



O VALOR PERCEBIDO PELO CLIENTE NA GESTÃO DE PROJETOS DE SOFTWARE, UTILIZANDO O FRAMEWORK SCRUM

THE VALUE PERCEIVED BY THE CUSTOMER IN SOFTWARE PROJECT MANAGEMENT USING FRAMEWORK SCRUM

 **Reginaldo Aparecido Carneiro**
Doutor em Engenharia de Produção
Universidade Cesumar – UniCesumar.
Maringá, Paraná – Brasil.
rcarneiro73@gmail.com

 **Mauro Luiz Martens**
Doutor em Engenharia de Produção
Universidade Paulista – UNIP e Universidade de São Paulo – POLI USP
São Paulo, São Paulo – Brasil
mauro.martens@gmail.com

 **Vanessa Mesquita Blas Garcia**
Doutora em Administração - Gestão de Projetos
Universidade Nove de Julho – UNINOVE
São Paulo, São Paulo – Brasil.
vanessa.blasgarcia@gmail.com

 **Ernesto Dr Santibanez Gonzales**
Doutor em Engenharia de Produção
Universidade Paulista – UNIP
São Paulo, SP- Brasil e
Universidad de Talca - UTAL
Talca, Chile
santibanez.ernesto@gmail.com

Resumo

A qualidade e o sucesso de um projeto dependem da forma como se capta as necessidades e as expectativas do cliente. Dessa forma, o objetivo geral deste artigo incide sobre a verificação de como o valor percebido pelo cliente é considerado na entrega de um projeto a partir da utilização do *framework* ágil *Scrum*. Como metodologia, foi utilizada um estudo de caso a partir da análise de uma empresa brasileira atuante na área de *software*, localizada no interior do estado do Paraná, na cidade de Maringá. As informações foram coletadas com base em entrevista semiestruturada, realizada *in loco* a partir de um protocolo de pesquisa contendo 14 questões abertas, confrontando a referida entrevista com a literatura científica do assunto. O estudo realizado permitiu levantar práticas executadas pela empresa objeto de estudo que variam de acordo com a literatura, tais como o atendimento dos princípios comuns das metodologias ágeis, e a utilização eficiente dos pilares do *framework Scrum*. Em contrapartida, percebeu-se algumas divergências junto ao atendimento do cliente, no que tange à comunicação com o solicitante do projeto, pois, além do *Product Owner* (PO), outros envolvidos com o projeto também podem, a qualquer momento, entrar em contato com o cliente. Com isso, surge a possibilidade desta atitude promover ruídos na comunicação entre os envolvidos, o que pode resultar em um projeto que não atenda rigorosamente ao valor percebido pelo cliente que o requisitou.

Palavras-chave: Gestão ágil de projetos. Valor percebido. Cliente. *Framework Scrum*. Ágil. Product owner. *Software*.

Abstract

Quality and project success depend on the way in which the needs and expectations of the customer are captured. The main objective of this article focuses on the evaluation of how the value perceived by the customer is considered in the delivery of a project from the use of the agile *Scrum* framework. As a methodology, a case study was used from the analysis of a Brazilian company operating in the software area, located in the countryside of the state of Paraná, in Brazil. The information was collected based on a semi-structured interview, carried out in site, based on a research protocol containing 14 open questions, comparing the aforementioned interview with the scientific literature on the subject. The study carried out allowed us to understand practices performed by the company object of study that vary according to the literature, such as meeting the common principles of agile methodologies, and the efficient use of the pillars of the Scrum framework. On the other hand, some divergences were noticed with the customer service, regarding communication with the project requester, because in addition to the Product Owner (PO), others involved with the project can also, at any time, contact the customer. With this, the possibility arises of this attitude promoting noise in the communication between those involved, which can result in a project that does not rigorously meet the value perceived by the client who requested it.

Keywords: Agile project management. Value perceived. Customer. *Scrum* Framework. Agile. Product owner. software.

Cite como

American Psychological Association (APA)

Carneiro, R. A., Martens, M. L., Garcia, V. M. B., & Gonzales, E. D. R. S. (2022, set./dez.). O valor percebido pelo cliente na gestão de projetos de software, utilizando o Framework Scrum. *Revista de Gestão e Projetos (GeP)*, 13(3), 149-176. <https://doi.org/10.5585/gep.v13i3.22991>.

1 Introdução

As organizações contemporâneas precisam responder às novas condições de mercado, mais dinâmicas, com frequentes mudanças por meio de projetos com abordagens voltadas para o cliente, com possibilidades de flexibilização e uma gestão adequada de suas necessidades (Antlova, 2014). A maior parte das empresas, especialmente no setor de Tecnologia da Informação (TI), está mudando para a prática da metodologia ágil, que ajuda a aproveitar os benefícios de uma melhor colaboração e experiência do cliente (Rajan & Santhosh, 2021).

O que se percebe, é que as organizações de TI estão migrando para a prática da metodologia ágil para atender à demanda em constante mudança do mercado. Segundo ainda Rajan e Santhosh (2021), a adoção da metodologia ágil mostra a tendência de que as pessoas estão deixando de experimentar os modelos tradicionais de gerenciamento de projetos para construir flexibilidade no ambiente de trabalho. Tsai *et al.* (2018) destacam que, atualmente, várias empresas implementam métodos ágeis em processos práticos de desenvolvimento de software.

No que tange à utilização das metodologias ágeis, Sverrisdottir *et al.* (2014) apontam alguns dos principais métodos ágeis aplicados no mercado: *Scrum*, Extreme Programming (XP) (Beck, 2000), Kanban, *Crystal Family*, FDD (*Feature-Driven Development*) e DSDM (*Dynamic Systems Development Method*). Segundo Sommerville (2007), cada metodologia propõe processos diferentes, contudo compartilha um conjunto de princípios comuns, sendo eles: envolvimento do cliente (sua participação é fundamental para que o resultado final seja alcançado em sua plenitude), entrega incremental (o *software* passa a ser desenvolvido em incrementos, e o cliente especifica os requisitos a serem incluídos em cada incremento), foco nas pessoas e não em processo (os participantes da equipe devem devolver suas próprias formas de trabalhar), aceitação de mudanças (os requisitos do sistema certamente vão mudar) e, por fim, a manutenção da simplicidade (trata-se de evitar ações complexas em detrimento ao modo simplista).

A abordagem ágil visa capacitar e habilitar a equipe de projetos para entregas rápidas, com foco em valor para o negócio, por meio da integração e entendimento das necessidades dos clientes (Augustine, 2005). Alguns estudos fornecem suporte para os efeitos positivos dos métodos ágeis de gerenciamento de projetos na satisfação do cliente (Ferreira & Cohen, 2008; Ghani *et al.*, 2015; Recker *et al.*, 2017; Serrador & Pinto, 2015).

Em razão da importância do cliente e do valor que se deve dar para este principal interessado no desenvolvimento de sistemas por meio de práticas ágeis, é que esta pesquisa foi desenvolvida, pois visa à investigação da gestão ágil de projetos com foco no cliente. Assim, o objetivo geral deste artigo incide sobre a verificação de como o valor percebido pelo cliente é considerado na entrega de um projeto a partir da utilização do *framework* ágil *Scrum*. Nesse sentido, por meio de um estudo de caso, pretende-se verificar se os resultados alcançados pela empresa alvo, com a utilização do *framework Scrum*, realmente agregam valor para o cliente final, na busca de um projeto de qualidade, uma vez que o seu sucesso depende da forma como se capta as suas necessidades e expectativas.

2 Referencial teórico

Neste capítulo são abordados conceitos que direcionam o modelo teórico do estudo, bem como proporciona a conexão com os resultados obtidos no estudo.

2.1 O valor percebido pelo cliente

Existem muitas frases prontas cuja afirmativa tem sido apontada no dia a dia das empresas: “o cliente acima de tudo”, “o cliente sempre tem a razão”, ou “surpreenda o seu cliente”. Todas elas são muito aplicadas e divulgadas nos meios de comunicação, mesmo que não haja uma razão e um sentido em todas essas expressões. Os métodos de desenvolvimento ágil de sistemas de informação (ISD) defendem a participação ativa e contínua dos clientes no processo de ISD para criar software que atenda às necessidades dos clientes (Matook & Maruping, 2014; Ramesh et al., 2010; Wang et al., 2012).

No desenvolvimento de um software não é diferente. Nesse caso, a colaboração entre os desenvolvedores e os clientes pode ter efeito positivo. No estudo de Kurniawan et al. (2020), foi destacado que a gestão ágil de projetos precisa ser enquadrada para o mercado, para criar maior valor para os clientes. Nesse contexto, a capacidade dinâmica representada pela gestão ágil de projetos precisa se manifestar na forma de uma solução tangível que crie valor para o cliente e supere as ofertas dos concorrentes.

Segundo Kevin Sungkur e Ramasawmy (2014), o envolvimento do cliente é fundamental para a obtenção de uma melhor visibilidade do processo de desenvolvimento e seus problemas, assim como para promover melhor a ideia dos progressos realizados. É importante destacar que esse envolvimento deve ocorrer desde a concepção até a entrega do

projeto. Ainda de acordo com os autores, isso é importante para que se possa ter uma estreita interação entre o cliente e os desenvolvedores, durante cada iteração de desenvolvimento (*sprint*), de forma que quando os desenvolvedores têm dúvidas sobre os recursos do sistema, o cliente possa interagir e, conseqüentemente, livrar-se de qualquer erro de interpretação.

A partir do contexto supracitado, surge um questionamento: como gerenciar projetos a partir de um envolvimento mais efetivo do cliente? Como promover maior integração entre ele e os desenvolvedores de um projeto? Para tal, nos últimos anos, dentre os diversos métodos disponíveis no mercado, os métodos ágeis se popularizaram, porque eles dão essa relativa atenção às necessidades dos seus clientes. Segundo Niemi-Grundstrom (2014), o princípio mais importante prioriza a satisfação do cliente.

Uma característica marcante de qualquer abordagem ágil está no seu foco explícito, no que diz respeito à criação de valor aos clientes (Abrahamsson et al., 2002; Abrahamsson, Conboy, & Wang, 2009; Abrahamsson, Oza, & Siponen, 2010). Na essência, em projetos de software ágeis, o processo de desenvolvimento promove a criação de valor que depende da participação ativa dos clientes. De acordo com Belsis *et al.* (2014), os métodos ágeis têm como objetivo maximizar o valor de negócios do produto final, garantindo com que as partes interessadas se unam desde o início do processo de engenharia de software.

No mundo dos negócios, onde a complexidade e a concorrência estão aumentando, produtos e serviços orientados para o cliente estão ganhando consideração. Como em todos os setores, é uma prioridade para as equipes de software atender às necessidades dos clientes da maneira mais precisa em produtos e serviços de software o mais rápido possível (Selim et al., 2021). Sumariando, os princípios de gestão ágil de projetos a partir do manifesto ágil foram introduzidos (Beck et al., 2001; Lima e Amaral, 2022). O primeiro princípio incide sobre a satisfação do cliente: planejamento e entrega de bons serviços requerem profundo conhecimento do cliente. Em primeiro lugar, é preciso saber quem são eles e que tipos de necessidades têm, a fim de atendê-las, assim como resolver os seus problemas.

Em linhas gerais, o Manifesto Ágil destaca a importância da colaboração do cliente e a busca de suas necessidades quando afirma que a maior prioridade está na satisfação do cliente a partir de uma precoce e contínua entrega valiosa de software (Beck et al., 2001). Logo, verifica-se que uma das principais razões do movimento ágil incide sobre a contínua busca em satisfazer tais clientes (Beck et al., 2001; Highsmith, 2004; Shalloway *et al.*, 2009).

2.2 Metodologias ágeis

Segundo El-Najar *et al.* (2019), as metodologias ágeis dependem completamente da comunicação efetiva e contínua entre clientes e equipes de desenvolvimento, sendo que a falta de sustentação desta comunicação resultará em falha ou atraso no projeto. Com isso, o processo de desenvolvimento ágil de software eliminou a necessidade de documentação excessiva e reduziu o tempo perdido na finalização de todos os requisitos do projeto antes que a equipe começasse a trabalhar nele. Seabra e Almeida (2015) destacam que em projetos ágeis as mudanças nos requisitos são bem-vindas, mesmo que isso ocorra na fase final do processo: esta prática representa uma vantagem competitiva para o cliente.

Não há um conjunto de requisitos iniciais, pois, ao invés disso, há uma visão de alto nível sujeita a mudanças frequentes. Essa é uma maneira de lidar com as mudanças que surgem da interação que a equipe deve promover com cliente, bem como a sua participação ativa, além de sua articulação em assumir as responsabilidades para a entrega do produto final (Schwaber, 2004).

Importante ressaltar que não existe a melhor metodologia, tampouco a mais indicada. Na verdade, elas são contingenciais, isto é, cada uma se adapta de acordo com as necessidades da empresa. Entretanto, para este estudo, apenas o *framework Scrum* será utilizado em função de sua aplicação intensiva no mercado.

Na pesquisa realizada em 2020 pela VersionOne, empresa pioneira no mercado de ferramentas de gestão ágil, os entrevistados afirmaram que já utilizaram métodos ágeis e o *Scrum* foi o método ágil mais citado, sendo utilizado por 58% das empresas respondentes. A seguir destaca-se os aspectos do *framework Scrum*.

2.3 Framework Scrum

O *framework Scrum* foi criado por Ken Schwaber e Jeff Sutherland no ano de 1995, cujo nome tem origem do jogo de Rugby. Trata-se de uma das metodologias ágeis mais citadas, sendo que os dois maiores benefícios apontados pelos especialistas sobre tal metodologia foram as entregas contínuas ao cliente (Younas *et al.*, 2018; Parker *et al.*, 2017; Grapenthin *et al.*, 2015; Massari, 2014; Asnawi *et al.*, 2012) e as iterações de 1 a 2 semanas (Thorgren e Caiman, 2019; Al-Sakkaf *et al.*, 2017; West *et al.*, 2010; Paasivaara *et al.*, 2008).

De acordo com Grapenthin *et al.* (2015), *Scrum* é um *framework* iterativo para a gestão de projetos de *software* de acordo com os princípios ágeis. Ele permite que as equipes entreguem os recursos adequados no prazo, no orçamento e com qualidade. Ele também auxilia na organização de desenvolvimento do *software*, a fim de se adaptar às novas exigências do negócio e necessidades das partes interessadas, além de proteger as equipes de interrupções improdutivas do seu fluxo de trabalho (Kevin Sungkur & Ramasawmy, 2014).

O *Scrum* é regido por três pilares bem caracterizados, sendo eles: a transparência (precisa ter visibilidade e ser do conhecimento da equipe envolvida, inclusive do cliente), a inspeção (todos os processos devem ser inspecionados com frequência), e a adaptação (o processo deverá ser ajustado a todo instante para evitar desvios e comprometer a entrega do projeto) (Massari, 2014). É um processo de baixa sobrecarga para gestão e acompanhamento de desenvolvimento de software, pois pretende controlar esse processo usando uma estrutura de gerenciamento de projeto que envolve levantamento de requisitos, design e programação (Fitzgerald, Hartnett & Conboy (2006), *Scrum* é constituído de três papéis principais: *scrum master* (SM), *scrum team* (ST) ou equipe de desenvolvimento, e o *product owner* (PO). Além disso, ainda existem as cerimônias (*sprint*, *sprint planning meeting*, *daily scrum*, *sprint review meeting* e retrospectiva da *sprint*), e os artefatos (*product backlog*, *sprint backlog* e a definição de pronto – *definition of done*) (Kevin Sungkur & Ramasawmy, 2014).

O papel do SM é a aprovação e tem responsabilidade pela aplicação dos valores e práticas do *Scrum*, além de ter papel fundamental na remoção de impedimentos. O ST é tipicamente uma equipe multifuncional composta, em média, de cinco a dez pessoas que trabalham no projeto em tempo integral de maneira auto-organizada. O PO é tipicamente um gerente de unidade funcional que conhece o produto e sabe o que precisa ser construído para permitir que o projeto progrida (Sutherland & Schwaber, 2007). De acordo com Sverrisdottir *et al.* (2014), o PO deve encontrar um equilíbrio entre as diferentes propriedades do produto. O PO é uma única pessoa, e não um grupo de pessoas, porém é o representante de todos os interessados para o projeto. Segundo De Souza *et al.* (2021), o proprietário do produto (PO) tem um papel importante na definição dos requisitos para atender às necessidades do cliente e liderar o projeto, mas isso também é arriscado, pois os riscos relativos às suas funções podem surgir ao longo do projeto, então eles devem ser gerenciados adequadamente.

O PO trabalha em colaboração com toda a equipe sem ter qualquer autoridade sobre ela. Ele, assim como toda equipe, é responsável pelo sucesso do projeto, de tal forma que ele fornece

orientação e apoio a todos os envolvidos com o desenvolvimento, garantindo que as decisões difíceis sejam tomadas quando necessário. Ele deve ter habilidades de comunicação e de negociações. Ser PO geralmente é um trabalho de tempo integral (Pichler, 2010). Pode-se ressaltar que a tarefa mais importante do PO é tomar uma decisão sobre o que não deve ser priorizado, e tirar as consequências dessa decisão.

Abordagens ágeis de desenvolvimento de software têm sido propostas para lidar com mudanças constantes nos requisitos do projeto. No *Scrum*, segundo De Souza et al. (2022), o PO gerencia tais mudanças para que o software desenvolvido traga valor significativo para os clientes. No entanto, existem riscos potenciais envolvidos nas responsabilidades do PO que, se não forem gerenciados adequadamente, podem levar ao fracasso do projeto e perdas financeiras significativas.

De acordo com Hannay et al. (2017), o valor para o cliente é o mantra na gestão ágil. O PO está envolvido ao longo do caminho, e os atrasos são priorizados, com o melhor das intenções para maximizar o valor do negócio antecipadamente e eliminar o desperdício. No entanto, ainda segundo os autores, em muitos projetos de desenvolvimento de TI, a perplexidade permanece quanto à forma de expressar o valor do negócio nas decisões de processo e no sistema entregue.

Em linhas gerais, a disponibilidade do cliente é importante, uma vez que as mudanças nas necessidades que estão sendo determinadas diretamente por ele podem acelerar o projeto, bem como priorizar as suas necessidades, criando incrementos de valor mais elevados. Entretanto, a indisponibilidade do cliente na tomada de decisões do projeto é um problema, já que o contato contínuo com a ST gera custos para a organização que solicitou o software (Inayat et al., 2015; Curcio et al., 2018; Elghariani & Kama, 2016).

Destaca-se que o processo *scrum* apresenta 5 atividades, sendo elas: a reunião de planejamento da *sprint*, a *sprint*, o *scrum* diário, e a revisão da reunião da *sprint*. A reunião de planejamento da *sprint* ocorre entre o time *scrum*, o *Scrum Master*, e o *Product Owner*. Tais encontros ocorrem em dois momentos. No primeiro deles, duas atividades principais acontecem: primeiro, o grupo define o backlog de produto, que condiz com uma lista básica dos requisitos do projeto. Logo após, a equipe determina o objetivo da *sprint*, que condiz com o resultado formalizado. No segundo momento da reunião, o foco das atividades está na criação do *Sprint Backlog* propriamente dito.

O *framework Scrum* permite o fornecimento de iterações regulares (*sprints*) com valor para o cliente. O processo é construído sobre o trabalho em equipe, recebendo *feedback* frequente e uma comunicação transparente dentro da equipe e da empresa, assim como para o cliente. O processo ocorre em ciclos regulares, geralmente entre 2 e 4 semanas (Paasivaara *et al.*, 2008). A vantagem das *sprints* é a regularidade. ao final de cada sprint, durante a cerimônia de sprint review, a entrega combinada é verificada de acordo com a definição de pronto e, se necessário, ajustado de acordo com a priorização estabelecida pelo PO. Outra característica relevante é a melhoria contínua do processo, pois ao final de cada sprint e após a reunião de sprint review, deve ser realizada a reunião de retrospectiva, cujo objetivo é avaliar pontos positivos e avaliar quais ações são necessárias para corrigir os pontos negativos.

Para Cervone (2011), as vantagens de uma abordagem baseada em *Scrum* está em sua simplicidade. Dentro de um projeto ágil, os papéis estão claramente definidos. Os recursos podem ser desenvolvidos e testados em ciclos curtos de iteração. Cada um dos membros da equipe possui grande responsabilidade pela sua parte do projeto. Os métodos de gerenciamento ágil de projetos impõem comunicação extensiva, que ajuda as equipes a se organizarem de forma mais eficaz. Logo, isso pode levar a uma maior produtividade para todos os envolvidos.

Enfim, *Scrum* é um *framework* iterativo para o gerenciamento de projetos de software de acordo com os princípios ágeis. Ele permite que as equipes entreguem os recursos adequados no prazo, no orçamento e com a qualidade solicitada, sempre se adaptando às novas exigências do negócio, além de proteger as equipes de interrupções improdutivas ao seu fluxo de trabalho (Kevin Sungkur & Ramasawmy, 2014). Logo, a partir do trabalho com enfoque na equipe e na participação efetiva do cliente, há uma promoção na melhoria da comunicação e maximiza o apoio de todos, promovendo uma sensação agradável com aquilo que estão fazendo, gerando um aumento de produtividade.

De acordo com Bambauer-Sachse e Helbling (2021), os funcionários com contato direto com o cliente precisam aprender a se comunicar adequadamente com seus clientes, identificar as suas necessidades e interpretar as suas reações ao longo do processo. Ainda segundo os autores, investir em melhorias consideráveis de processo leva a uma melhoria mais substancial na satisfação do cliente do que simplesmente mudar de uma abordagem de gerenciamento de projetos orientada a planos para uma abordagem de gerenciamento de projetos ágil.

3 Procedimentos metodológicos

O método utilizado nesta pesquisa foi o estudo de caso que, de acordo com Goode e Hart (1975), trata-se de uma forma de organizar os dados, preservando o caráter unitário do objeto social estudado. Geralmente, essa abordagem inclui o desenvolvimento dessa unidade, que pode ser uma pessoa, uma família ou outro grupo social, um conjunto de relações ou processos, ou mesmo, uma cultura.

Yin (2005) elenca seis fontes de evidências que são utilizadas para a realização de um estudo de caso: documentação, registros em arquivos, entrevistas, observação direta, observação participante e artefatos físicos. Além dessas fontes, Yin (2005) também observa que inúmeras outras podem servir de base para pesquisa, incluindo filmes, fotografias, histórias de vida, entre outras.

Este artigo utilizou como fontes de evidência a observação direta, a entrevista e os registros bibliográficos concernentes ao tema pesquisado. Em linhas gerais, o artigo condiz com um estudo qualitativo descritivo realizado a partir da aplicação de uma entrevista qualitativa semiestruturada junto de um gestor de projetos de desenvolvimento de uma empresa na área de software de Gestão de Transporte localizada na cidade de Maringá, estado do Paraná.

Como o nome sugere e, segundo informações da própria empresa, ela atua com o desenvolvimento de software no segmento da logística empresarial, promovendo soluções na área de controle de entregas, emissão de notas fiscais, gestão de frotas e transportes, rastreamento de veículos, agendamento de cargas e descargas, entre outros serviços. O caso foi selecionado por acessibilidade.

A coleta de informações ocorreu nas dependências da própria empresa, com a participação direta do gerente de projetos. Como mencionado, utilizou-se a entrevista qualitativa semiestruturada. Com relação aos roteiros de entrevista semiestruturada, Richardson (2017) aponta que ela se aproxima mais de um diálogo, focado em um assunto específico, do que de uma entrevista mais formalizada. Isso porque, esse tipo de roteiro deve desdobrar-se em indicadores fundamentais que contemplem a abrangência das informações esperadas. Desse modo, o roteiro para uma entrevista semiestruturada deve ser elaborado com vistas a permitir flexibilidade na conversação, com o intuito de facilitar a obtenção de novos temas angariados pelo pesquisado, desde que sejam de relevância para a pesquisa.

A referida entrevista, com duração aproximada de 2 horas, foi realizada nas dependências da própria empresa, tendo como suporte a utilização de um protocolo de pesquisa

composto por 14 questões abertas. Os questionamentos foram realizados a partir de questões gerais sobre o *framework Scrum*, apontamentos gerais sobre papéis, processos e artefatos utilizados, encerrando com algumas considerações gerais que se fizeram necessárias perante as discussões ocorridas. Buscou também evidências suficientes para investigar o valor percebido pelo cliente na gestão de projetos a partir do *Scrum* como metodologia ágil de desenvolvimento de softwares.

De acordo com Yin (2005), ao estabelecer um protocolo define-se a questão da pesquisa, as fontes de informação e as questões que serão respondidas pela pesquisa. Observa-se que o protocolo de estudo de caso prevê que questões levantadas não serão direcionadas aos entrevistados, mas sim ao próprio pesquisador, com o objetivo de fornecer-lhe um guia de obtenção de um conjunto de evidências necessárias e suficientes que respondam às questões do protocolo.

A pesquisa foi realizada mediante a utilização de um equipamento eletrônico para fins de gravação. Optou-se por esta prática para que o entrevistado se sentisse mais à vontade na troca de informações durante a realização da pesquisa. Posteriormente, a entrevista passou por uma transcrição digital para não perder nenhuma informação importante ao conteúdo do artigo.

Considerando as fontes utilizadas no presente artigo, a triangulação de métodos e dados também foi utilizada para facilitar a análise de conteúdo. Essa técnica de triangulação permite que o pesquisador utilize diferentes métodos de pesquisa, como análises qualitativas e quantitativas, para o estudo de um mesmo fenômeno.

Com o resultado transcrito e a utilização da literatura sobre metodologias ágeis, pôde-se alcançar informações de qualidade efetiva diante dos objetivos angariados neste trabalho. Foi justamente a partir dessas informações que os resultados do estudo de caso foram compilados, bem como a discussão realizada e, conseqüentemente, a conclusão efetivada. Na sequência, estão contemplados os resultados.

4 Apresentação dos resultados

A empresa *LS*, alvo do estudo de caso, está localizada na cidade de Maringá, interior do estado do Paraná, considerada como um centro de excelência no desenvolvimento de *softwares*, sendo também sede de um Arranjo Produtivo Local (APL) nesta área, com reconhecimento nacional. É importante ressaltar que atualmente a empresa faz parte do *Software By Maringá*, que é uma entidade de classe que congrega empresas de desenvolvimento de *software* que

objetiva a busca da melhoria na qualificação e no desenvolvimento das empresas de *software* e de seus colaboradores.

A empresa foi fundada em 2005, e tem foco na área de Gestão de Transportes (logística), buscando comprometimento com a qualidade de seus Sistemas de Gestão. A sua missão é “*oferecer soluções de valor para o segmento de transportes no apoio à decisão gerando riquezas a todos os envolvidos*”. O quadro de colaboradores contempla profissionais qualificados, programadores, consultores e administradores, onde, juntos, desenvolvem pesquisas para desenvolver *softwares* customizados para cada empresa em seus variados segmentos.

O foco da *LS* é o desenvolvimento de Sistemas de Gestão Corporativo, direcionados a operadores logísticos, empresas com frotas próprias, transportadoras, entre outros segmentos. Atualmente, a empresa não oferece nenhum *software* de prateleira, pois todos os seus produtos são customizados.

Destaca-se que no primeiro semestre de 2015, a *LS* recebeu a certificação MPS-BR, nível F. Trata-se de um programa mobilizador para melhorar a capacidade de desenvolvimento de *software* nas empresas brasileiras. Essa certificação leva em consideração normas e modelos reconhecidos internacionalmente, além das boas práticas da engenharia de *software* e as necessidades de negócio da indústria de *software* nacional.

Em suma, a empresa *LS* acredita que a integração tecnológica de todos os setores de uma empresa aperfeiçoa a cadeia produtiva para obter melhores resultados, proporcionando vantagem competitiva.

4.1 Aplicabilidade do Framework Scrum

Segundo o gestor de projetos, em função das necessidades da empresa, a direção realizou algumas adequações no que tange à aplicação do *framework Scrum*. A organização trabalha com projetos mensais, realizando *sprints* no meio e no final do período (15 e 30 dias, respectivamente), salvo em situações específicas, como nas solicitações de órgãos do governo. A empresa executa um único projeto por período, de tal forma que eles são mensais, isto é, geralmente o projeto é executado do primeiro dia até o último dia útil do mês. Logo, um projeto tem início imediatamente após o término de outro, e assim sucessivamente. A empresa apresenta um *backlog* de produto bem extenso (fila).

De acordo com Kevin Sungkur e Ramasawmy (2014), as práticas do *Scrum* requerem uma estreita colaboração entre o *Product Owner* (PO), o *Scrum Master* (SM) e a equipe *Scrum* (ST). O envolvimento do cliente ativo é crucial, desde o início até a entrega do software, de modo que uma melhor visibilidade do processo de desenvolvimento é obtida. Na prática do *Scrum*, a *LS* possui um PO, um SM, quatro pessoas responsáveis pelo time de desenvolvimento, duas pessoas representam a equipe de suporte (suporte interno representado pelo diretor e um subordinado), além de 4 pessoas representando a equipe de consultoria e implantação (que se desloca até o cliente). Após a certificação MPS-BR, a empresa passou a contar com um analista de qualidade (responsável em verificar se os processos estão sendo seguidos conforme a documentação). Desta forma, percebe-se que a aplicação e o funcionamento do *framework Scrum* ocorre mediante o envolvimento de 13 colaboradores.

Segundo Kevin Sungkur e Ramasawmy (2014), o PO representa as partes interessadas ou clientes. Seu papel é o de comunicar com o SM e com a equipe, que é composta tipicamente de cinco a nove pessoas, número este que é composto principalmente de desenvolvedores de software. Na empresa, em tempo integral, o PO detém a função de gerente de projeto de clientes, e ele não tem ligação alguma com o gerente de produto.

Para a realização do projeto, a empresa faz uso de cerimônias onde há o envolvimento do PO, do SM e do ST. Para Schwaber e Beedle (2001), a reunião *scrum* diária normalmente leva 15 minutos, durante a qual, cada membro da equipe responde a algumas perguntas *Scrum*. No caso da empresa em análise, para cada projeto, um único PO. Essa reunião serve para fazer o levantamento das necessidades, priorizar as solicitações e definir a sequência que o PO estabeleceu. Destaca-se que antes da reunião de priorização é definido um cronograma de entrega. O ST faz uso do *planning poker* em suas atividades, que é uma técnica estimativa cujo objetivo é permitir que todos os envolvidos estejam comprometidos na implementação do *sprint* para que juntos possam chegar a um indicador comum para o time.

A equipe de desenvolvimento não tem contato com o cliente em si. Na verdade, o cliente do ST é a equipe de suporte, bem como a equipe de consultoria e implantação da empresa, onde o PO é o seu líder. Quando a equipe de desenvolvimento precisa realizar alguma apresentação, ela o faz diretamente para a equipe de consultoria e suporte. Logo, a apresentação para o cliente final (após homologação) também é realizada pela equipe de consultoria.

Sempre que necessário, a empresa substitui o PO em função de possíveis interesses apresentados nessa função. Responsável pela verificação do valor percebido pelo cliente, com

a chegada da solicitação, o PO faz uma priorização das ações. O SM também pode estimar a tarefa, rejeitando-a ou não, além da possibilidade de pedir remoções. Se estiver tudo correto, a situação é encaminhada para o PO, este a prioriza de acordo com a necessidade da empresa solicitante (cliente final), conversando com o suporte se houver necessidade.

A função do SM é desempenhada pelo próprio gerente de projetos de desenvolvimento (ou gerente de produção), sendo responsável pela coordenação, distribuição das tarefas e subsídio de informações, além de facilitar e gerenciar conflitos no sentido de a equipe desenvolver o produto com a maior eficiência e eficácia possível. É ele quem negocia com o PO, realizando os controles das cerimônias dentro do *sprint* de um projeto. O SM também negocia com a equipe de desenvolvimento, sendo uma espécie de filtro junto a outros envolvidos com o processo. Para Oliveira e Lima (2012), o SM é quem garante que tudo esteja sendo seguido em todos os seus fundamentos, tendo como responsabilidades a remoção dos impedimentos, a melhoria da produtividade do ST, a promoção da comunicação e a proteção do time de interferências externas.

Em se tratando do ST, a empresa ainda trabalha com duas equipes de desenvolvedores (desktop e web), contudo ela está unificando ambas as equipes aos poucos. A sua pretensão é unificar o conhecimento de ambas nas áreas e formatar uma única equipe. Atualmente, a LS não tem mais um “testador” para os seus produtos finais. A própria equipe que produz o produto tem a incumbência de entregá-lo funcionando, e em condições de uso. É importante ressaltar que não existe um líder específico dentro do ST.

Ainda em relação ao ST, esta equipe tem bastante liberdade na realização de suas tarefas. Na verdade, ela sabe qual é a ordem das tarefas em função de suas prioridades. Se houver alguma dúvida, eles buscam saná-la. Em caso de dúvida técnica, recorre-se àquele integrante da equipe que tem mais experiência no assunto. Para dúvidas mais genéricas, a equipe busca o SM. O importante é que no final do *sprint* todas as tarefas estejam entregues, agregando valor às necessidades do cliente final, caso contrário, deve-se negociar com o PO.

4.2 Contribuição do scrum na transparência, inspeção e adaptação nos projetos

Para a LS, como o *Scrum* é um *framework* aberto para todos, os envolvidos têm como visualizar o andamento do projeto, promovendo transparência no processo. Em um estudo de Mchugh, Conboy e Lang (2012), os autores descobriram que os métodos ágeis promovem maior confiança, aumentando assim a transparência, a prestação de contas, a comunicação, a partilha

de conhecimentos e feedback. Todas as equipes ágeis que foram pesquisadas no trabalho desses autores concordaram que a utilização das práticas ágeis proporciona maior transparência e visibilidade nos seus projetos, tanto dentro da equipe quanto na organização como um todo. Com isso, a empresa consegue medir a produtividade de seus colaboradores mediante suas entregas.

Apesar da empresa estudada ainda não possuir indicadores específicos por colaborador, ela tem condições de levantar tais informações: nem sempre o colaborador mais rápido é melhor que outro mais lento. A inspeção na empresa é realizada periodicamente para cada projeto, envolvendo nessa tarefa tanto o PO quanto os *stakeholders*. No período de 30 dias (onde ocorrem os dois *sprints*), a empresa realiza um monitoramento a cada 15 dias. Havendo algum contratempo, há uma negociação junto ao PO para rever o atendimento da necessidade do cliente (valor).

À medida que o projeto é executado, ele pode sofrer qualquer tipo de modificação, demonstrando uma adaptação à circunstância. O desenvolvimento ágil de software abrange metodologias caracterizadas por uma disponibilidade contínua para perceber rapidamente a mudança, sendo proativa ou reativa em aceitá-la, aprendendo com ela, e contribuindo simultaneamente para o valor do cliente (Van Waardenburg & Van Vliet, 2013). Quando isso, o gestor de projeto recebe uma solicitação de mudança e, junto do PO, adapta a situação em busca do melhor resultado para o cliente. Essas mudanças podem ocorrer tanto por motivos internos (situações atreladas aos desenvolvedores), quanto por motivos externos (algum conhecimento insuficiente para com uma parte do produto a ser desenvolvido).

Em linhas gerais, a abordagem de gerenciamento ágil de projetos promove uma ação mais direta, flexível, iterativa e com menos intervenção gerencial para produzir inovação superior e valor para o cliente (Arefazar et al., 2022; Conforto et al., 2014). A orientação para o mercado é uma orientação estratégica que visa criar e entregar valor superior aos clientes em busca de vantagens competitivas no mercado (Guo et al., 2019).

4.3 Envolvimento do PO com o ST

O PO, junto da equipe de suporte e da equipe de consultoria, é tratado pela empresa como a linha de frente com o cliente final do projeto (solicitante). Ele tem como qualificação os seguintes atributos: formado em informática, com experiência em consultoria, além de *know how* em regra de negócio. Entretanto, ele não tem nenhuma certificação na função, pois os

únicos que têm certificação na empresa são o gestor de projetos e o desenvolvedor sênior (certificação *ScrumMaster* pela *ScrumAlliance*).

Existe muita discussão a respeito da busca da melhor certificação ágil. Muitos eram adeptos à certificação *Scrum Alliance*. Outros apoiavam a certificação *Scrum.org*. Entretanto, com o objetivo de unir suas forças e minimizar essas diferenças, houve a criação de um padrão único para o *Scrum*, denominado de *SCRUMGuides.org* (Schwaber & Sutherland, 2013).

É importante destacar que, na relação do PO junto ao ST, a comunicação ocorre apenas na forma verbal, sem nenhum tipo de documentação. Existe uma liberdade nesse sentido, pois os colaboradores (inclusive o PO) se encontram no mesmo ambiente de trabalho, permitindo melhor comunicação entre eles. É perceptível que a empresa busca evitar muita formalidade nesse sentido, pois a questão é fazer a coisa acontecer, entregar o produto no prazo, funcionando e com qualidade.

A partir dessa análise, fica evidente a importância da comunicação no que diz respeito ao projeto. Segundo Parker *et al.* (2017), há evidências claras que mostram a relevância de uma comunicação eficaz ao executar uma intervenção para buscar a melhoria do desempenho da equipe. Em particular, entender a necessidade das partes interessadas é primordial, permitindo desenhar uma estratégia de comunicação. De acordo ainda com os autores, cada fase de uma intervenção baseada em projeto requer diferentes estilos de comunicação.

Atualmente, a autoridade da *LS* se encontra centralizada no gerente de projeto e não no PO, pois ele acaba sendo o facilitador da equipe (enquanto *SM*). Na verdade, o PO e os desenvolvedores estão dispostos no mesmo nível. Para que tudo isso ocorra, o apoio da alta gerência é fundamental, sendo isso muito evidente em relação à *LS*. Muitas dúvidas acabam sendo sanadas junto à proprietária da empresa pelo fato de ela ser a responsável pelo desenvolvimento da ferramenta.

Na relação entre PO e desenvolvedores, a colaboração entre eles flui naturalmente, salvo em raras situações que envolvam problemas de personalidade. Independentemente dessa frequência, o gerente de projeto sempre se envolve para minimizar tais problemas. Como ele fica na mesma sala onde o PO e os desenvolvedores se encontram, facilmente ele tem acesso a qualquer ocorrência.

É importante ressaltar que o tipo de produto que é produzido pode influenciar no nível de *stress* dos colaboradores. Sistemas mais complexos certamente estarão mais sujeitos ao aumento deste nível em detrimento a sistemas mais “tradicionalis”. Isso é pontuado no final

(*sprint* de retrospectiva) por meio de um debate, em que são disponibilizadas informações sobre os pontos de melhoria, os pontos negativos, os pontos positivos daquele *sprint*, e a verificação da necessidade de treinamento para potencializar o conhecimento da equipe.

O trabalho executado pelos colaboradores da empresa com ciclos mensais e entregas quinzenais possibilita a inserção de novos requisitos no projeto de forma contínua. Quando há muita coisa a se produzir, o gestor de projeto promove “quebras” para coisas mais simples, o que traz facilidades para a equipe de desenvolvedores. *O planejamento, estimativa e controle de mecanismos de modelos de processos ágeis dependem significativamente de um conjunto fixo de tarefas que deve ser estabelecido para cada sprint* (Grapenthin, 2015). Isso traz um impacto significativo sobre o fator motivacional dos envolvidos, pois segundo a empresa, quanto mais particionado for o projeto, maior será a motivação da equipe para a sua realização. Projetos grandes e com entregas demoradas podem desestimular os envolvidos e, por conseguinte, prejudicar o resultado final do projeto.

4.4 A valor percebido pelo cliente no processo

A LS tem a percepção de que o uso do *framework Scrum* contribui positivamente no atendimento e na busca da qualidade para o seu cliente, sempre com a pretensão de agregar valor perceptível no produto final. Problemas podem ocorrer na fase intermediária, que pode surgir entre a transformação daquilo que o cliente realmente deseja para aquilo que será de fato construído. Destaca-se aqui a necessidade no detalhamento máximo da informação. Bergadano *et al.* (2014) destacam que as metodologias ágeis de desenvolvimento regulam a interação entre clientes e desenvolvedores. Segundo os autores, elas reconhecem a importância de um fluxo contínuo de comunicação entre os participantes do projeto. Tal fluxo deve compreender toda a duração desse projeto e não deve ser limitado apenas às primeiras fases.

Segundo a empresa, as possíveis falhas junto do cliente incorrem de três causas distintas: (1) ausência de um posicionamento adequado do cliente junto à sua solicitação, (2) o PO deixa de expor algum detalhe importante para o projeto, e (3) quando o próprio time de desenvolvedores não compreende muito bem algumas das tarefas. Pelo histórico da LS, a segunda causa é a mais recorrente. Segundo Parket *et al.* (2017), o PO desempenha um papel essencial durante todo o ciclo de vida do projeto, com a responsabilidade de gerir os requisitos e assegurar que o software traz um valor significativo para os clientes.

Para evitar tais falhas, a empresa pratica algumas ações: um dia antes da entrega da *sprint*, existe uma reunião (chamada de cerimônia de homologação), em que o PO analisa a apresentação e identifica a ausência de algo. A empresa busca o atendimento da necessidade do cliente, e se isto não for entregue, não há continuidade no processo diante da possibilidade de surgimento de um problema (isso pode baixar a nota da empresa). Se deixasse para fazer a averiguação apenas no final (característica típica dos projetos tradicionais), certamente não daria tempo para converter a situação.

Mensalmente, ocorre uma reunião na *LS* onde são apontados alguns indicadores. Geralmente ela ocorre na metade do mês, pois isso promove condições e tempo hábil para geração de algumas métricas, tais como: consultoria, comercial, suporte (PO). Existe também uma pesquisa de satisfação que é avaliada pelo cliente final após recebimento do produto. A pretensão dessa prática é permitir averiguar se o valor percebido pelo cliente com o produto (durante e ao final de todo o processo) teve efetividade. Nesse caso, há uma pessoa responsável pelo questionamento sobre a qualidade do produto da empresa, que se utiliza de ligações para alguns clientes, bem como de *e-mails*, contato direto com o usuário, ou até mesmo, junto ao diretor da empresa (solicitante do produto).

Por sua vez, o cliente final atribui a sua nota de acordo com o produto, e na reunião junto da empresa, essas notas (indicadores) são apresentadas. É importante ressaltar que cada departamento da empresa possui metas específicas para serem cumpridas (número de bugs registrado em um período, quantidade de hora para novas funcionalidades, capacitação, e assim por diante). A empresa também adota um sistema de bonificação pelo qual o colaborador recebe uma gratificação caso o seu departamento alcance as metas estipuladas.

E suma, a *LS* trabalha a favor do cliente final, pois se esse identificar algum problema, então há necessidade de analisar os fatores internos para identificação da causa do problema. Existem as avaliações parciais e uma avaliação final por parte do cliente do projeto para verificar o seu grau de satisfação. A avaliação durante o projeto é realizada pela equipe interna (mediante as métricas estipuladas pela empresa). O cliente somente pode avaliar a qualidade do produto quando da sua utilização, isto é, quando ele é entregue definitivamente para o uso. Esse foco na qualidade do processo não apenas melhora a satisfação do cliente, mas também otimiza a experiência geral do cliente e pode apoiar o sucesso a longo prazo da organização (Witell et al., 2020).

5 Discussão dos resultados

Após confrontar informações entre o funcionamento do *framework Scrum* de acordo com a literatura e a forma com que este método ágil é aplicado em uma empresa de *software* do interior do Paraná, na área de Gestão de Transporte, chegou-se em algumas constatações. No geral, relata-se que o produto resultante de um projeto que é executado a partir da adoção do *Scrum* pode promover maior valor ao cliente final e, conseqüentemente, melhoria na qualidade a partir do seu envolvimento junto à produção do produto.

Para facilitar a compreensão, as conclusões oriundas deste estudo foram subdivididas em dois momentos. No primeiro, é destacado as ações praticadas pela empresa de *software* que segue à risca algumas particularidades da metodologia *Scrum* em alinhamento com a literatura. No segundo momento, são pontuadas algumas contradições, isto é, itens que a empresa não põe em prática de acordo com os apontamentos evidenciados pela literatura. Ao final, contemplam-se uma conclusão seguida de algumas sugestões para trabalhos futuros.

5.1 Alinhamento com a literatura

De acordo com Sommerville (2007), as metodologias ágeis compartilham um conjunto de princípios comuns, sendo eles: o envolvimento do cliente, a entrega incremental, o foco nas pessoas e não em processo, a aceitação de mudanças, e a manutenção da simplicidade. A empresa atende a todos esses princípios com uma pequena ressalva no primeiro item que será mais bem detalhado adiante (item 5.2).

Massari (2014) deixa evidente que o *framework Scrum* é regido por três pilares distintos, sendo eles: transparência, inspeção e adaptação. A empresa em estudo atende todos os pilares a partir da aplicação da metodologia, pois a sua equipe tem visibilidade no andamento do projeto, com processos inspecionados dentro de uma frequência controlada, e permitindo adaptações ao longo do seu desenvolvimento, com claras possibilidades de mudanças ao longo do projeto.

A literatura é muito clara quando relata que o processo deve ocorrer em ciclos regulares, e que geralmente eles não ultrapassem um intervalo de 30 dias. O *Scrum* estrutura um projeto a partir de uma série de iterações que duram de uma a quatro semanas (Schwaber, 2004). No caso da empresa, essa regularidade está presente, de tal forma que a entrega acontece exatamente nesse período, com uma revisão (*sprint*) alocada exatamente na metade desse prazo (15 dias). Essa regularidade promove condições favoráveis para a empresa, pois com tais

verificações os ajustes podem ocorrer até o final do processo, e o atendimento do cliente final é realizado com maior precisão.

5.2 Gaps e possibilidades de melhoria no alinhamento com a literatura

Conforme relatado no estudo, o contato junto ao cliente final (solicitante do projeto) pode ocorrer via equipe de suporte e equipe de consultoria e implantação. Nesta última, o PO é o responsável pela equipe, o que o torna soberano nas decisões finais. Desse modo, destaca-se que tais contatos vão ao encontro daquilo que é relatado na literatura. Assim, de acordo com Sverrisdottir *et al.* (2004), o PO é constituído por uma única pessoa, e não um grupo de pessoas, pois ele é o representante de todos os interessados para o projeto.

Nesse mesmo sentido, significa dizer que a necessidade do cliente pode ser comunicada para vários canais representantes da empresa. Como a empresa ostenta uma equipe de suporte (2 representantes) e uma equipe de consultoria e implantação (4 representantes) em contato direto junto ao cliente final, há possibilidades da comunicação se tornar truncada, pois seis pessoas podem, simultaneamente, ter acesso e/ou solicitar informações junto ao cliente final (seja no início ou durante o projeto). Na verdade, isso pode provocar a ausência de alguma informação importante e, conseqüentemente, impactar no resultado final do projeto. Reforça-se aqui a afirmação de Bambauer-Sachse e Helbling (2021), que destacam que os funcionários com contato direto com o cliente precisam aprender a se comunicar adequadamente com seus clientes, identificar as suas necessidades e interpretar as suas reações ao longo de todo o processo.

Ainda segundo a literatura, o time de desenvolvedores (ST) deve ser composto de 5 a 10 pessoas trabalhando no projeto em tempo integral. De acordo com Paasivaara *et al.* (2008), o número de pessoas em cada equipe varia entre os projetos de liberação e iterações, principalmente porque diferentes versões enfatizam diferentes módulos. Segundo os autores, o tamanho da equipe varia de dois a nove pessoas.

Além disso, destaca-se ainda que o papel de liderança na equipe não deve ser fixo, com possibilidades de mudanças de acordo com as necessidades específicas da empresa. No desenvolvimento ágil, segundo Moe *et al.* (2009), a liderança é partilhada, e decisões importantes sobre o que fazer e como fazê-las são realizadas por meio de um processo iterativo que envolve muitas pessoas e que influenciam uns aos outros, e não apenas uma única pessoa.

Na organização em análise, o ST é composto por apenas quatro colaboradores. O mais importante é frisar que não existe nenhum papel de liderança dentro da equipe de desenvolvedores, pois este é representado pelo *Scrum Master*. Quando a equipe e os líderes da equipe compartilham a liderança, ela é transferida para a pessoa com a chave de conhecimentos, competências e habilidades relacionadas com as questões específicas que enfrentam a equipe a qualquer momento (Pearce, 2004).

A literatura aponta que os métodos ágeis vão ao encontro daquilo que é de valor para o cliente. Isto é, a qualidade do produto requerido dependerá do seu envolvimento junto à produção da empresa. Conforme Nerur *et al.* (2005), em desenvolvimento ágil de software, desenvolvedores trabalham em pequenas equipes com clientes que representam os usuários do sistema. Os recursos a serem implementados em cada ciclo de desenvolvimento são decididas conjuntamente pelo cliente e o resto da equipe de desenvolvimento. Ainda de acordo com os autores, a tomada de decisão colaborativa, envolvendo as partes interessadas com diversas origens e objetivos é, portanto, uma característica do desenvolvimento ágil.

No caso da *LS*, ficou evidente que ela atende apenas um projeto por período (projetos mensais), de tal forma que se outras demandas surgirem nesse intervalo, a empresa não conseguirá atendê-lo. Não é sem motivo que a empresa destaca que o seu *backlog* de produto é bem extenso, gerando grandes filas. Numa projeção otimista de aumento de demanda no curto prazo é provável que a empresa tenha algum tipo de problema por conta do não atendimento de todos os clientes que solicitem novos projetos.

Apesar de não ser uma informação em evidência, Pichler (2010) menciona que o PO deve ser qualificado e de fácil acesso para que a equipe seja capaz de entregar o projeto de acordo com aquilo que foi solicitado. Por sua vez, o que foi percebido é que o PO da empresa objeto de estudo não tem uma certificação específica para sua função, mesmo com outros atributos que o destacou para assumir o referido cargo.

Segundo relato, o histórico da empresa estudada demonstra que a maior causa de falhas na comunicação ocorrida junto do cliente final acontece quando o PO deixa de expor algum detalhe importante para o desenvolvimento do projeto. Segundo Kevin Sungkur e Ramasawmy (2014), tanto o PO quanto a equipe de suporte e de consultoria têm contato direto com o cliente, sendo o PO soberano. Para a empresa, os motivos dessa intermediação junto ao cliente estão nos fatores praticidade e responsabilidade. Muitos clientes da empresa não sabem o que querem, e daí esse intermediário facilita o levantamento da informação.

6 Considerações finais

Ao final da pesquisa ficou perceptível que a empresa faz uso do *framework Scrum* por ter consciência sobre a qualidade que esse método ágil promove diante da entrega de um produto para o seu cliente final. Isto é, ela tem a percepção de que esta metodologia promove melhor contato junto ao seu solicitante, maximizando o seu valor percebido e, como consequência, o projeto resultante sendo alcançado com maior efetividade.

Dentre os problemas identificados e anunciados nas discussões supracitadas, pode-se reforçar que a maior falha identificada reside na possibilidade do surgimento de ruídos no princípio de todo o processo, e que podem perdurar nos próximos contatos também. Como se presenciou a participação de vários atores, além do PO, no contato junto do cliente solicitante do projeto, então há de se constatar que essa forma de interação pode não surtir uma comunicação efetiva entre as partes e, por conseguinte, um detalhamento ideal da qualidade esperada da informação. Unindo a essa situação posta, a ausência de uma certificação do PO, nessa função, também pode ser um quesito a mais que corrobora com uma possível ineficiência no manuseio das informações corretas e de extrema necessidade para alimentar a todos os envolvidos no processo de estruturação de um projeto.

Enfim, apesar das falhas identificadas, fica evidente que a empresa estudada busca o atendimento de seu cliente final, sempre disposta a analisar os fatores que a levam na causa do problema. Evidencia-se que a empresa tem a incumbência de entregar o produto ao cliente dentro do prazo estipulado, em perfeitas condições de uso. Logo, constata-se no estudo realizado que o *framework Scrum* utilizado em sua essência pode contribuir favoravelmente na qualidade do produto solicitado e, por fim, no valor percebido pelo cliente solicitante do projeto. Isso ficou evidenciado junto às palavras do próprio gestor de projetos que foi entrevistado. Significa dizer que, quando bem empregada, o valor percebido pelo cliente é tido como um meio valioso para a concretização de um projeto a partir da utilização do *framework Scrum*.

Em se tratando de limitações de estudo, o referido trabalho apresenta apontamentos merecedores de atenção. Em primeiro lugar, trata-se de um estudo de caso único. Significa dizer que os argumentos utilizados e destacados aqui podem não refletir a realidade de outras empresas do mesmo segmento. Além disso, condiz com uma empresa de pequeno porte. Isso também tem uma influência representativa, uma vez que outras organizações, mesmo que seja desse mesmo segmento, porém, de porte diferente (maior ou menor), podem apresentar uma combinação de fatores diferentes dos que foram trabalhados e argumentados neste estudo.

Por fim, por se tratar de um caso escolhido por acessibilidade, é provável que, em algum momento, algumas variáveis trabalhadas neste estudo poderão não refletir da mesma forma em outras empresas, o que desponta para a necessidade de outros estudos, necessitando de uma análise detalhada das variáveis envolvidas nesse mesmo contexto, e posterior conclusão de tal investigação.

Como sugestões para trabalhos futuros, este estudo poderia ser realizado com um número maior de empresas (multicasos) da área de *software* para um levantamento mais completo de informações e uma constatação mais precisa diante da literatura do assunto. Outras áreas (prestação de serviços e construção) também merecem destaque para se ter uma percepção se o *framework Scrum* também pode contribuir a partir do cliente e, conseqüentemente, oferecer um produto de maior valor percebido. Além disso, sugere-se desenvolver estudos junto dos outros frameworks ágeis citados nesse artigo, para averiguar a efetividade desses métodos em detrimento ao *Scrum*, com o objetivo de verificar se há alguma conexão entre os tipos de negócios e os *frameworks* mais ideais para cada um deles.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) e a UniCesumar pelo suporte financeiro.

Referências

- Abrahamsson, P.; Oza, N.; & Siponen, M. T. (2010). Agile Software Development Methods: A Comparative Review 1. In: Agile software development. *Springer Berlin Heidelberg*, 31-59. https://doi.org/10.1007/978-3-642-12575-1_3
- Abrahamsson, P; Conboy, K; & Wang, X. (2009). ‘Lots done, more to do’: the current state of agile systems development research. *European Journal of Information Systems*, 18(4), 281-284. <https://doi.org/10.1057/ejis.2009.2>
- Abrahamsson, P; Salo, O.; Ronkainen, J.; & Warsta, J. (2002). Agile software development methods: Review and analysis. <https://doi.org/10.48550/arXiv.1709.08439>.
- Al-Sakkaf, A. M., Hashim, N. L., & Omar, M. (2017). Using hierarchical cluster analysis to generate clusters of agile practices. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(1-2), 53-56.
- Antlova, K. (2014). Agile Approach in the Project Management of the Czech Companies. *Procedia Technology*, 16 (2014), 929 – 933. <https://doi.org/10.1016/j.protcy.2014.10.045>
- Asnawi, A. L.; Gravell, A. M.; & Wills, G. B. (2012). Emergence of agile methods: perceptions from software practitioners in Malaysia. In: AGILE India, 2012. IEEE, 2012. <https://doi.org/10.1109/AgileIndia.2012.14>
- Arefazar, Y., Nazari, A., Hafezi, M. R., & Maghool, S. A. H. (2022). Prioritizing agile project management strategies as a change management tool in construction projects. *International Journal of Construction Management*, 22(4), 678-689. <https://doi.org/10.1080/15623599.2019.1644757>
- Augustine, S. (2005). *Managing Agile Projects*. Virginia: Prentice Hall PTR.
- Bambauer-Sachse, S., & Helbling, T. (2021). Customer satisfaction with business services: is agile better? *Journal of Business & Industrial Marketing*. <https://doi.org/10.1108/jbim-04-2020-0221>
- Beck, K., Beedle, M., Van Bennekum, A., 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.
- Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley professional.
- Belsis, P.; Koutoumanos, A.; & Sgouropoulou, C. (2014). PBURC: a patterns-based, unsupervised requirements clustering framework for distributed agile software development. *Requirements Engineering*, 19(2), 213-225. <https://doi.org/10.1007/s00766-013-0172-9>

- Bergadano, F.; Bosio, G.; & Spagnolo, S. (2014). Supporting collaboration between customers and developers: a framework for distributed, agile software development. *International Journal of Distributed Systems and Technologies*, 5(2), 1-16. <https://doi.org/10.4018/ijdst.2014040101>
- Cervone, H. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives*, 27, 18-22. <https://doi.org/10.1108/10650751111106528>
- Conboy, K. (2009). Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329-354. <https://doi.org/10.1287/isre.1090.0236>
- Conforto, E. C., Salum, F., Amaral, D. C., Da Silva, S. L., & De Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development? *Project Management Journal*, 45(3), 21-34. <https://doi.org/10.1002/pmj.21410>
- Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 139, 32-50. <https://doi.org/10.1016/j.jss.2018.01.036>
- De Souza Lopes, S., de Souza, R. C. G., de Godoi Contessoto, A., de Oliveira, A. L., & Braga, R. T. V. (2022). Automated Support for Risk Management in Scrum Agile Projects. In *International Conference on Enterprise Information Systems* (pp. 236-255). Springer, Cham.
- De Souza Lopes, S., de Souza, R. C. G., Contessoto, A. D. G., de Oliveira, A. L., & Braga, R. T. V. (2021, April). A Risk Management Framework for Scrum Projects. In *ICEIS (2)* (pp. 30-40). <https://doi.org/10.5220/0010448300300040>
- Devmedia (2016). Agile Development: XP e Scrum em uma Abordagem Comparativa. Disponível em: <http://www.devmedia.com.br/agile-development-xp-e-scrum-em-uma-abordagem-comparativa/30808>.
- Elghariani, K., & Kama, N. (2016). Review on Agile requirements engineering challenges. In *2016 3rd International conference on computer and information sciences (ICCOINS)* (pp. 507-512). IEEE. <https://doi.org/10.1109/ICCOINS.2016.7783267>
- El-Najar, T., Ahmad, I., & Alkandari, M. (2019). Easycomm: A Framework and Tool to Solve Client Communication Problem in Agile Development. *IAENG International Journal of Computer Science*, 46(1).
- Fernandez, D. J.; & Fernandez, J. D. (2008). Agile project management-Agilism versus traditional approaches. *Journal of Computer Information Systems*, 49(2), 10-17. <https://doi.org/10.1080/08874417.2009.11646044>

- Ferreira, C., & Cohen, J. (2008, October). Agile systems development and stakeholder satisfaction: a South African empirical study. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology* (pp. 48-55).
- Fitzgerald, B.; Hartnett, G.; & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200-213. <https://doi.org/10.1057/palgrave.ejis.3000605>
- Ghani, I., Bello, M., & Bagiwa, I. L. (2015). A survey-based analysis of agile adoption on performances of IT organizations. *Journal of Internet Computing and Services*, 16(5), 87-92. <http://dx.doi.org/10.7472/jksii.2015.16.5.87>
- Goode, W.; & Hatt, P. (1975). *Métodos em pesquisa social*. 5ª ed. São Paulo: Ed. Nacional.
- Grapenthin, S.; Poggel, S.; Book, M.; & Gruhn, V. (2015). Improving task breakdown comprehensiveness in agile projects with an Interaction Room. *Information and Software Technology*, 67, 254-264. <https://doi.org/10.1016/j.infsof.2015.07.008>
- Guo, C., Kulviwat, S., Zhu, J., & Wang, Y. J. (2019). Competing in an emerging market: antecedents and consequences of market orientation and the role of environmental factors. *Journal of Strategic Marketing*, 27(3), 248-267. <https://doi.org/10.1080/0965254X.2017.1411386>
- Hannay, J. E., Benestad, H. C., & Strand, K. (2017). Benefit points: the best part of the story. *IEEE Software*, 34(3), 73-85. DOI:10.1109/MS.2017.66
- Highsmith, J. (2004). *Agile Project Management: creating innovative products*. Boston: Addison-Wesley.
- Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, 51, 915-929. <https://doi.org/10.1016/j.chb.2014.10.046>
- Kevin Sungkur, R.; & Ramasawmy, M. (2014). Knowledge4Scrum, a novel knowledge management tool for agile distributed teams. *Vine*, 44(3), 394-419.
- Kurniawan, R., Budiastuti, D., Hamsal, M., & Kosasih, W. (2020). The impact of balanced agile project management on firm performance: the mediating role of market orientation and strategic agility. *Review of International Business and Strategy*. <https://doi.org/10.1108/RIBS-03-2020-0022>
- Lima, A. P. R., & Amaral, D. C. (2022). O que fazer quando não há Product Owner? Um estudo sobre a ausência desse papel. *Revista de Gestão e Projetos (GeP)*, 13(2), 38-65. <https://doi.org/10.5585/gep.v13i2.22209>

- Massari, V. (2014). Gerenciamento Ágil de Projetos. Rio de Janeiro: Brasport.
- Matook, S., & Maruping, L. M. (2014). A competency model for customer representatives in agile software development projects. *MIS Quarterly Executive*, 13(2).
- Mchugh, O.; Conboy, K.; & Lang, M. (2012). Agile practices: The impact on trust in software project teams. *Ieee Software*, 29(3), 71-76. <https://doi.org/10.1109/MS.2011.118>
- Moe, N. B.; Dingsøyr, T.; & Dybå, T. (2009). Overcoming barriers to self-management in software teams. *IEEE software*, 26(6), 20-26.
- Nerur, S.; Mahapatra, R.; & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
- Niemi-Grundström, M. (2014). Developing, evaluating and managing library with agile methods. *Library Management*, 35(6/7), 481-485. <https://doi.org/10.1108/LM-02-2014-0022>
- Oliveira, E.; & Lima, R. (2012). Estado da arte sobre o uso do *Scrum* em ambientes de desenvolvimento distribuído de software. *Revista de Sistemas e Computação-RSC*, 1(2).
- Paasivaara, M.; Durasiewicz, S.; & Lassenius, C. (2008). Using scrum in a globally distributed project: a case study. *Software Process: Improvement and Practice*, 13(6), 527-544. <https://doi.org/10.1002/spip.402>
- Parker, D. W., Kunde, R., & Zeppetella, L. (2017). Exploring communication in project-based interventions. *International Journal of Productivity and Performance Management*, 66(2,), 146-179. <https://doi.org/10.1108/IJPPM-07-2015-0099>
- Pearce, C. L. (2004). The future of leadership: Combining vertical and shared leadership to transform knowledge work. *The Academy of Management Executive*, 18(1), 47-57. <https://doi.org/10.5465/AME.2004.12690298>
- Pichler, R. (2010). Agile product management with scrum: creating products that customers love. Addison-Wesley Professional.
- Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P.; & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337. <https://doi.org/10.1007/s10664-008-9065-9>
- Rajan, E. R., & Santhosh, V. A. (2021). Adoption of Agile Methodology for improving it project performAnce. *Serbian Journal of Management*, 16(2). DOI:10.5937/sjm16-26854
- Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20(5), 449-480. <https://doi.org/10.1111/j.1365-2575.2007.00259.x>

- Recker, J., Holten, R., Hummel, M., & Rosenkranz, C. (2017). How agile practices impact customer responsiveness and development success: A field study. *Project Management Journal*, 48(2), 99-121. <https://doi.org/10.1177/875697281704800208>
- Richardson, R. J. (2017). Pesquisa social: métodos e técnicas. 4. ed. São Paulo: Atlas.
- Schwaber, K.; & Sutherland, J. (2013). Guia do Scrum™. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>, 2022.
- Schwaber, K. (2004). Agile project management with Scrum. Microsoft press.
- Schwaber K, & Beedle M. (2010). Agile Software Development with Scrum. Prentice Hall: New York.
- Seabra, C.; & Almeida, A. M. (2015). Project management on Multimedia Projects: preliminary results on communication, interaction and team work dynamics. *Procedia Computer Science*, 64, 816-823. <https://doi.org/10.1016/j.procs.2015.08.633>
- Selim Senyurt, B., Basri Yuksel, H., Asilsamanci, E., & Ozkil, A. (2021, August). Turkuaz: The Modernization of an ERP Application with Agile Methodology in The Automotive Sector in Turkey. In *2021 The 11th International Conference on Information Communication and Management* (pp. 1-10).
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040-1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Shalloway, A.; Beaver, G.; & Trott, J. (2009). Lean-agile software development: achieving enterprise agility. Pearson Education.
- Sommerville, I. (2007). Engenharia de Software. 8ª ed. São Paulo: Pearson.
- Sutherland, J., & Schwaber, K. (2014). The Scrum Papers. *Nuts, Bolts and Origins of an Agile Process*.
- Sverrisdottir, H.; Ingason, H.; & Jonasson, H. (2014). The role of the product owner in scrum-comparison between theory and practices. *Procedia-Social and Behavioral Sciences*. 119, 257-267. <https://doi.org/10.1016/j.sbspro.2014.03.030>
- Thorgren, S., & Caiman, E. (2019). The role of psychological safety in implementing agile methods across cultures. *Research-Technology Management*, 62(2), 31-39. <https://doi.org/10.1080/08956308.2019.1563436>
- Tsai, W. L., Chen, C. Y., & Chen, C. S. (2018). Snowman: Agile development method with institutionalized communication and documentation for capstone projects. *Asia Pacific Management Review*, 23(1), 12-19. <https://doi.org/10.1016/j.apmr.2017.01.002>

- Tseng, M.; & Du, X. (1998). Design by customers for mass customization products. *CIRP Annals-Manufacturing Technology*, 7(1), 103-106. DOI: 10.1007/978-0-85729-224-7_41
- Van Waardenburg, G.; & Van Vliet, H. (2013). When agile meets the enterprise. *Information and software technology*, 55(12), 2154-2171. <https://doi.org/10.1016/j.infsof.2013.07.012>
- Versionone (2020) 14th-Annual-State-of-Agile-Development-Survey. Disponível em: <https://stateofagile.com/home#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>.
- Wang, X., Conboy, K., & Pikkarainen, M. (2012). Assimilation of agile practices in use. *Information Systems Journal*, 22(6), 435-455. <https://doi.org/10.1111/j.1365-2575.2011.00393.x>
- West, D.; Grant, T.; Gerish, M.; & D'Silva, D., (2010). Agile development: mainstream adoption has changed agility. *Forrester Research*, 2(1), 41.
- Witell, L., Kowalkowski, C., Perks, H., Raddats, C., Schwabe, M., Benedettini, O., & Burton, J. (2020). Characterizing customer experience management in business markets. *Journal of Business Research*, 116, 420-430. <https://doi.org/10.1016/j.jbusres.2019.08.050>
- Yin, R. K. (2005). *Estudo de caso: planejamento e métodos*. 3ª ed. Porto Alegre: Bookman.
- Younas, M., Jawawi, D. N., Ghani, I., Fries, T., & Kazmi, R. (2018). Agile development in the cloud computing environment: A systematic review. *Information and Software Technology*, 103, 142-158. <https://doi.org/10.1016/j.infsof.2018.06.014>