uma questão opcional, obteve-se 30 respostas. Em todas as respostas os participantes confirmam que, sim, a utilização do *framework* como seu formato de trabalho agrega valor a seus projetos e times, pode-se observar alguns exemplos de respostas no Quadro 4.

Quadro 4 –Respostas positivas dos participantes

Participante	**Resposta**
Anônimo 9	Acredito que o Scrum adiciona muita visibilidade e muito poder aos "ranks" inferiores no processo de desenvolvimento, além de dar uma expectativa muito mais realista sobre a entrega aos POs e clientes.
Anônimo 12	Ele dá mais clareza sobre o que está sendo trabalhado, ao mesmo tempo que permite que as pessoas trabalhem como caber (sozinho, programação em pares, convocação de reuniões para lidar com alguns detalhes, etc)
Anônimo 29	Acredito que o framework Scrum ajuda muito no contexto do projeto para dar visibilidade das entregas e permitir que a equipe tenha autonomia através da flexibilidade do framework. De uma forma muito simples, podemos ver se o backlog está atrasado ou para não tomar uma decisão sobre a equipe (por exemplo, alterar o tamanho da equipe), a equipe pode organizar o trabalho de forma que as pessoas aprendam juntas e usem o seu melhor habilidades para entregar o escopo. Além disso, a flexibilidade de escopo permite atender da melhor forma as necessidades dos clientes, priorizando ou mesmo alterando as entregas das equipes.

Fonte: Elaborado pelo autor.

Apesar de todas as respostas serem positivas, avalia-se alguns desvios ou não adaptação de contratos em projetos que utilizam o Scrum, segundo alguns participantes, como pode-se constatar no Quadro 5. Deixando um sentimento entre os participantes de que utilizam Scrum apenas internamente na empresa, e não com os clientes. Como sugestão de melhoria para os contratos dos projetos que utilizam o framework, Opelt *et al.* (2013) sugerem um modelo onde o valor do projeto é fechado, mas o escopo aberto, e combina-se com os clientes preços e tamanhos de cada item, ou funcionalidade, a ser desenvolvido no projeto, e a cada *sprint* e

incremento do projeto, desconta-se do preço total os valores dos itens entregues, e prioriza-se junto com os *stakeholders* do projeto os próximos itens a serem desenvolvidos, repetindo esses movimento a cada iteração do projeto. Essa sugestão de modelos de contrato funciona caso o cliente entenda o valor de projetos ágeis e esteja disposto a estar presente e renegociar escopo e priorização a cada *sprint* do projeto.

Quadro 5 –Respostas com ressalvas dos participantes

Participante	**Resposta**
Anônimo 3	SCRUM é bom, no entanto, sinto que às vezes não o aplicamos corretamente, e talvez nem possamos aplicá-lo, pois "temos que entregar tudo" de qualquer maneira. Às vezes eu sinto que é apenas *waterfall* com cerimônias extras no topo
Anônimo 8	Sinto que ainda não sabemos quanta flexibilidade podemos dar ao cliente sem afetar o cronograma original. No papel, funciona perfeitamente, a história aponta para remover / adicionar recursos / alterações, mas na prática, onde temos que negociar com o cliente, não vejo funcionando muito bem. Normalmente, o cliente só deseja adicionar mais e mais.
Anônimo 11	O próprio Scrum é (e deve ser) incompleto, da minha perspectiva. Nós o usamos como referência, mas sempre temos que adicionar mais coisas para que funcione para nós. Definitivamente, há valor em retrospectiva, diária, revisão e planejamento, por exemplo. Por outro lado, a forma como a negociação é feita na minha empresa não cabe no "ágil", então sempre há essa contradição.

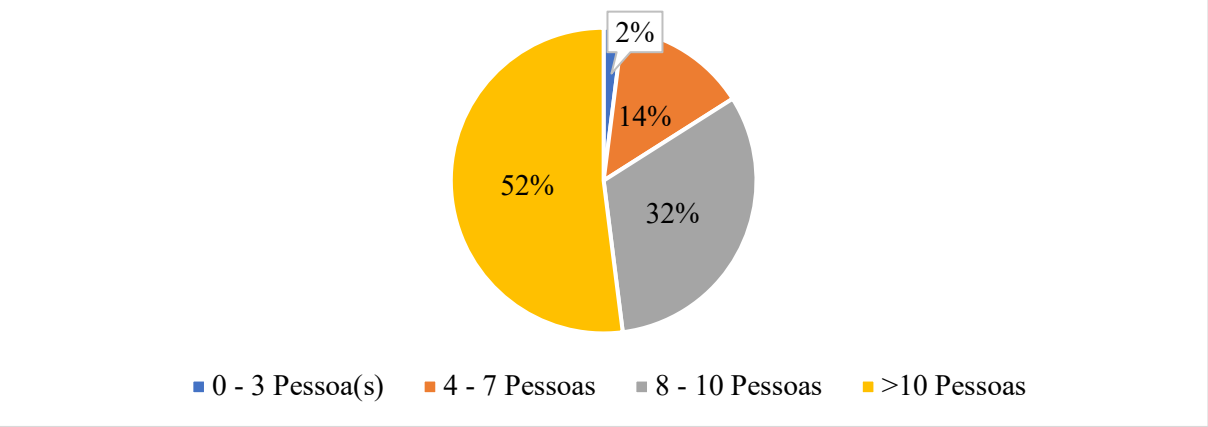
Fonte: Elaborado pelo autor.

4.2 Organização do time

Sobre a seção do questionário referente a organização do time que o participante está inserido, observa-se, primeiramente, um dado chamando atenção sobre o tamanho dos times, como pode-se identificar na Figura 3, mais de 50% dos times são compostos por mais de 10 pessoas. Segundo Mundra, Misra e Dhawale (2013), apesar de existirem diferentes interpretações sobre o tamanho que deve ser um time de Scrum, todas teorias concordam de que o time deve ser pequeno ao ponto de ter de 3 a 9 pessoas envolvidas, garantindo um rápido fluxo de comunicações entre os membros, além de facilitar encontros do time, e o autogerenciamento do mesmo.

Para times maiores do que 10 pessoas o gerenciamento e a organização começam a ficar conturbado. Logo, sugere-se que se quebre em times menores para os projetos, para isso, pode-se usar alguns conceitos como *Scrum of Scrums* (SoS), ainda segundo Mundra, Misra e Dhawale (2013), SoS são várias equipes de Scrum trabalhando e coordenando-se no mesmo produto ou para o mesmo objetivo. A palavra *mesmo* é a chave aqui, pois se os produtos forem diferentes, precisa de equipes Scrum diferentes ao mesmo tempo e nenhuma coordenação é necessária. Pode-se também usar SoS se a equipe de produto for em vários locais e cada local tendo equipes Scrum totalmente funcionais. Isso torna a execução de produtos grandes gerenciável. Além disso, traz alguns novos desafios como a atribuição de tarefas, é uma tarefa relevante no desenvolvimento de software e é mais importante uma vez que o desenvolvimento de software é distribuído.

Figura 3 - Tamanho dos times

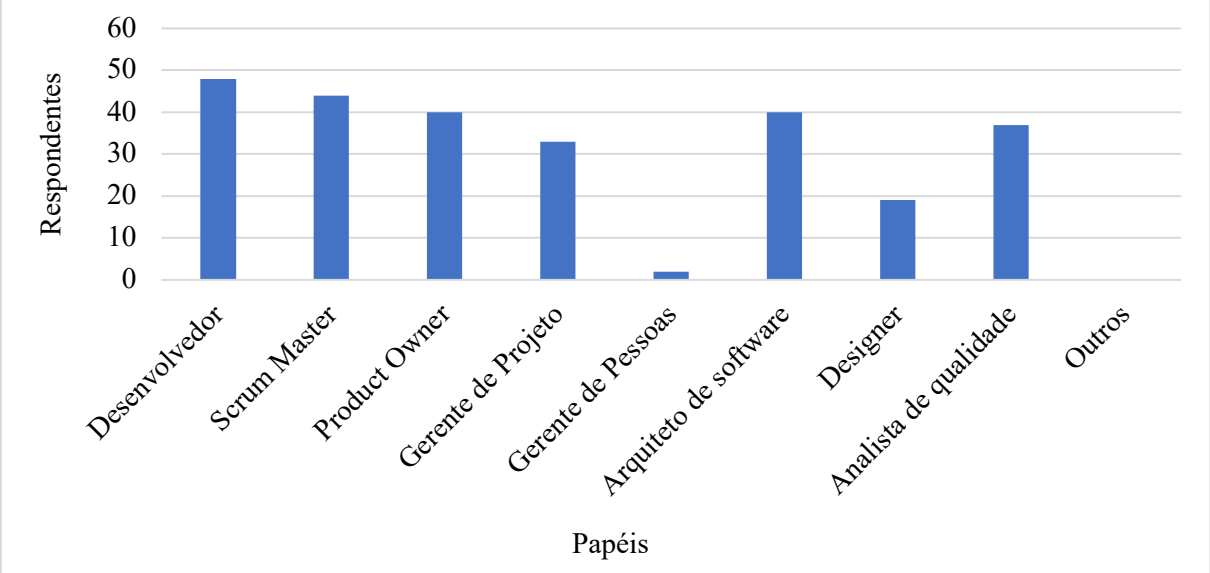


Fonte: Elaborado pelo autor.

Na mesma seção do questionário indaga-se sobre quais são os papéis que existem dentro dos times, segundo Schwaber e Sutherland (2013), o time Scrum consiste em um *Product Owner*, o Time de Desenvolvimento e um *Scrum Master*. Times Scrum são auto organizáveis e multifuncionais. Times auto organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do time. Times multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. O modelo de time no Scrum é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade. O Time Scrum demonstra-se estar aumentando sua efetividade para todos os usos anteriormente citados, e qualquer trabalho complexo. Conquanto, para um time ser multidisciplinar, em alguns casos, precisa-se de alguns novos papéis, como *designer*, analista de qualidade, entre outros.

Observa-se na Figura 4, no setor onde acontece o presente estudo de caso, existe uma pluralidade de papéis dentro dos times de Scrum, no qual não impacta ou fere algum princípio do *framework*. Pode-se observar uma grande presença de *Project Manager* dentro dos projetos do setor em questão, como citado por Mundra, Misra e Dhawale (2013), esta não é uma função definida no Scrum. No entanto, por sua experiência, os gerentes de projeto são de grande ajuda. A função do Gerente de Projeto deve ser prevenir a equipe de perturbações externas. Além disso, o Gerente de Projeto está ciente do impacto do projeto na organização e de outros projetos dentro da organização que provavelmente irá impactar este projeto. Essa não é frequentemente a perspectiva do *Scrum Master* nem do *Product Owner*. Gerente de projeto, a partir de seu ponto de vista pode apontar a equipe dos possíveis problemas. Entretanto, o Gerente de Projeto não deve interferir nas estimativas da equipe ou estilo de trabalho ou qualquer outro aspecto do desenvolvimento do projeto. No Scrum, se usado, a função do gerente de projeto é de apoiar e não uma função com autoridade.

Figura 4 - Papéis dentro do time

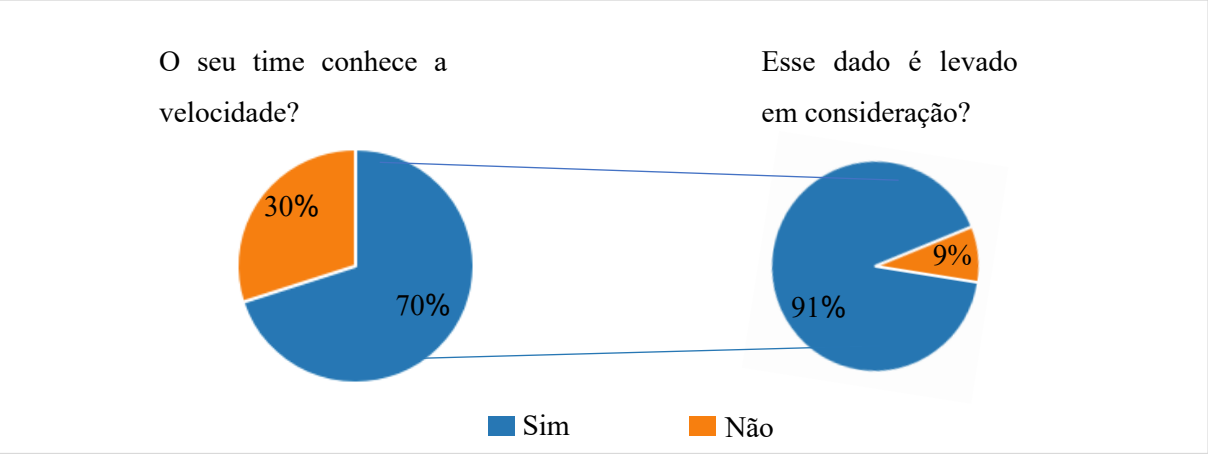


Fonte: Elaborado pelo autor.

Seguindo nas perguntas do questionário que se enquadram dentro da seção de organização do time, especificamente a questão referente ao conhecimento do time sobre a sua velocidade de entrega e se esse dado é levado em consideração para decidir a carga de trabalho de uma *sprint* do time. Obteve-se uma resposta relativamente positiva para o questionamento, onde pelo menos 70% de confirmação de que sim, esse dado é levado em consideração ao decidir o escopo de uma *sprint* que se inicia nos times, como pode-se identificar na Figuras 5. Esse dado é muito importantes ser levado sempre em consideração e sempre ser constantemente

atualizados e visíveis para o time todo. Segundo Schwaber e Sutherland (2013), essas informações são a prova de quanto um time consegue entregar em um período de tempo, sendo esse período de tempo, a duração das *sprint*s do time.

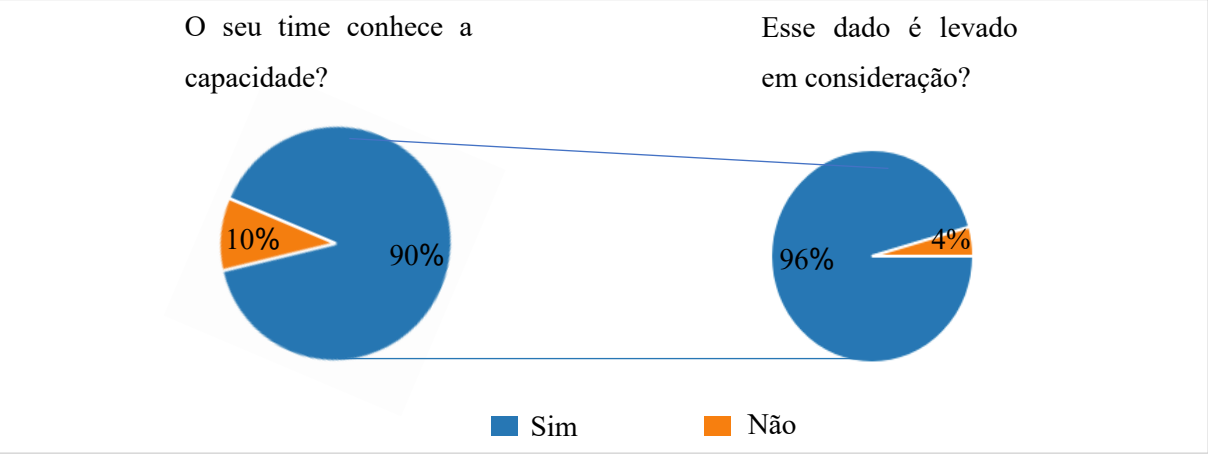
Figura 5 - Velocidade do time



Fonte: Elaborado pelo autor.

Ainda sobre a organização do time, questiona-se sobre o conhecimento do time referente a sua capacidade de entrega, outro dado importante que se enquadra na citação anterior de Schwaber e Sutherland (2013), sobre o autoconhecimento equipe e a importância do mesmo. Obteve-se uma reposta muito positiva, como pode-se ver na Figura 6, onde 90% dos participantes confirmam que sim, os dados sobre a capacidade do time são levados em consideração ao ser escolhida a carga de trabalho do time. Somando-se as repostas refletidas na Figura 5 e na Figura 6, pode-se dizer que as equipes dos participantes possuem maturidade sobre suas capacidades de entrega, resultando assim, em um bom autoconhecimento da equipe.

Figura 6 - Capacidade do time



Fonte: Elaborado pelo autor.

Seguindo nas perguntas do questionário referentes a seção de organização do time, pode-se observar nas respostas referentes ao poder de decisão do time relativo a quem determina quais serão os próximos itens a serem desenvolvidos pela equipe na *sprint*, no qual, segundo o Guia do Scrum, Schwaber e Sutherland (2013), afirmam que o *Product Owner* deve ter o controle dos itens a serem desenvolvidos. Já as respostas não fogem dessa expectativa, 80% dos respondentes do questionário, confirmam que essa atribuição é do *Product Owner* em seus projetos. Como esta é uma pergunta onde pode-se ter múltiplas respostas, e, em 20% das respostas, confirma-se que o Arquiteto de Software, uma forma de líder técnico dos times, também possuem essa incumbência. No qual, ainda segundo Schwaber e Sutherland (2013), não deve ser um problema, desde que o arquiteto esteja escolhendo itens técnicos para o time, e não ultrapassando os seus limites de tarefas com os de os *Product Owners*, no qual devem limitar-se ao gerenciamento do escopo de trabalho, sendo este, funcional e não técnico.

Ainda sobre ao poder de decisão do time, agora referente a como o time divide as tarefas que serão desenvolvidas ao decorrer de uma *sprint*, obteve-se a reposta de 88% dos casos de que o time de desenvolvimento tem a autonomia de realizar essa divisão do trabalho a ser feito. Esse dado reflete que os times se auto organizam, isso é positivo, segundo Martins e Menezes (2016), este é bom indicativo pois demonstra sinais de que a equipe define a melhor forma de desenvolver seu próprio trabalho. Demonstrando assim ser independente e autogerida, pontos importantes que também Schwaber e Sutherland (2013) destacam como fundamental para o sucesso de uma equipe Scrum.

Seguindo na pergunta que avalia a estabilidade da equipe, questionando sobre a estrutura de membros do time mudar com frequência de três *sprints* ou menos, 78% das respostas confirmam esta estabilidade, Sutherland, Harrison e Riddle (2014), apontam que equipes estáveis tendem a conhecer sua capacidade, o que permite que o time tenha previsibilidade de suas ações. Apesar de que, algumas mudanças no time irão acontecer de qualquer forma, como saída de algum membro, férias, folgas, etc. Quanto mais estável o time, mais previsível e gerenciável o seu trabalho se torna.

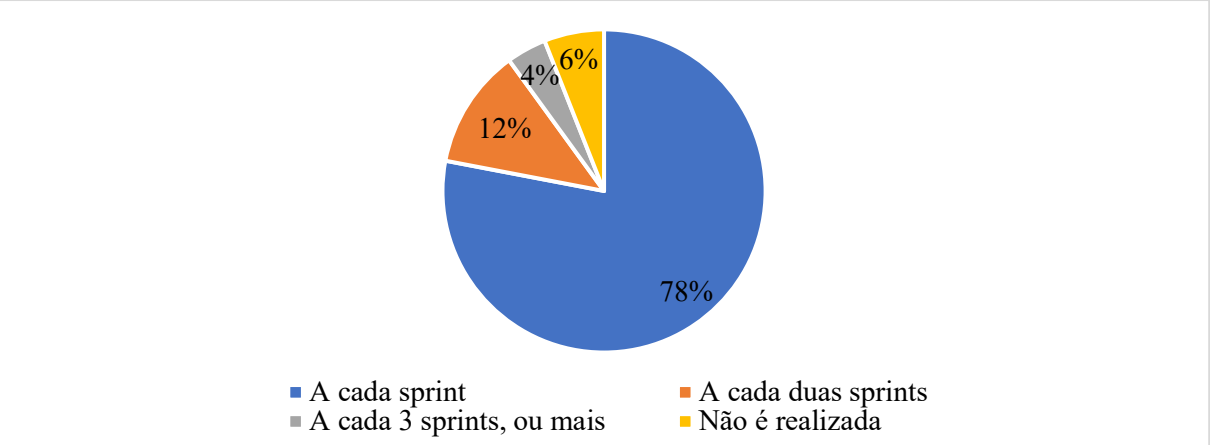
4.3 Cerimônias do time

Na seção do questionário designada para avaliar as cerimônias que o *framework* Scrum propõe, pode-se observar que a equipe segue as recomendações do Guia do Scrum, Schwaber e Sutherland (2013), onde todas as repostas confirmam que as *sprints* não possuem duração maior do que 3 semanas, sendo isso um ótimo sinal, pois o guia sugere que as *sprints* não devem

passar de 4 semanas. Um outro ponto destacando a maturidade da equipe sobre o uso da ferramenta é sobre as cerimônias que as equipes fazem, onde mais de 87% das respostas confirmam que todas cerimônias que o Guia do Scrum, Schwaber e Sutherland (2013), sugere, são feitas, sendo elas: *Sprint Planning*, *Daily Meetings*, *Sprint Review* e *Sprint Retrospective*.

Seguindo nas repostas relacionadas as cerimônias que as equipes fazem, especificamente sobre a frequência de que os times realizam a cerimônia de retrospectiva da *sprint*, onde, encontra-se uma confirmação de 78% dos casos de que a cerimônia acontece a cada *sprint*, como observa-se na Figura 7. Conforme sugerido por Schwaber e Sutherland (2013), na qual dizem que é a oportunidade para o time Scrum inspecionar a si próprio e criar um plano para melhorias a serem aplicadas na próxima *sprint*, garantindo o melhoramento contínuo da equipe e dos membros. Sabendo disso, sugere-se que os outros 22% das equipes dos respondentes que não realizam a retrospectiva a cada *sprint* tenham atenção e realizem a cerimônia com a frequência sugerida, de cada *sprint*, colhendo assim, os frutos da autoanálise constante da equipe e amadurecendo constantemente seus processos.

Figura 7 - Frequência das Retrospectivas de *sprint*



Fonte: Elaborado pelo autor.

Em outro dado importante a ser destacado é sobre o esclarecimento de todo o time sobre o objetivo da *sprint*, ou *Sprint Goal*, no qual segundo Arafeen e Bose (2009), citam que ter um objetivo definido para a *sprint* é imprescindível para o sucesso da mesma, se não existir um objetivo definido e entendido por todos os membros, não existe como avaliar se a *sprint* foi concluída com sucesso ou não. Observa-se na nas respostas que 22% dos respondentes não possuem claro para si o objetivo de suas *sprints*, e, como citado por Arafeen e Bose (2009), o principal motivo por trás do scrum é que cada membro da equipe deve trabalhar de forma independente, mas com o mesmo objetivo, como uma equipe esportiva. Recomenda-se

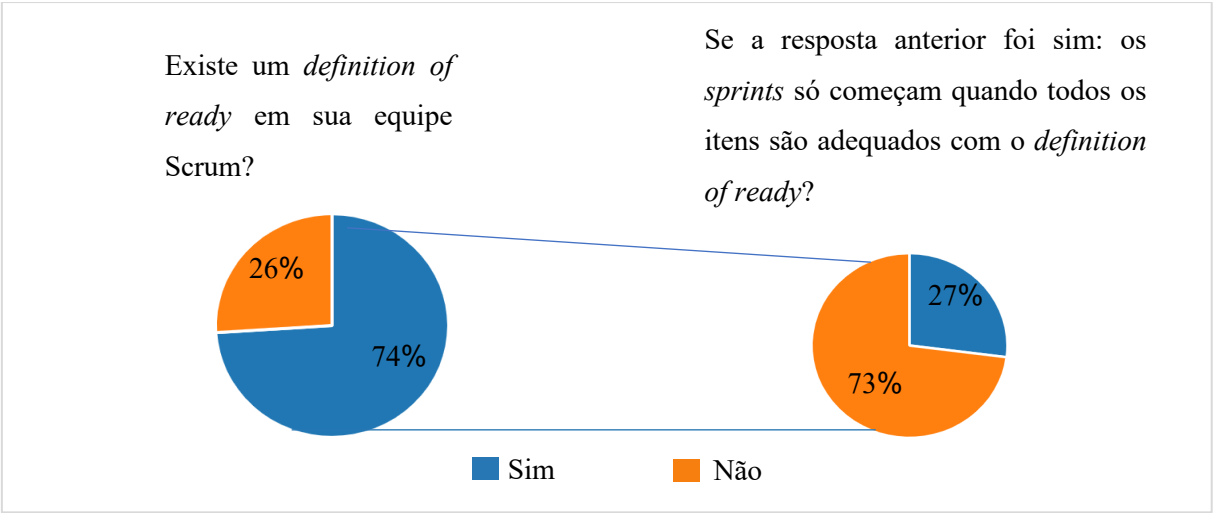
fortemente que o objetivo das *sprints* esteja clarificado e relembrado em todas as reuniões diárias das equipes.

4.4 Padrões e artefatos do time

O Guia do Scrum, Schwaber e Sutherland (2013), cita que no Scrum decisões para otimizar o valor e o controle de riscos são feitos com base na percepção existente do estado dos artefatos. Na medida em que a transparência é plena, estas decisões têm uma base sólida. Na medida em que os artefatos não são completamente transparentes, estas decisões podem ser falhas, valores podem diminuir e riscos podem aumentar. Quando o item do *Backlog* do Produto ou um incremento é descrito como pronto, todos devem entender o que o pronto significa. Embora, isso varie significativamente de um extremo ao outro para cada Time Scrum, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo, assegurando a transparência. Esta é a *Definition of Done* (DoD) para o Time Scrum e é usado para assegurar quando o trabalho está completado no incremento do produto. Assim como o *Definition of Ready* (DoR) garante que os itens á serem desenvolvidos estão com todas as informações completas que o time necessita para implementá-los.

Sobre os artefatos do time observa-se na Figura 9 que grande parte dos respondentes têm definições de *Ready* em seus projetos, o que demonstra um bom sinal. Entretanto, na Figura 9 identifica-se que mesmo existindo um DoR nas equipes, ele não é seguido. O DoR deve ser seguido por quem cria os *backlog items* do time, normalmente feito pelos *Product Owners*, por consequência, sugere-se que os responsáveis pelos itens criados a serem desenvolvidos nos projetos sigam as definições, caso isso não aconteça, o time de desenvolvimento pode barrar esses itens antes de fazer a reunião de *planning* da *sprint*, garantindo a conformidade e requisitando aos responsáveis que sigam os modelos previamente estabelecidos por todos.

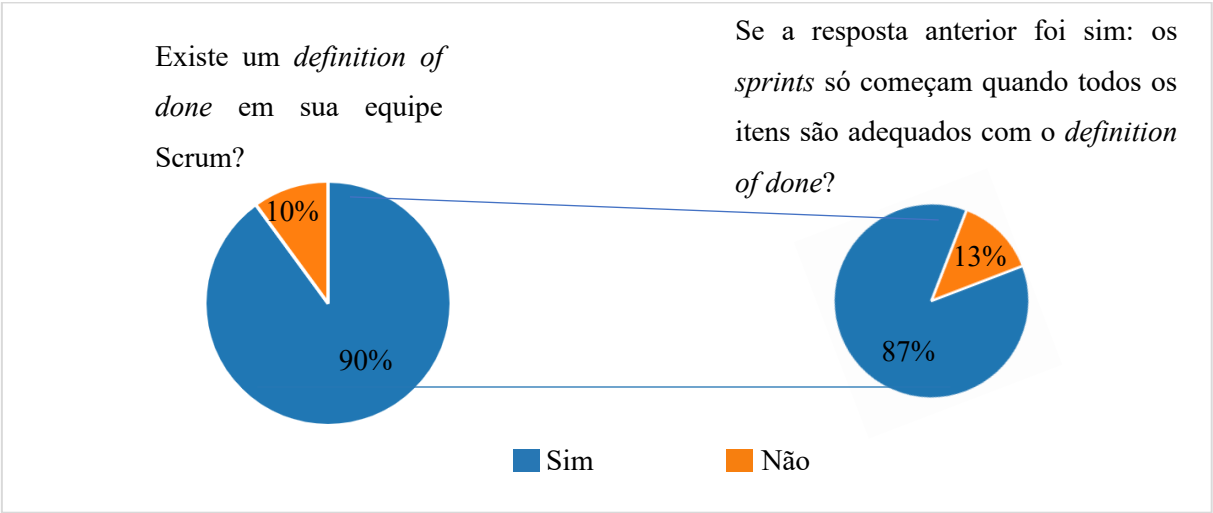
Figura 8 - Definition of Ready



Fonte: Elaborado pelo autor.

Já na Figura 10 observa-se que grande parte dos respondentes têm definições de *Done* em seus projetos, constatando-se que o DoD existe e é seguido nos projetos por mais de 87% dos participantes, diferentemente do contexto citado anteriormente na Figura 8, onde o DoR existe, mas não é seguido. Sendo assim, demonstra-se uma ótima preocupação das equipes sobre as suas definições, como citado, sendo muito importante para esclarecer entre todos envolvidos no projeto sobre o que significa pronto no contexto de trabalho que está á ser desenvolvido pela equipe.

Figura 9 - Definition of Done



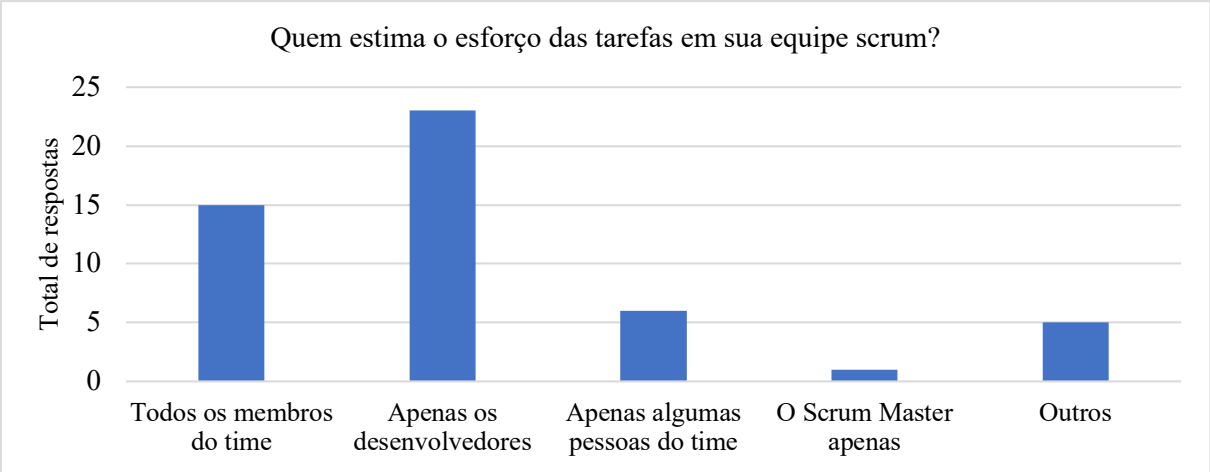
Fonte: Elaborado pelo autor.

4.4 *Backlog items*

Segundo Cho (2008), o refinamento e a manutenção do *Backlog* do Produto é a ação de adicionar detalhes, estimativas e ordem aos itens á serem desenvolvidos no projeto. Este é um processo contínuo em que o *Product Owner* e o Time de Desenvolvimento colaboram nos detalhes desses itens. Durante o refinamento do *Backlog* do Produto, os itens são analisados e revisados. O Time de Desenvolvimento decide como e quando o refinamento está *pronto*. Sabendo-se desses pontos, pode-se dizer que os times dos participantes da pesquisa possuem maturidade na manutenção dos seus itens de trabalho, como observa-se na nas respostas dos participantes, grande maioria das respostas, especificamente 88% confirmam que os itens do *backlog* são priorizados e ordenados ao longo do projeto.

Já na Figura 12 observa-se que a grande maioria dos respondentes confirma que a estimativa desses itens é feita pelo time de desenvolvimento e/ou por todos os membros do time. Segundo Sutherland, Harrison e Riddle (2014), além de sinais de autogerenciamento, quanto mais os desenvolvedores estimam as suas tarefas, mais perto do tempo gasto estará essas estimativas, refletindo o autodesenvolvimento da equipe, e assim tornando-a mais confiante e madura em seus desenvolvimentos e estimativas de projetos. Sutherland, Harrison e Riddle (2014), também citam que quanto mais baseado em históricos de *sprints* passadas a estimativa de desenvolvimento, mais propensa ao acerto a mesma está. Deixando assim esta sugestão para a equipe sempre observar seu próprio histórico ao realizar novas estimativas.

Figura 10 - Estimativas

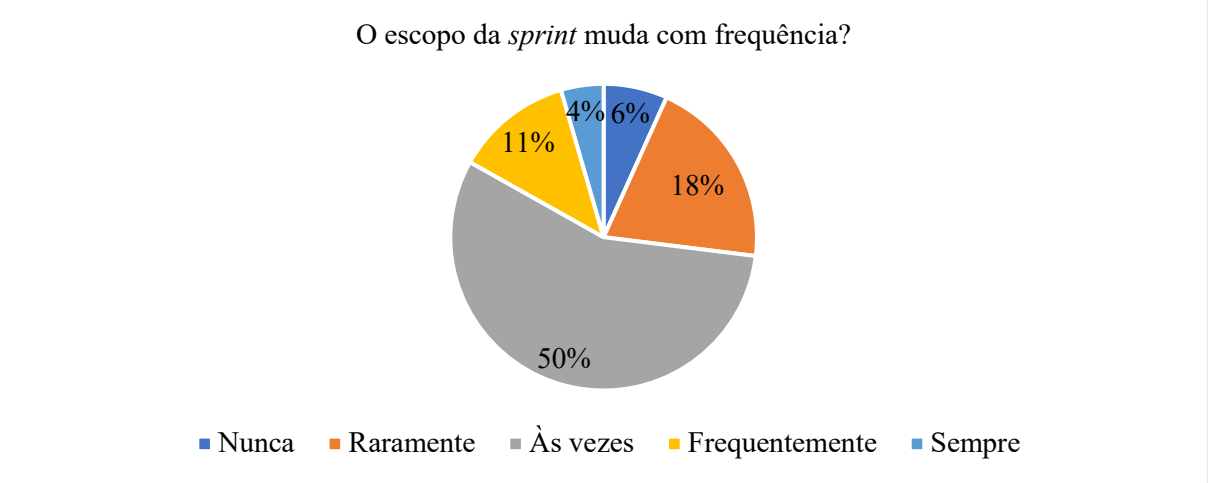


Fonte: Elaborado pelo autor.

Seguindo nas questões do questionário relativas ao gerenciamento do *backlog* do projeto, observa-se na Figura 13 que o escopo da *sprint* muda com bastante frequência, o que,

segundo Schwaber e Sutherland (2013), atrapalha e confunde o objetivo da *sprint*. Pode-se correlacionar com a informação provida na Figura 8, onde grande maioria dos participantes confirmam que o objetivo da *sprint* de seus times normalmente não é claro para os membros do time. Ainda segundo Schwaber e Sutherland (2013), durante a *sprint* não devem ser feitas mudanças que possam por em perigo o objetivo da mesma, caso isso aconteça, a *sprint* poderá ser cancelada se o objetivo se tornar obsoleto. Isto pode ocorrer se a organização mudar sua direção ou se as condições do mercado ou das tecnologias mudarem. Geralmente a *sprint* deve ser cancelada se ela não faz mais sentido às dadas circunstâncias. No entanto, devido a curta duração da *sprint*, raramente cancelamentos fazem sentido.

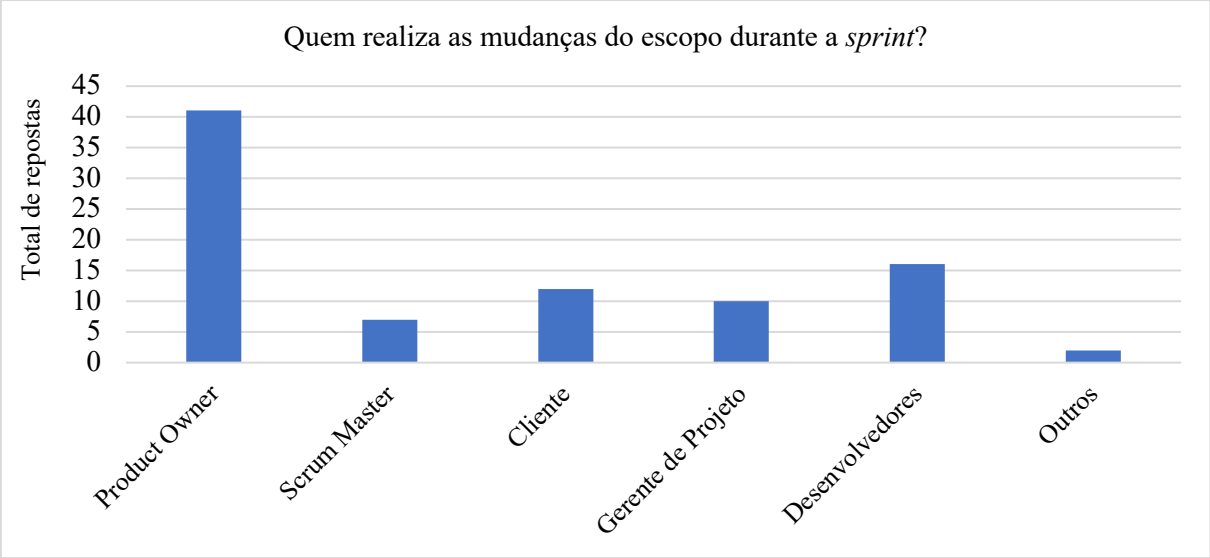
Figura 11 - Frequência de mudança no escopo da *sprint*



Fonte: Elaborado pelo autor.

Na Figura 14 questiona-se aos participantes sobre quem faz essas mudanças no escopo da *sprint*, e constata-se que majoritariamente é o *Product Owner* quem as realizam. Sugere-se que os *Product Owners* tenham cuidado e sejam orientados, mesmo que pelo time de desenvolvimento, de que, caso seja necessária a mudança na *sprint,* seja deprecado o objetivo da *sprint* e realizem uma nova reunião de *Sprint Planning,* para avaliar e estimar os novos itens que estão sendo incorporados na iteração. Novamente, sugere-se que os *Product Owners* recebam treinamentos e atualizações periódicas sobre boas práticas de gerenciamento de escopo e *backlog*, evitando assim, que situações como essa, de mudanças repentinas na *sprint*, sejam frequentes.

Figura 12 - Mudanças de escopo ao longo da *sprint*



Fonte: Elaborado pelo autor.

5. CONSIDERAÇÕES FINAIS

Este trabalho realizou uma pesquisa de estudo de caso, analisando práticas e ferramentas do *framework* Scrum usadas por um setor de desenvolvimento de projetos de software de uma empresa multinacional situada no Vale do Rio dos Sinos, Rio Grande do Sul. Afim de sugerir melhores práticas e melhorias para os integrantes desta equipe, baseando-se no Guia do Scrum, Schwaber e Sutherland (2013), e diversos outros estudos da área. Além de ser útil para o setor do estudo de caso, essas melhorias também podem ser proveitosas para outras empresas de desenvolvimento de software que aplicam o Scrum em seu trabalho.

Observa-se no capítulo de Coleta e Análise e Dados que de forma geral a empresa e seus integrantes possuem experiência com o *framework* e suas práticas, e compreendem e usufruem alguns dos benefícios das práticas sugeridas pela ferramenta. Também se constata que em diversas áreas do Scrum a equipe ainda precisa amadurecer suas práticas e metodologias de gerenciamento de projetos. Como exemplo na Figura 13 onde observa-se que o escopo das *sprints* mudam com frequência. Trazendo outros exemplos de melhorias para os projetos dos participantes, cita-se os papéis desempenhados pelos integrantes do time. Pode-se observar no capítulo 4.1 onde integrantes acumulam papéis que são antagônicos entre si, ferindo um dos princípios do Scrum, além de diminuir qualidade do trabalho, e de vida dos mesmos e aumentar estresse. Seguindo em melhorias destaca-se o tamanho dos times, onde a maioria dos projetos contam com times maiores do que as referências bibliográficas sugerem.

Indo de encontro aos objetivos desta pesquisa, pode-se concluir que existem desvios na aplicação dos processos de Scrum e também esses processos estão causando falhas das mais diferentes formas nas equipes. Como as já citadas, além de *sprints* sem objetivos claros, vide-se Figura 8, levando assim entregas de *sprints* que talvez não agreguem aos *stakeholders*. Além de falhas pode-se observar que a equipe perde algumas oportunidades de melhorias contínuas, como nota-se na Figura 7 onde por parte de alguns participantes não se realizam as cerimônias de retrospectiva da *sprint* com a frequência sugerida. Além de padrões criados nos projetos que não são seguidos posteriormente, como o exemplo do *Definition of Ready*, Figura 9, entre outros exemplos.

Como trabalhos futuros sugere-se que se aplique as melhorias sugeridas ao longo do capítulo de Análise e discussão dos resultados nos desvios apontados, e valide-se a eficácia destas sugestões. Pode-se também seguir com pesquisas mais profundas e entrevistas com usuários de casos específicos de respostas, como exemplo dos projetos que não fazem a reunião de retrospectiva de *sprint* com regularidade, ou dos participantes que não levam em consideração o *Definition of Ready* ao longo do projeto, ou o motivo dos *Product Owners* frequentemente alterarem o escopo das *sprints*, entre outros fatores que poderiam ser foco das entrevistas.

REFERÊNCIAS

ABRAHAMSSON, P.; CONBOY, K.; WANG, X. Lots done, more to do: The current state of agile systems development research. **European Journal of Information Systems**, v. 18, n. 4, p. 281–284, 2009.

ANNOSI, M. C. *et al*. Social Conduct, Learning and Innovation: An Abductive Study of the Dark Side of Agile Software Development. **Creativity and Innovation Management**, v. 25, n. 4, p. 515–535, 2016.

ARAFEEN, J.; BOSE, Saugata. Improving Software Development Using Scrum Model by Analyzing Up and Down Movements on The Sprint Burn Down Chart: Proposition for Better Alternatives. **International Journal of Digital Content Technology and its Applications**, v. 3, n. 3, 2009.

BECKET, K. *et al.* **Manifesto for agile software development**. [*S. l.*]: Agile Manifesto, c2001. Disponível em: http://agilemanifesto.org/. Acesso em: 28 out. 2019.

CHO, J. Issues and Challenges of Agile Software Development With Scrum. **Issues in Information Systems**, v. 9, n. 2, p. 188–195, 2008.

DINGSØYR, T. *et al*. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213–1221, 2012.

EBRAHIM, N. A.; AHMED, S.; TAHA, Z. Virtual teams: a literature review. **Australian Journal of Basic and Applied Sciences**, [S.l.], v. 3, p. 2653–2669, Nov. 2009.

ELORANTA, Veli-Pekka; KOSKIMIES, Kai; MIKKONEN, Tommi. Exploring ScrumBut - An empirical study of Scrum anti-patterns. **Information and Software Technology**, [*s. l.*], v. 74, p. 194-203, 2016.

JANES, Andrea A.; SUCCI, Giancarlo. The dark side of agile software development. In: **Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software**. ACM, 2012. p. 215-228.

MARCONI, M. A.; LAKATOS, E. M. **Técnicas de pesquisa**. 7. ed. São Paulo: Editora Atlas, 2012.

MARTINS, J.; DE MENEZES, R. M. T. Metodologia Ágil – Framework SCRUM – em Gerenciamento de Projetos De Software. **Cognitio/Pós-Graduação UNILINS 1.7**, p. 1–13, 2016.

MUNDRA, Ashish; MISRA, Sanjay; DHAWALE, Chitra A. Practical scrum-scrum team: Way to produce successful and quality software. In: **2013 13th International Conference on Computational Science and Its Applications**. IEEE, 2013. p. 119-123.

OPELT, Andreas et al. Agile contracts: creating and managing successful projects with Scrum. 1 ed. Canada: **John Wiley & Sons**, 2013.

POHL, Christoph; HOF, Hans-Joachim. Secure scrum: Development of secure software with scrum. **arXiv preprint arXiv:1507.02992**, 2015.

SCHWABER K.; SUTHERLAND J**.** Um guia definitivo para o Scrum: as regras do jogo. **Guia do Scrum,** [*s. l.*], p. 1-19, jul. 2013. Disponível em: http://www.scrumguides.org/docs/scrumguid e/v1/Scrum-Guide-Portuguese-BR.pdf. Acesso em: 06 out. 2019.

SCHWABER, Ken. Scrum development process. In: **Business object design and implementation**. Springer, London, 1997. p. 117-134.

SUTHERLAND, Jeff; HARRISON, Neil; RIDDLE, Joel. Teams that finish early accelerate faster: a pattern language for high performing scrum teams. In: **2014 47th Hawaii International Conference on System Sciences**. IEEE, 2014. p. 4722-4728.

TAKEUCHI, Hirotaka; NONAKA, Ikujiro. The new new product development game. **Harvard business review**, v. 64, n. 1, p. 137-146, 1986.