

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

ARTHUR MARQUEZE BATISTA

**Estudo de caso: Aplicabilidade da metodologia ágil *Scrum* no desenvolvimento
de um produto para o mercado imobiliário**

FLORIANÓPOLIS, 2022

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECATRÔNICA**

ARTHUR MARQUEZE BATISTA

**Estudo de caso: Aplicabilidade da metodologia ágil *Scrum* no desenvolvimento
de um produto para o mercado imobiliário**

Trabalho de Conclusão de Curso submetido
ao Instituto Federal de Educação, Ciência
e Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Engenheiro em Mecatrônica

Orientador:
Prof. Delcino Picinin Júnior, Doutor

FLORIANÓPOLIS, 2022

Ficha de identificação da obra elaborada pelo autor.

Batista, Arthur

Estudo de caso: Aplicabilidade da metodologia ágil

Scrum no desenvolvimento de um produto para o mercado imobiliário

/ Arthur Batista; orientação de Delcino Picinin

Junior. - Florianópolis, SC, 2022.

91 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica. Departamento Acadêmico de Metal Mecânica.

Inclui Referências.

1. Scrum. 2. Imobiliárias. 3. Processos. 4. Transformação digital. 5. Metodologias ágeis. I. Picinin Junior, Delcino. II. Instituto Federal de Santa Catarina. III. Estudo de caso: Aplicabilidade da metodologia ágil Scrum no desenvolvimento de um produto para o mercado imobiliário.

ESTUDO DE CASO: APLICABILIDADE DA METODOLOGIA ÁGIL SCRUM NO DESENVOLVIMENTO DE UM PRODUTO PARA STARTUP

ARTHUR MARQUEZE BATISTA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 28 de fevereiro de 2022.

Banca Examinadora:

Delcino Picinin Júnior, Prof. Dr.

Widomar Pereira Carpes Júnior, Prof. Dr. Eng.
IFSC

Maurício Edgar Stivanello, Prof. Dr.
IFSC

Dedico este trabalho a todas as pessoas
que passaram pela minha vida e
contribuíram com minha jornada

Stay Hungry, Stay Foolish - Steve Jobs

RESUMO

O presente trabalho é referente à aplicabilidade da metodologia ágil Scrum na criação de um protótipo que visa à automação de processos imobiliários, através de um produto criado pela startup Nia, do grupo Brognoli. O objetivo deste produto, que é uma solução digital, é uniformizar os processos relacionados aos serviços imobiliários, e utilizar diversas tecnologias para automatizá-los. Este produto nasceu para ajudar imobiliárias de pequeno e médio porte, que estão atrasadas na transformação digital, dando mais controle à jornada dos imóveis, centralizando as informações e melhorando a comunicação com todas as partes dos processos, como locatários, compradores e proprietários. Portanto, utilizando o know-how da startup no mercado imobiliário, foi possível idealizar um produto padrão que pode ser vendido para diversas imobiliárias, sem necessidade de customização e com um tempo de *setup* baixo.

Devido a complexidade de criar um produto digital, é importante que o desenvolvimento seja feito de forma transparente e ágil, onde todos os envolvidos compreendam o produto como um todo. Além disso, é comum nesses projetos a necessidade de atualizações rápidas e adições de novos recursos de forma constante. Portanto, é necessário o uso de uma metodologia de desenvolvimento de software, com o propósito de padronizar os métodos e notações, que culminam em um processo mais eficiente. Então, o objetivo deste trabalho é entender as diversas ferramentas de desenvolvimento de software que existem e avaliar os resultados da aplicação do Scrum como metodologia para gerenciamento de projetos, aplicado ao produto da startup Nia.

Palavras-chave: Scrum, Imobiliária, transformação digital, Processos, Metodologia ágil

ABSTRACT

The present work refers to the applicability of the Scrum agile methodology in the creation of a prototype that aims to automate real estate processes, through a product created by startup Nia, from the Brognoli group. The objective of this product, which is a digital solution, is to standardize processes related to real estate services, and use different technologies to automate them. This product was born to help small and medium-sized real estate companies, which are lagging behind in the digital transformation, giving more control to the real estate journey, centralizing information and improving communication with all parts of the processes, such as tenants, buyers and owners. Therefore, using the startup's know-how in the real estate market, it was possible to devise a standard product that can be sold to several real estate agencies, without the need for customization and with a low setup time.

Due to the complexity of creating a digital product, it is important that the development is done in a transparent and agile way, where everyone involved understands the product as a whole. In addition, it is common in these projects the need for fast updates and additions of new features on a constant basis. Therefore, it is necessary to use a software development methodology, with the purpose of standardizing the methods and notations, which culminate in a more efficient process. So, the objective of this work is to understand the different software development tools that exist and evaluate the results of applying Scrum as a methodology for project management, applied to the product of startup Nia.

Keywords: Scrum, Real Estate, Digital Transformation, Processes, Agile methodology

LISTA DE FIGURAS

Figura 1 – Transição Organizacional	17
Figura 2 – Modelo cascata	21
Figura 3 – Ciclo de vida do <i>Scrum</i>	28
Figura 4 – Visão do <i>Bitrix</i> - Desenvolvimento	36
Figura 5 – Visão do <i>Bitrix</i> - Chamados	37
Figura 6 – Visão do <i>Bitrix</i> - Cartões	38
Figura 7 – Visão do <i>Bitrix</i> - Contatos	39
Figura 8 – Visão do <i>Jira</i> - <i>Backlog</i>	43
Figura 9 – Visão do <i>Jira</i> - <i>Sprint</i>	44
Figura 10 – Visão do <i>Jira</i> - Atividades	45
Figura 11 – Tempo que o colaborador trabalha com desenvolvimento de software	49
Figura 12 – No início do processo de desenvolvimento, é elaborado uma lista de atividades que serão desenvolvidas	50
Figura 13 – Toda a equipe participa da estimativa das atividades a serem desen- volvidas	51
Figura 14 – O software utilizado, "Jira", ajuda na organização e no acompanha- mento do desenvolvimento	52
Figura 15 – A estimativa feita para a equipe é condizente com o que é possível de produzir no tempo estipulado	53
Figura 16 – As atividades do <i>Sprint Backlog</i> são detalhados, de forma que todos os membros consigam entender a atividade a ser feita"	54
Figura 17 – Existe uma pessoa na equipe responsável por checar se o <i>Scrum</i> está sendo corretamente aplicado	56
Figura 18 – Existe uma pessoa na equipe responsável por gerenciar os backlogs	56
Figura 19 – Os impedimentos levantados nessas reuniões são rapidamente re- solvidos	58
Figura 20 – Como o colaborador avalia a flexibilidade durante o projeto?	60
Figura 21 – Como o colaborador avalia a eficiência durante o projeto?	61
Figura 22 – Como o colaborador avalia a própria satisfação durante o projeto? .	62
Figura 23 – Como o colaborador avalia o <i>Scrum</i> como metodologia de desenvol- vimento de projeto?	62

LISTA DE TABELAS

Tabela 1 – Comparação entre metodologias ágeis	24
Tabela 2 – Comparação entre metodologias ágeis e modelo tradicional	25
Tabela 3 – Quais metodologias de gestão são conhecidas de acordo com experiência	49
Tabela 4 – Respostas dos colaboradores para a questão 5 de acordo com a experiência	51
Tabela 5 – Respostas dos colaboradores para a questão 9 de acordo com a experiência	54
Tabela 6 – Respostas dos colaboradores para a questão 11 de acordo com a experiência	55
Tabela 7 – Respostas dos colaboradores para a questão 14 de acordo com a experiência	57
Tabela 8 – Respostas dos colaboradores para a questão 16 de acordo com a experiência	58
Tabela 9 – Respostas dos colaboradores para a questão 20 de acordo com a experiência	59
Tabela 10 – Respostas dos colaboradores para a questão 24 de acordo com a experiência	60
Tabela 11 – Colaborador 1	84
Tabela 12 – Colaborador 2	85
Tabela 13 – Colaborador 3	86
Tabela 14 – Colaborador 4	87
Tabela 15 – Colaborador 5	88
Tabela 16 – Colaborador 6	89
Tabela 17 – Colaborador 7	90
Tabela 18 – Colaborador 8	91

LISTA DE ABREVIATURAS E SIGLAS

CRM	<i>Customer Relationship Management</i>
IA	Inteligência Artificial
IFSC	Instituto Federal de Santa Catarina
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
RPA	<i>Robot Process Automation</i> (Automação de processos com Robô)
XP	<i>Extreme Programming</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Formulação do problema	14
1.2	Objetivo Geral	14
1.2.1	Objetivos Específicos	14
1.3	Estrutura do Texto	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Gerenciamento de Projetos	16
2.2	Principais Metodologias	18
2.3	Projetos de Software	19
2.3.1	Metodologia Cascata	20
2.4	Metodologia Ágil	22
2.4.1	Origem	22
2.4.2	Principais metodologias ágeis	23
2.4.3	Comparação com o método tradicional	24
2.5	Scrum	25
2.5.1	Times do <i>Scrum</i>	27
2.5.2	Eventos <i>Scrum</i>	28
2.5.3	Artefatos <i>Scrum</i>	30
2.6	Trabalhos correlatos	30
2.6.1	A utilização da metodologia <i>Scrum</i> sob a percepção da equipe de desenvolvimento em uma empresa privada de <i>software</i> : Um estudo de caso	30
2.6.2	Desenvolvimento de <i>software</i> : Aplicabilidade de metodologias ágeis no mercado de <i>software</i> em Vitória da Conquista - BA	31
2.6.3	Aplicabilidade das práticas do <i>framework Scrum</i> na gestão dos projetos de desenvolvimento de <i>software</i> do exército brasileiro	31
3	ESTUDO REALIZADO	33
3.1	Apresentação do Negócio	33
3.2	Produto	34
3.3	Tecnologias utilizadas	35
3.4	Aplicação do <i>Scrum</i>	40
3.4.1	Pessoas e responsabilidades	40
3.4.2	Atividades e <i>backlog</i>	40
3.4.3	Eventos	42
3.4.4	Software utilizado	42
4	RESULTADOS E DISCUSSÕES	46
4.1	Procedimentos de coleta dos dados	46
4.2	Análise dos dados	48
4.2.1	Planejamento	50
4.2.2	Artefatos	52
4.2.3	Papéis e Responsabilidades	54
4.2.4	Eventos	57
4.2.5	Pessoas	59
4.2.6	Finalização	62
5	CONCLUSÃO	64

REFERÊNCIAS	65
APÊNDICES	67
APÊNDICE A – QUESTIONÁRIO E RESPOSTAS	68
APÊNDICE B – RESPOSTAS INDIVIDUAIS	84

1 INTRODUÇÃO

Atualmente, o mercado imobiliário utiliza da informatização de seus processos, porém há ainda uma grande quantidade de processos que são realizados de forma manual. Além disso, há uma perda significativa de informações relativas aos clientes e aos imóveis, assim como dos processos já realizados, gerando gastos desnecessários e experiências negativas dos clientes.

Visando se manter relevante no setor, as imobiliárias vêm procurando cada vez mais soluções para a transformação digital. Esses produtos vão de softwares de gestão, como CRM's (*Customer Relationship Management*), até plataformas de divulgação de imóveis. Porém, a transformação digital é mais do que simplesmente inserir tecnologias na empresa. Segundo o livro "A Jornada da Transformação Digital"(LOBÃO LUIS; ZILLI, 2020), essas mudanças devem ser vistas como um processo, sustentado por 3 pilares: Pessoas, Processos e Tecnologia. Logo, é preciso que as imobiliárias ajustem seus processos de acordo com os gargalos gerados pelas pessoas, para que a tecnologia possa ser inserida.

Então, diversas empresas de tecnologia da informação e *startups*, vêm inovando em novos produtos digitais para o mercado imobiliário, visando resolver os maiores problemas do setor. Porém, o mercado está cada vez mais competitivo e as companhias precisam ser ágeis na criação de valor ao público alvo. E devido a grande quantidade de variáveis que permeiam o ambiente de desenvolvimento de software, é importante que seja feito um gerenciamento de projeto eficaz e rápido, mantendo o nível de qualidade. Outro fator que corrobora com isso, é que devido a alta rotatividade de produtos e a necessidade de adaptação, os *softwares* devem ser fáceis de serem desenvolvidos e mantidos.

Logo, o autor compreendendo as dificuldades do setor imobiliário, dos desafios do desenvolvimento de um produto digital e da importância na escolha da metodologia mais adequada, almeja tornar possível realizar um estudo das diversas metodologias de desenvolvimento de projetos existentes e avaliar a utilização da ferramenta *Scrum* no contexto da *startup* Nia. Considerando as condições descritas, a metodologia *Scrum* é uma das mais recomendadas para produtos digitais que precisam entregar valor de forma rápida, além de já prever recursos para que a comunicação entre os colaboradores seja feita de forma transparente, minimizando as chances de retrabalho e impedimentos. Como de praxe, nem toda aplicação e metodologia é realizada da forma ideal, e por isso é importante a flexibilidade no uso dessas ferramentas, para que seja possível a adaptação de acordo com a realidade que cada aplicante se encontra inserido.

1.1 Formulação do problema

Ao longo dos anos, as metodologias de gerenciamento de projetos foram se criando e adaptando para a realidade vivida, visando sempre a organização e eficiência nas empresas. Considerando que o mundo atual se encontra em um momento onde os produtos digitais, os softwares, estão permeando o dia a dia das pessoas, é visto que as empresas desenvolvedoras desses sistemas estão cada vez mais engajadas em entregar seus produtos com mais eficiência e agilidade. Logo, é importante estudar e avaliar as diferentes metodologias de gerenciamento de projetos para entender a aplicabilidade nos diferentes cenários vividos pelas empresas, e efetivamente conseguir medir com base em estudos e pesquisas embasadas os benefícios e malefícios da aplicação das metodologias nestas empresas de desenvolvimento de sistemas. Então, o objetivo deste estudo é avaliar o *framework Scrum*, uma das mais conhecidas metodologias de gerenciamento de projetos existentes, que cresceu como alternativa da gestão tradicional de projetos devido aos recorrentes problemas vivenciados durante o processo de desenvolvimento de sistemas, que dificilmente eram entregados nos prazos corretos e de acordo com os requisitos do cliente. O estudo avaliará essa metodologia no contexto de uma empresa de tecnologia que busca eficiência, transparência e flexibilidade no desenvolvimento de um novo produto.

1.2 Objetivo Geral

O objetivo desse trabalho é realizar um estudo de caso da utilização da metodologia ágil *Scrum* no desenvolvimento de um produto para o mercado imobiliário.

1.2.1 Objetivos Específicos

Os objetivos específicos para esse trabalho são:

- Avaliar se as boas práticas do *Scrum* estão sendo seguidas;
- Analisar se os papéis e responsabilidades *Scrum* estão sendo respeitados;
- Verificar se o planejamento está sendo efetivo e transparente;
- Avaliar a efetividade da priorização;
- Avaliar o desempenho e satisfação com o uso do *Scrum* por parte do time (equipe de desenvolvimento).

1.3 Estrutura do Texto

O presente trabalho conta com 5 capítulos, onde o primeiro capítulo faz uma introdução sobre o estudo a ser feito com uma apresentação geral do estudo e

sua devida importância, além de esclarecer o objetivo do estudo. O segundo capítulo descreve toda a base teórica do trabalho, abordando os principais conceitos relativos à gestão de projetos, projetos de software, metodologias ágeis e *Scrum*, bem como sua relevância para o desenvolvimento de software. O terceiro capítulo traz a apresentação do estudo de caso, bem como os detalhes sobre o produto e a empresa envolvida. O capítulo quatro descreve a metodologia de coleta dos dados, abordando os resultados da pesquisa e sua análise. O quinto e último capítulo apresenta a conclusão do trabalho, apresentando as considerações finais da pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo, serão abordados os conceitos e fundamentos que sustentam a pesquisa e o estudo de caso. levando a uma melhor compreensão sobre o gerenciamento de projetos, as metodologias tradicionais de desenvolvimento de *software*, Metodologias Ágeis e *framework Scrum*.

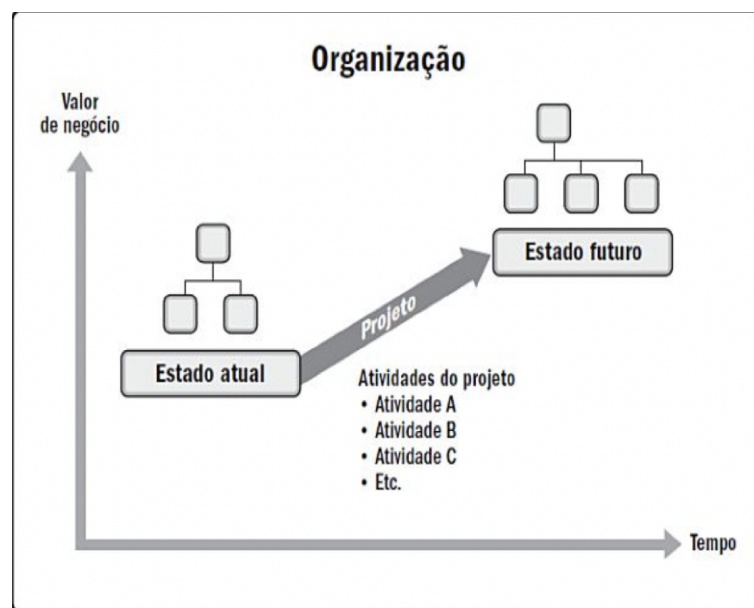
2.1 Gerenciamento de Projetos

De acordo com o PMBOK (PMI, 2013), um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único. Diversos projetos foram executados ao longo da história, e mesmo com divergências temáticas, temporárias ou financeiras, todos possuem as seguintes características em comum (KEELING, 2017):

- são empreendimentos independentes;
- possuem propósito e objetivos específicos;
- têm duração limitada, com início e fim bem definidos;
- entregam um resultado único;
- recursos próprios (incluindo financeiros e humanos);
- administração e estrutura administrativa própria.

Outro ponto importante sobre o projeto são os interessados (*stakeholders*, em inglês). Essas são as pessoas que serão afetadas (positiva ou negativamente) pela execução ou resultado do projeto. Portanto, é importante que os objetivos desses *stakeholders* sejam claros, para que as entregas sejam feitas de acordo com as expectativas desse grupo de pessoas, pois o não cumprimento de requisitos impostos por eles pode impactar diretamente no prazo, escopo e custo do projeto. Entre os interessados, podem haver: financiadores, avalistas, proprietários, fornecedor, empreiteiros, entre outros.

Os projetos podem servir para realizar uma transição de um estado organizacional dentro de uma corporação. Na figura 1 é mostrado de forma gráfica, representada pelos eixos do Tempo e Valor de Negócio, o impacto causado por um projeto, que após uma série de atividades pré-definidas ao longo de um tempo definido, pode mudar a estrutura de uma organização, resultando em um benefício às partes interessadas. Esse ganho pode ser tangível, como em ativos monetários ou acionários, porém, pode também ser intangível, aumentando o reconhecimento da marca e sua reputação.

Figura 1 – Transição Organizacional

Fonte: (PMI, 2013).

É possível compreender então, que o projeto necessita de uma série de esforços coordenados com o objetivo de entregar os resultados dentro de um prazo definido, e por isso, surge a necessidade da gestão desses projetos. Segundo o PMI (PMI, 2013), gerenciamento de projetos pode ser definido como "a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades de um projeto a fim de atender os seus requisitos. Harold Kerzner (KERZNER, 2020) defende que o gerenciamento de projetos deixou de ser um conjunto de processos recomendável, para uma metodologia tida como mandatória para a sobrevivência das empresas, visto que em muitos momentos um gerente de projetos atua sobre decisões de negócio, direcionando e viabilizando um caminho sustentável e eficiente para as organizações criarem cada vez mais valor. Ele (KERZNER, 2020), também define que existem 6 forças motrizes que levam os executivos a reconhecer a necessidade da gestão de projetos:

- Projetos de capital;
- Expectativas do cliente;
- Competitividade;
- Compreensão por parte dos executivos;
- Desenvolvimento de novos projetos;
- Eficácia e eficiência.

O gerenciamento de projetos significa então, a coordenação de todas as fases do projeto, desde o planejamento, execução, controle dos recursos e entregas e conclusão cumprindo os objetivos estabelecidos (LEAL, 2008). Compreendendo as diferenças existentes entre os projetos, cabe ao gestor de projetos definir quais são as melhores práticas a serem utilizadas, pois a complexidade e a dimensão do trabalho são fatores decisivos para uma padronização e estruturação do processo de gestão dos projetos. A adoção das metodologias corretas e de uma formalização do processo de gestão de projetos por uma organização, juntamente do forte comprometimento com o seguimento do fluxo, resulta em entregas de qualidade para os interessados, aumentando a chance de sucesso do negócio.

2.2 Principais Metodologias

O gerenciamento de projetos não é um tema atual, pois segundo Kerzner (KERZNER, 2020), os primeiros relatos da ocupação de um profissional que fazia o papel de gestor de projetos datam da década de 40, onde o conceito utilizado era o gerenciamento "por cima da cerca", onde o gestor de área, após finalizado a sua parte, passava as entregas para a próxima área. Se o projeto falhasse, a responsabilidade era da área que estava atuando naquele momento. No fim da década de 50, durante a Guerra Fria, os Estados Unidos necessitavam de uma padronização para o gerenciamento dos projetos aeroespaciais e de defesa, formalizando então alguns processos com o objetivo de realizar um controle de custos e de planejamento. A partir da década de 60, criou-se então entendimento mais difundido sobre o que era o gerenciamento de projetos, e foi o momento onde a maioria das empresas que realizavam grandes projetos começou a realizar um gerenciamento informal desses projetos, onde o "gerente" era visto como um monitor, não podendo tomar grandes decisões e com pouca autoridade. Nessa década, foram fundadas duas organizações com o objetivo de difundir informações e promover as melhores práticas dessa área, o IPMA (Associação Internacional de Gestão de projetos) em 1965, e o PMI, (Instituto de Gestão de Projetos) em 1969. Na década de 80 e 90, foi necessário por parte das empresas realizar uma reestruturação interna com o objetivo de melhor gerenciar os projetos. Nessa época, observou-se um aumento do uso dos computadores, que trouxe conectividade às diversas áreas da empresa, facilitando a troca de informações entre elas. Com o aumento do uso da internet e expansão da tecnologia da informação, a partir dos anos 2000, foram criadas e aperfeiçoadas as metodologias ágeis, que serão abordadas a frente.

Como citado previamente, o PMI é uma associação internacional com o objetivo de difundir boas práticas de gestão de projetos, procurando também evoluir os profissionais da área oferecendo treinamentos e certificações reconhecidas.

Com isso, o PMI concebeu o Guia PMBOK® (Corpo de Conhecimento em Gerencia de Projetos), visando descrever os conceitos e padronizar processos de gerência, apresentando um conjunto de conhecimentos genéricos que podem ser aplicados em diversas áreas. Essa característica abrangente do guia, permite que os gerentes e empresas utilizem as definições e ferramentas apresentadas de acordo com a realidade e necessidade da organização. O PMBOK® (PMI, 2013) se define como iterativo e incremental, e não algo definitivo e imutável, podendo sofrer alterações ao longo das edições de acordo com as melhores práticas observadas. O Guia PMBOK® não é considerado uma metodologia, mesmo que busque sugerir conceitos utilizados em todas as fases do projeto.

Esse guia é dividido em 10 áreas do conhecimento: Integração, Escopo, Cronograma, Custos, Qualidade, Recursos, Comunicações, Riscos, Aquisições e Partes Interessadas. Cada uma dessas áreas é mostrada como um conjunto de processos com escopo definido, com as entradas e saídas geradas em cada fluxo e ferramentas a serem utilizadas. Cada área é interligada com as outras, criando um mecanismo único e funcional.

2.3 Projetos de Software

Sommerville (SOMMERVILLE, 2011) , define um projeto de *software* como uma descrição de estrutura de *software* a ser implementada, dos dados que são parte do sistema, das interfaces entre os componentes do sistema e, algumas vezes, dos algoritmos utilizados. Santiago (SANTIAGO, 2012), em seu estudo sobre engenharia de *software*, reporta que independente da abordagem utilizada para gerenciar um projeto de desenvolvimento de *software*, todo projeto passa pelas seguintes etapas: análise de negócio, de requisitos projeto técnico, construção e validação.

Porém, como foi mostrado no livro "Gestão de projetos: Melhores práticas"(KERZNER, 2020), as metodologias de projetos evoluíram e trouxeram a possibilidade de customização no uso das ferramentas de acordo com a área que o projeto está compreendido. Por sua vez, os projetos de *software* possuem alguns atributos que os diferenciam dos projetos de outras áreas (OLIVEIRA, 2014):

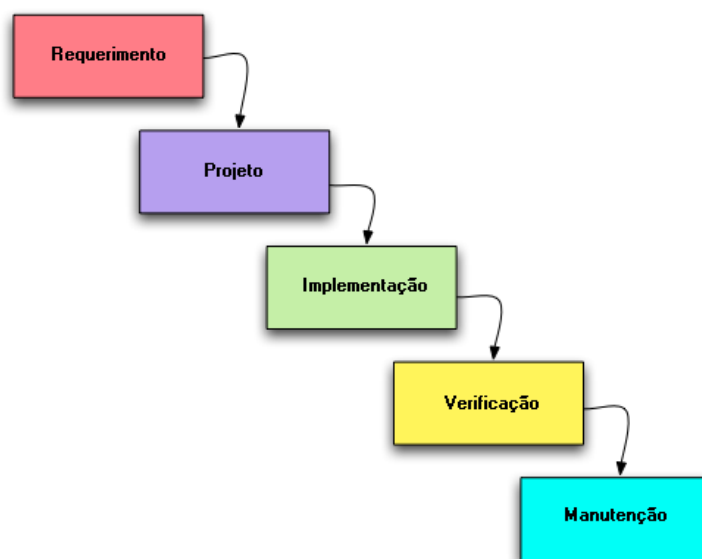
- a estrutura organizacional da empresa;
- produto não é visível de imediato;
- complexidade do produto devido à sofisticação dos sistemas e dos usuários;
- adaptação do produto de acordo com a necessidade do cliente;
- facilidade que o produto pode ser modificado.

Então, com o aumento exponencial da indústria de tecnologia da informação, com essas características que diferenciam um projeto de *software* das demais áreas, tornou-se necessário buscar e estruturar modelos ou metodologias que sejam compatíveis com a realidade vivida nesse projetos. Dessa necessidade, foram criadas as metodologias tradicionais, que foram adaptadas de outras áreas, com o objetivo de estruturar os projetos de *software* com base em um processo definido (SANTIAGO, 2012). Porém, essas metodologias surgiram em um cenário distinto do atual, onde o custo de um projeto de *software* era muito elevado devido ao acesso limitado a máquinas como computadores (PRESSMAN, 2011).

As metodologias tradicionais são caracterizadas pela separação bem definida das fases do projeto, que consistem em (SANTIAGO, 2012): levantamento de requisitos, análise, desenho, implementação, testes e implantação. Cada uma das fases só começa quando a anterior estiver finalizada. Além dessa característica, outro atributo marcante é a definição e planejamento do escopo do projeto, estruturando todo o sistema, funcionalidades e características, havendo um grande trabalho de documentação. O responsável do projeto também deve tentar prever todos os pontos possíveis de serem problemáticos, para evitar retrabalho e atrasos na entrega.

2.3.1 Metodologia Cascata

A metodologia cascata, ou *Waterfall*, em inglês, consiste de um modelo de gerenciamento de desenvolvimento de *software* que surgiu na década de 70 com etapas lineares e sequenciais, que ao final, ao cumprir todas as fases pré definidas, resultarão em um *software* pronto. A figura a seguir ilustra a lógica do modelo, onde a saída de uma etapa consiste no início da próxima fase, seguindo uma sequência:

Figura 2 – Modelo cascata

Fonte: (WIKIPEDIA, 2021)

As etapas e objetivos do modelo são (PRESSMAN, 2011):

- **Requerimentos:** Detalhamento das funcionalidades e lógica do sistema, com previsão de alocação de recurso. Definição do escopo do projeto por meio de documentações validadas junto do cliente;
- **Projeto (Design):** Traduz as exigências documentadas na primeira fase em atributos, concentrados em estrutura de dados, arquitetura do *software*, detalhes procedimentais e caracterização de interface;
- **Implementação:** O desenvolvimento propriamente dito, feito pelo time de desenvolvimento;
- **Verificação:** Realização de testes para validar se o sistema cumpre os requisitos definidos inicialmente. Garantia de que todas as funcionalidades sejam testadas e, se identificado algum erro, realização de ajustes;
- **Manutenção:** Após a entrega ao cliente, algumas mudanças podem ser solicitadas ou mais erros serão encontrados, portanto, deve ser previsto que os desenvolvedores precisarão realizar manutenções no sistema.

Como citado previamente, com a evolução dos sistemas, o modelo cascata está caindo em desuso, embora seja ainda aplicado em algumas empresas (NOLETO, 2020). Os principais motivos desse desuso se encontra na inflexibilidade do modelo, já que após o planejamento, o projeto não pode ser mudado, e isso consiste na afirmação que o levantamento de requisitos não tenha erros. Além disso, a falta de interação com

o cliente e com o time leva ao retrabalho e ao desenvolvimento de funcionalidades que são inutilizadas na prática, gerando insatisfação por parte de todos os envolvidos.

2.4 Metodologia Ágil

Nesse tópico, será abordado o tema das "Metodologias Ágeis", dando ênfase nos principais métodos e como eles se diferenciam do método tradicional.

2.4.1 Origem

Filho (FILHO, 2008), em seu trabalho, traz da seguinte forma a história sobre a origem das metodologias ágeis:

Durante a evolução dos processos de Engenharia de Software, a indústria se baseou nos métodos tradicionais de desenvolvimento de *software*, que definiram por muitos anos os padrões para criação de *software* nos meios acadêmico e empresarial. Porém, percebendo que a indústria apresentava um grande número de casos de fracasso, alguns líderes experientes adotaram modos de trabalho que se opunham aos principais conceitos das metodologias tradicionais. Aos poucos, foram percebendo que suas formas de trabalho, apesar de não seguirem os padrões no mercado, eram bastante eficientes. Aplicando-as em vários projetos, elas foram aprimoradas e, em alguns casos, chegaram a se transformar em novas metodologias de desenvolvimento de *software*. Essas metodologias passaram a ser chamadas de leves por não utilizarem as formalidades que caracterizavam os processos tradicionais e por evitarem a burocracia imposta pela utilização excessiva de documentos. Com o tempo, algumas delas ganharam destaque nos ambientes empresarial e acadêmico, gerando grandes debates, principalmente relacionados à confiabilidade dos processos e à qualidade do *software*.

Então, no ano de 2001, 17 especialistas em processos de desenvolvimento de *software* se encontraram em Utah, nos Estados Unidos, para discutirem abordagens alternativas ao método tradicional de desenvolvimento de *software*. Nesse encontro, foram discutidos as experiências vividas em diferentes realidades dos envolvidos para conseguir documentar um padrão dos processos de desenvolvimento. Como resultado, foi formulado o Manifesto Ágil, um conjunto de princípios e ideias comuns que podem ser aplicados de diversas formas e em diferentes contextos. Desde a publicação do Manifesto, as metodologias ágeis vêm prevalecendo sobre os métodos tradicionais. Esse Manifesto conta com 12 princípios (FADEL, 2010):

- Prioridade é satisfazer o cliente através da entrega contínua e adiantada de *software* com valor agregado;
- As mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;

- Frequentes entregas do *software* funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
- As pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- Os projetos devem ser construídos em torno de indivíduos motivados. Dando o ambiente e o suporte necessário e confiança para fazer o trabalho;
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
- Software funcionando é a medida primária de progresso;
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- Contínua atenção a excelência técnica e bom design aumentam a agilidade;
- Simplicidade para maximizar, a quantidade de trabalho não realizado é essencial;
- As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis;
- Em intervalos regulares, a equipe deve refletir sobre como tornar-se mais efetiva, e então, ajustar-se de acordo com seu comportamento.

2.4.2 Principais metodologias ágeis

O Manifesto Ágil não é uma metodologia, e sim um conjunto de princípios, e portanto, muitos estudiosos usaram esse manifesto como base para criar suas próprias metodologias. Mesmo que cada uma tenha suas diferenças, elas compartilham de uma série de características para serem consideradas metodologias ágeis. As principais metodologias ágeis atualmente são XP, Lean e *Scrum* (NATHAN-REGIS, 2012), onde as duas primeiras serão descritas brevemente a seguir. O *Scrum*, como é o objeto do estudo de caso, será detalhado na próxima seção.

A metodologia *Extreme Programming* (XP) surgiu como uma tentativa para solucionar os problemas causados pelos ciclos de desenvolvimento longos dos modelos de desenvolvimento tradicional. (FRANCO, 2007). Existem 5 princípios chave da metodologia XP são (JUSTO, 2019): Comunicação, Simplicidade, *Feedback*, Respeito e Coragem. A base prática para a condução do projeto é priorizar as atividades que devem ser desenvolvidas antes juntamente do cliente, e entregar apenas o que foi planejado no início do projeto, por meio de pequenos desenvolvimentos ao longo dos ciclos (que devem ser curtos) para que sejam validados com o cliente. Além disso, por

ser uma metodologia utilizada em projetos de Software, os autores defendem que a programação deve ser feita em dupla e ninguém deve realizar a modificação do código fonte, pois é uma construção conjunta, e deve ter padrões definidos para que todos envolvidos possam compreender o código.

A metodologia Lean surgiu no Japão, mais especificamente na empresa Toyota, e tem como sua base a produção Enxuta. Essa metodologia foi criada para resolver os maiores problemas industriais vividos no cenário pós Segunda Guerra Mundial (POPPENDIECK, 2011). Ao longo dos anos, o Lean pode ser adaptado para a realidade do desenvolvimento de Software, e foi formalizada com os 7 princípios que guiam essa metodologia, como mostra o estudo de Pereira (PEREIRA, 2012): eliminar o desperdício, amplificar o aprendizado, adiar comprometerimentos, entregar rápido, tornar a equipe responsável, construir integridade e visualizar o todo. Pereira (PEREIRA, 2012) cita também que o princípio de eliminar o desperdício foca que este pode ocorrer em vários sentidos, entre eles: dinheiro, recurso, tempo, esforço, espaço.

A tabela abaixo, então, faz uma avaliação simplificada sobre as principais diferenças entre as metodologias ágeis mais utilizadas. É possível compreender que elas têm uma base em comum, o Manifesto Ágil, porém tem diferenças bem marcantes que são vistas quando aplicadas às realidades de empresas. Portanto, uma empresa deve escolher com cautela a melhor metodologia para utilizar de acordo com a sua realidade, dependendo da complexidade do projeto, da quantidade de pessoas envolvidas, e entre outros pontos que devem ser levados em consideração.

Tabela 1 – Comparação entre metodologias ágeis

Metodologia	Práticas
XP	Programação em duplas, priorização, planejamento prévio
Lean	Eliminar desperdícios, maximizar o fluxo processual, evitar retrabalho
Scrum	Entregas incrementais, Empoderar pessoas, <i>backlog</i> , Reuniões constantes

2.4.3 Comparação com o método tradicional

Existem diversas características que diferenciam os modelos tradicionais das metodologias ágeis. Como visto previamente, a metodologia ágil tem o enfoque nas pessoas, enquanto as metodologias tradicionais focam em processos. Essa característica resulta nos modelos tradicionais serem preditivos, isto é, se propõem a analisar previamente todos os problemas que podem decorrer do processo, além de prever o uso de recursos, sejam eles técnicos ou humanos. Já as metodologias ágeis então, são adaptativas, se moldando de acordo com o desenvolvimento do projeto e dos retornos dos projetos, com as adaptações vindas de uma construção conjunta do time.

Outro ponto importante que diferencia esses modelos é a entrega ao cliente. Enquanto nos modelos tradicionais, a entrega é feita de forma integral ao final do projeto, nos modelos ágeis, o produto é entregue de forma parcelada e utilizável, agilizando o retorno do investimento do cliente, e dessa forma, podem surgir alterações e modificações que podem ser alteradas durante o projeto, sem necessidade de retrabalho ou custos adicionais de projeto.

Tabela 2 – Comparação entre metodologias ágeis e modelo tradicional

	Metodologia ágil	Modelo tradicional
Fundamentação	Melhoria contínua	Sistemas específicos
Gestão	Colaboração e transparência	Controle de processo
Desenvolvimento	Entregas incrementais	Cascata
Comunicação	Informal	Formal

2.5 Scrum

Scrum é um *framework* de gerenciamento de projetos dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível (SCHWABER; SUTHERLAND, 2014). Foi criado no início dos anos 90 por Jeff Sutherland e Ken Schwaber, porém foi apenas na década seguinte que teve seu uso difundido, por conta do aumento de desenvolvimento de *software*. Esse *framework* é uma abordagem onde a organização dos negócios é feita de forma enxuta, para que toda a perda no processo seja eliminada ou reduzida. Atualmente, o *Scrum* é o método ágil mais utilizado no mundo corporativo (EMPRESÔMETRO, 2020)

Essa metodologia ágil é comumente utilizada em projetos de desenvolvimento de *software*, porém seu uso não se limita a essa frente, podendo ser utilizado em várias áreas distintas. Além das diferenças de escopo de projeto, o *Scrum* pode ser aplicado em projetos de centenas de milhares de dólares, ou para projetos simples sem orçamento, podendo também variar a quantidade de pessoas envolvidas.

O *Scrum* é fundamentado nas teorias empíricas, afirmando que o conhecimento vem da experiência e de tomada de decisões baseadas em práticas já conhecidas. Para Schwaber e Sutherland (SCHWABER; SUTHERLAND, 2014), existem 3 pilares que sustentam a implementação de controle do processo empírico:

- **Transparência:** Os aspectos importantes do processo devem estar visíveis e acessíveis aos responsáveis e envolvidos. Esse pilar requer um alinhamento constante com uma linguagem comum para todos os participantes;

- Inspeção: Os usuários *Scrum* devem, constantemente, fazer a inspeção do progresso do projeto, com o objetivo de verificar possíveis inconsistências;
- Adaptação: O processo não deve ser imutável e indiscutível, e ele deve ser ajustado caso o inspetor determine que o resultado será inaceitável.

Portanto, o *Scrum* traz uma mudança para a gestão de projetos, com uma série de benefícios comparado com os métodos tradicionais. Segundo Sabbagh em seu livro "*Scrum: Gestão ágil para projetos de Sucesso*", os benefícios do uso do *Scrum* são:

- entregas frequentes de retorno ao investimento dos clientes;
- redução dos riscos do projeto;
- maior qualidade no produto gerado;
- mudanças utilizadas como vantagem competitiva;
- visibilidade do progresso do projeto;
- redução de desperdício;
- aumento da produtividade.

O *Scrum* possibilita que sejam entregues pequenas partes do produto funcionando, proporcionando um rápido retorno de valor para o investimento do cliente. Diferentemente da gestão de projetos tradicional, onde a entrega do produto é feita integralmente no final do projeto, no *Scrum*, cada uma dessas entregas permite que o cliente passe *feedbacks* sobre o produto, para que as mudanças possam ser feitas de forma mais ágil, diminuindo a necessidade de retrabalho. Por conta dessa produção de curtos ciclos e de entregas constantes, o *Scrum* reduz os riscos de negócios de seus projetos, pois ele é construído incrementalmente de acordo com o retorno do cliente, trazendo mais segurança e assertividade no resultado do produto. A progressão do produto, então, é vista de forma mais palpável pelos clientes e pelo time, trazendo visibilidade e transparência no processo de desenvolvimento.

Durante o desenvolvimento, o *feedback* do cliente pode ser negativo, gerando mudanças no andamento do projeto e que podem resultar em alterações do produto final. Porém, no *Scrum*, isso é visto como uma oportunidade de entregar mais valor nas entregas do cliente, descobrindo constantemente mais detalhes sobre o mercado, a empresa e o produto.

O *Scrum* permite também potencializar a produtividade do time, pois são instigados a trabalhar em equipe e ter autonomia na realização do trabalho, com

reuniões constantes com os colaboradores e gerentes capazes de o aconselhar sobre o andamento das atividades e para remoção de impedimentos. Isso gera um senso de construção conjunta entre o time, aumentando sua motivação para entregar resultados de qualidade.

Esse *framework* consiste em times que são associados aos papéis, eventos, artefatos e regras, sendo cada um desses componentes essencial para a aplicação com sucesso do *Scrum*. Esses componentes irão ser explicados a seguir.

2.5.1 Times do *Scrum*

Segundo o Guia *Scrum* (SCHWABER; SUTHERLAND, 2014) o time *Scrum* é auto-organizável e multifuncional, projetado para aperfeiçoar a flexibilidade, criatividade e produtividade. Por ter essas características, não é preciso ter um gerente ou supervisor, pois a própria estrutura do time permite que os mesmos decidam, junto do cliente, como trabalhar para entregar a meta. Além disso, os criadores do *Scrum* também recomendam que as pessoas envolvidas não mudem ao longo do projeto, para ter mais agilidade na entrega final, evitando retrabalhos e alinhamentos adicionais. Para que seja alcançado o melhor resultado para o projeto com o *Scrum*, o time deve ser composto pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*.

- *Product Owner* (PO): é o "dono do produto", e essa função tem a responsabilidade de maximizar o valor do produto para o cliente e para o time de desenvolvimento. O objetivo do dono do produto é definir o projeto, gerenciando os requisitos do produto, realizando mudanças nas priorizações de acordo com o contato constante com o time e com os stakeholders, e com isso, é possível manter a visão do produto constante ao longo do projeto. Ele é responsável também por gerenciar o *backlog* de Produto, garantindo que este seja claro para todos os envolvidos do projeto, e ordenar as atividades do *backlog*, criando um Roadmap do produto, de uma forma que seja possível alcançar as metas estipuladas. Esses artefatos serão explicados de forma mais clara nos próximos tópicos. Segundo Schwaber e Sutherland, para que o PO tenha sucesso, a organização como um todo deve respeitar suas decisões, não sendo permitido que nenhum colaborador possa alterar as configurações de prioridade;
- Time de Desenvolvimento: esse time é composto pelos responsáveis pelo desenvolvimento do produto, e por conhecerem melhor sobre os recursos necessários para desenvolver, são necessários para estipular as durações das atividades e podem estipular as metas de trabalho para cada espaço de tempo. O time deve ser multidisciplinar e ter todas as habilidades necessárias para desenvolver as atividades. Esse time utiliza das priorizações feitas pelo PO para compreender

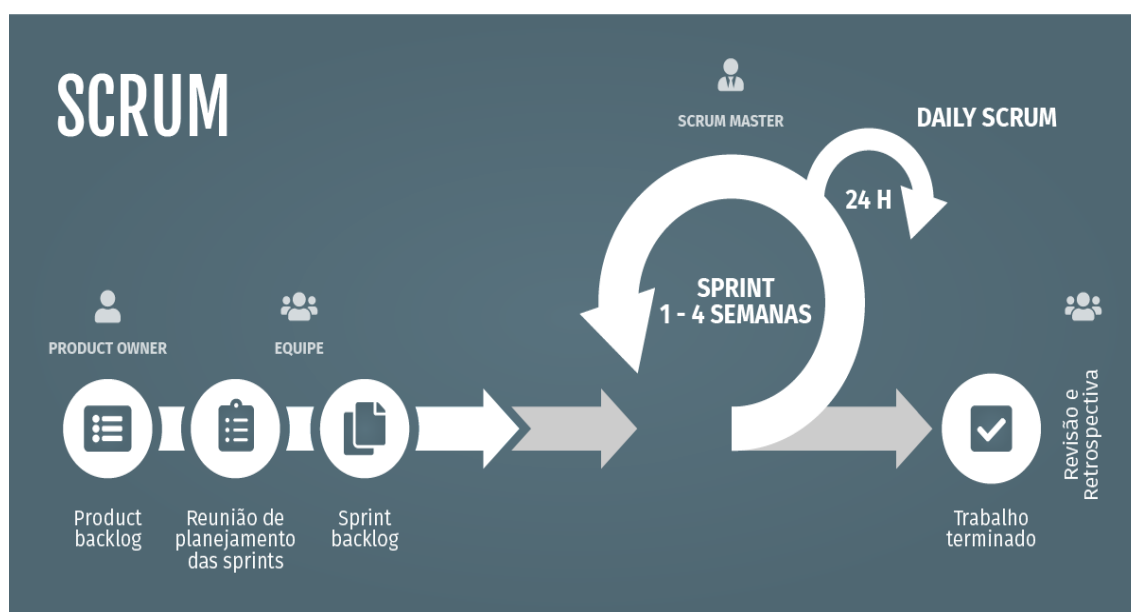
melhor seus objetivos do *Sprint*, e devem ser capazes de compreender o trabalho a ser feito. É importante que o *Time Scrum* esteja alinhado sobre a conclusão das atividades, isto é, o que significa uma atividade estar "Pronta", para que sejam evitados retrabalhos, atrasos e outras problemáticas. O tamanho ideal do time de Desenvolvimento deve ser entre 3 e 9 integrantes, pois dessa forma, o time se mantém ágil. Se forem times muito grandes, muita coordenação e maior necessidade de reuniões de alinhamento são necessários, diminuindo a agilidade da entrega. Porém, se os times forem muito pequenos, o time pode não ter todas as habilidades necessárias e o tempo para que o projeto seja concluído;

- *Scrum Master*: a responsabilidade dessa função é garantir que o *Scrum* seja entendido, e que seja aplicado de acordo com a teoria e com as melhores práticas do *Scrum*. Dessa forma, o *Scrum Master* serve como facilitador não só ao time, mas para toda a organização, removendo impedimentos e aumentando a produtividade da equipe.

2.5.2 Eventos *Scrum*

O objetivo de ter eventos prescritos no *Scrum* é criar uma rotina e minimizar a necessidade de reuniões extras, mantendo a transparência e validação constante. O ciclo de vida do *Scrum* pode ser representado de forma simplificada, mostrando seus eventos, artefatos e responsáveis, como na figura 3.

Figura 3 – Ciclo de vida do *Scrum*



Fonte: (EMPRESôMETRO, 2020)

O coração do *Scrum* é a *Sprint*, como aponta o Guia *Scrum* (SCHWABER; SUTHERLAND, 2014). Cada *Sprint* deve ter de 1 à 4 semanas de duração, e tem

o objetivo de criar uma versão incremental e utilizável do produto, que seja possível entregar para o cliente e obter seus *feedbacks*. A duração não pode ser maior, pois quanto maior o horizonte de tempo, maior a chance de grandes mudanças no projeto, que afetam o dia-a-dia do time. A *Sprint* não deve mudar sua duração durante o seu desenvolvimento, e após a finalização de uma *Sprint*, imediatamente um novo ciclo começa. As *Sprints* são compostas de:

- Reunião de Planejamento de *Sprint*: As atividades a serem desenvolvidas serão planejadas nessa reunião, onde o plano é criado com o trabalho do time. É importante nessa reunião, ficar claro para todos os envolvidos o trabalho a ser feito, e todos estarem alinhados com qual é a definição de pronto para o incremento em questão. Nessa reunião, o time colabora de forma a explicitar quais atividades devem ser feitas para concluir o objetivo do *Sprint* e o esforço necessário para cada atividade que será feita, com o objetivo de maximizar o desenvolvimento e evitar retrabalhos ou falhas de planejamento. O *Product Owner* deve ter a responsabilidade de comunicar a visão do cliente para a funcionalidade em questão, para que o time consiga compreender o nível de complexidade e flexibilidade que pode existir no projeto;
- Reunião diária (*Daily*): Esse evento ocorre diariamente e deve ter a duração de 15 minutos, com todos os envolvidos do projeto presentes. Nessa reunião, todos os integrantes devem comunicar o que foi feito no último dia, o que será feito no dia atual e se existem impedimentos ou dúvidas em suas atividades. O objetivo desse momento é todos terem conhecimento do progresso do incremento, e que o *Product Owner* ou *Scrum Master* possam desimpedir o trabalho dos envolvidos ou saneiam as dúvidas, para que o projeto não atrase ou necessite de retrabalho;
- Revisão do *Sprint*: Esse momento ocorre ao final do *Sprint* e tem o objetivo de revisar o incremento feito e realizar as adaptações nas atividades que serão desenvolvidas no futuro de acordo com o que foi experienciado no *Sprint*. Além disso, é mostrado o que foi possível de concluir durante o ciclo, e o que está pendente. É importante que o time de desenvolvimento esteja presente, juntamente das partes interessadas no produto, para promover a colaboração de todos os envolvidos;
- Retrospectiva do *Sprint*: Essa reunião é feita após a revisão de *Sprint*, onde todo o Time *Scrum* está presente, com o objetivo de inspecionar a si próprio para revisar o que foi feito de certo e errado, criando planos de melhoria para que os problemas sejam solucionados e não voltem a acontecer. Esse momento não só analisa os resultados técnicos, mas também inspecionar como o *Sprint* foi em relação as pessoas e relacionamentos.

2.5.3 Artefatos *Scrum*

Os artefatos *Scrum* representam o trabalho ou o valor para o fornecimento de transparência e oportunidades de inspeção e adaptação, segundo o Guia *Scrum* (SCHWABER; SUTHERLAND, 2014). Estes artefatos tem o objetivo de deixar mais claro e acessível as informações necessárias para o desenvolvimento, e por isso, devem manter uma linguagem padrão e simples para que todo o time consiga compreender o objetivo de cada informação. Os principais artefatos *Scrum* são:

- *Backlog* do Produto: Este artefato é uma lista ordenada de todas as atividades a serem feitas no produto, onde as atividades que estão no topo da lista, são as mais prioritárias. A criação, modificação e priorização devem ser feitas pelo *Product Owner*. Este artefato é dinâmico e nunca está completo, isto é, ele muda de acordo com o andamento do projeto, onde atividades podem ser incluídas ou excluídas, priorizadas ou despriorizadas, de acordo com os *feedbacks* dos clientes e do time de desenvolvimento. Cada item do *Backlog* deve possuir uma descrição, ordem de priorização, estimativa de tempo e responsável;
- *Backlog* da *Sprint*: O *Backlog* da *Sprint* funciona como a lista de atividades a serem feitas naquele *Sprint*. A seleção das atividades para o *Sprint* é feita pelo *Product Owner*, juntamente dos retornos passados pelo time e pelo cliente. Cada atividade deve ser bem descrita para que o responsável consiga entender a atividade a ser feita, bem como a definição de pronto;
- Incremento: esse artefato é a junção de todos os itens completados durante a *Sprint* com o valor dos incrementos das *Sprints* anteriores. É importante que o incremento esteja em uma condição utilizável e atender a definição de "Pronto".

2.6 Trabalhos correlatos

Ao realizar as pesquisas bibliográficas para o presente trabalho, o autor encontrou diversas pesquisas que abordam os temas de gestão ágil de projetos com *Scrum*. A seleção dos trabalhos foi realizada de forma manual a partir de buscas em sites, utilizando de palavras-chave como "metodologias ágeis", "*Scrum*", "Estudo de caso *Scrum*", "Aplicação *Scrum*". Após a leitura de diversos artigos, o autor realizou análises que compilam as pesquisas condizentes, com o objetivo de contribuir com o trabalho.

2.6.1 A utilização da metodologia *Scrum* sob a percepção da equipe de desenvolvimento em uma empresa privada de *software*: Um estudo de caso

O estudo feito (OLIVEIRA, 2014), traz um estudo sobre a aplicação do *Scrum* em uma empresa privada da cidade de João Pessoa, na Paraíba, que trabalha com

desenvolvimento de sistemas para diversas frentes, como Corpo de Bombeiros, Juntas Comerciais, Prefeituras e entre outros. O objetivo da pesquisa foi a analisar a adesão da metodologia sob a visão dos times de desenvolvimento, utilizando dados qualitativos retirados das coletas feitas pela autora. É citado no estudo, que previamente a empresa em questão realizou tentativas de utilizar outras metodologias, com a XP e o próprio *Scrum*. Porém, por falta de treinamento, houveram dificuldades que não permitiram a aplicação correta do método. Então, foi apenas em março de 2013 que houve uma proposta de adesão total do *Scrum*, por parte do time de desenvolvimento.

Após uma análise das respostas dadas pelo time, foi percebido que várias questões foram bem compreendidas, porém foi encontrado espaço para melhorias em pontos que não são bem aplicados na empresa, para que ocorram melhorias no prazo de entrega dos produtos e na satisfação do cliente.

2.6.2 Desenvolvimento de *software*: Aplicabilidade de metodologias ágeis no mercado de *software* em Vitória da Conquista - BA

O estudo apresentado por Alan Moutinho Santiago (SANTIAGO, 2012), tem como objetivo estudar como as empresas da cidade de Vitória da Conquista no estado da Bahia, vêm utilizando as metodologias ágeis na prática. Para isso, foi feito pelo autor uma fundamentação teórica focada nas diversas metodologias ágeis existentes. Ao realizar a análise proposta, foi possível compreender melhor o perfil do quadro de pessoal das empresas da cidade em questão, quais os programas de *software* são utilizados pelas empresas estudadas, e os principais pontos fortes e fracos do processo de desenvolvimento de *software* baseado nas metodologias ágeis.

O artigo ao fim atinge o seu objetivo com a análise feita após a coleta dos dados via questionários *on-line*. Conclui então, que as metodologias ágeis mais utilizadas na cidade são XP e *Scrum* e que segundo constatação dos gestores, esses métodos permitem mais flexibilidade e agilidade ao processo. Além disso, foi pontuado o aumento notável de satisfação junto ao cliente e o fato do time de desenvolvimento estar bem informado durante todo o processo, evitando retrabalhos e jornadas pesadas e cansativas de desenvolvimento.

2.6.3 Aplicabilidade das práticas do *framework Scrum* na gestão dos projetos de desenvolvimento de *software* do exército brasileiro

O trabalho de Leonardo Arthur Lourenço (LOURENÇO, 2019) partiu da conjuntura de três fatores que despertaram a curiosidade do autor, são eles: Constatação do pesquisador que atua no Quadro de Engenheiros Militares, a iniciativa de coletar as percepções sobre os profissionais que atuam na organização, e como as técnicas e práticas ágeis podem atuar sobre o desenvolvimento de *software* no Exército Brasileiro.

O estudo de caso foi então aplicado com os profissionais do Exército Brasileiro que trabalham com Software, e foram coletados dados qualitativos por meio de questionários disponibilizados *on-line*. Após a análise dos resultados, foi visto que a aplicação do *Scrum* teria um alto índice de concordância por meio dos desenvolvedores do Exército Brasileiro, mesmo que o próprio processo existente já seja suficiente.

3 ESTUDO REALIZADO

Esse capítulo objetiva apresentar o estudo realizado, desde a descrição do negócio e do produto, até a aplicação da metodologia ágil *Scrum* na empresa em questão.

3.1 Apresentação do Negócio

A Brognoli é uma tradicional imobiliária de Florianópolis, fundada na década de 50 e atualmente uma das maiores imobiliária do estado de Santa Catarina. Devido ao grande volume de imóveis na pauta de locação, na última década, a imobiliária viu a necessidade de se adaptar às tendências do século XXI, e então começou uma grande jornada de transformação digital na empresa. Com diversas iniciativas acontecendo dentro da imobiliária, alguns colaboradores da área de TI criaram um segmento novo dentro da Brognoli, chamado Nia que trabalhava com mapeamento e automação dos processos da imobiliária, além de utilizar soluções de *chatbot* para resolver o problema da alta necessidade de atendentes devido ao grande volume de atendimentos diariamente. Com o sucesso desses produtos dentro da Brognoli, o Nia começou a ganhar visibilidade de outras imobiliárias ao redor do Brasil, com suas soluções robustas e tecnológicas, e em 2020, culminou na separação do Nia da imobiliária Brognoli, sendo agora uma *startup* do Grupo Brognoli. Hoje, o Nia atende mais de 30 imobiliárias em todas as regiões do Brasil com seus produtos de mapeamento e modelagem de processos, automação de processos, *chatbot* e RPA (*Robot Process Automation*). A *startup* hoje resolve as principais dores que as imobiliárias têm: falta de controle do processo, processos muito manuais e problemas nas comunicações com os clientes.

Com o intuito de identificar e resolver os maiores problemas vivenciados no cotidiano das imobiliárias, o Nia foi dividido em 5 times:

- Operações: são os colaboradores voltados às operações dos processos internos do próprio Nia, como gerenciamento de projetos, comunicação, marketing e pessoas;
- RPA: são os colaboradores responsáveis pela criação e manutenção dos robôs que podem realizar extração de boletos, leituras de documentos e etc. Em inglês “Robotic Process Automation”;
- *Chatbot*: realizam a criação e manutenção dos robôs de conversação, utilizados em imobiliárias para realizar o atendimento dos clientes e direcioná-los de acordo com a intenção. Muitos deles contêm Inteligência Artificial, utilizando NLPs

(*Natural Language Processing*, ou no português “Processamento de linguagem natural”);

- CRM: responsáveis pela criação e manutenção dos funís das imobiliárias no software *Bitrix*. Nesses funís, também é possível criar automações, como atribuição automática de responsável, envio automático de *e-mail*, mensagens e condições de acordo com informações;
- Processos: colaboradores responsáveis por entrar em contato com as imobiliárias, mapear seu processo interno por meio de fluxogramas e modelar novos processos utilizando o software *Bitrix*, prevendo automações para facilitar o cotidiano da operação da imobiliária.

As *startups* são empresas de produtos e soluções em tecnologia da informação que procuram criar produtos que possam ser comercializados em grandes quantidades, sem aumentar os custos operacionais (SEBRAE, 2014). E então, o Nia como *startup*, sentiu a necessidade da criação de um novo produto que pudesse ser distribuído em larga escala. Dessa maneira, foi idealizando um produto inovador, utilizando a modelagem de processos em conjunto das soluções de automação do CRM, criando fluxos de trabalho pré-definidos e automatizados e prevendo um curto tempo de *setup* para cada imobiliária. Dessa forma a *startup* poderia atender uma maior quantidade de imobiliárias e com um custo menor.

Para o Nia criar esse novo produto, foi necessário escolher uma metodologia que fosse adequada as condições vividas pela *startup* naquele momento. Com um time pequeno e trabalhando em regime *home-office*, o grande desafio da empresa foi adequar sua operação para que fosse possível manter os projetos passados, que sustentavam financeiramente a estrutura do Nia, e ao mesmo tempo, criar esse novo produto. Dessa forma, após o estudo das diferentes ferramentas de gerenciamento de projeto, foi escolhido a metodologia *Scrum*.

O estudo de caso da aplicabilidade da metodologia *Scrum*, portanto, está atrelado diretamente ao novo produto da *startup* Nia, concentrando-se em avaliar se o *Scrum* está sendo bem aplicado, além de medir a eficiência do desenvolvimento para o time de desenvolvedores, a transparência de informações sobre o projeto e a validação constante com os *stakeholders*.

3.2 Produto

O produto a ser desenvolvido é um sistema de gestão do relacionamento com o cliente ou *Customer Relationship Management*, em inglês. Esse produto é destinado para imobiliárias de locação de pequeno e médio porte, que tem interesse em iniciar um processo de transformação digital dentro de suas empresas. A partir disso,

foi definido junto com os *Stakeholders* do Nia, os requisitos do projeto deste produto: utilizar tecnologia de forma a aumentar a eficiência das imobiliárias e a transparência com os clientes.

Então, após um estudo dos projetos previamente realizados pelo Nia, foram definidos os principais processos de relacionamento existentes em uma imobiliária de locação, sendo eles:

- **Captação:** compreende o processo de entrada de imóveis, que começa com o interesse do proprietário em alugar seu imóvel;
- **Pré-vendas:** processo de entrada dos interessados em alugar um imóvel, do interesse à visita;
- **Locação:** fluxo desses interessados alugarem um imóvel, da visita à assinatura do contrato;
- **Onboarding:** processo dos primeiros momentos do locatário no imóvel alugado;
- **Chamados:** abrange todo o atendimento de chamados financeiros ou de manutenção do imóvel que pode ser solicitado pelo locatário;
- **Negociação:** processo de negociação contratual de valores ou taxas que pode ser solicitado pelo proprietário;
- **Desocupação:** processo que marca o fim da jornada do locatário na imobiliária, quando o locatário deixa de alugar o imóvel;
- **Distrato:** processo que marca o fim da jornada do proprietário na imobiliária, quando o proprietário tira o imóvel da administração da imobiliária.

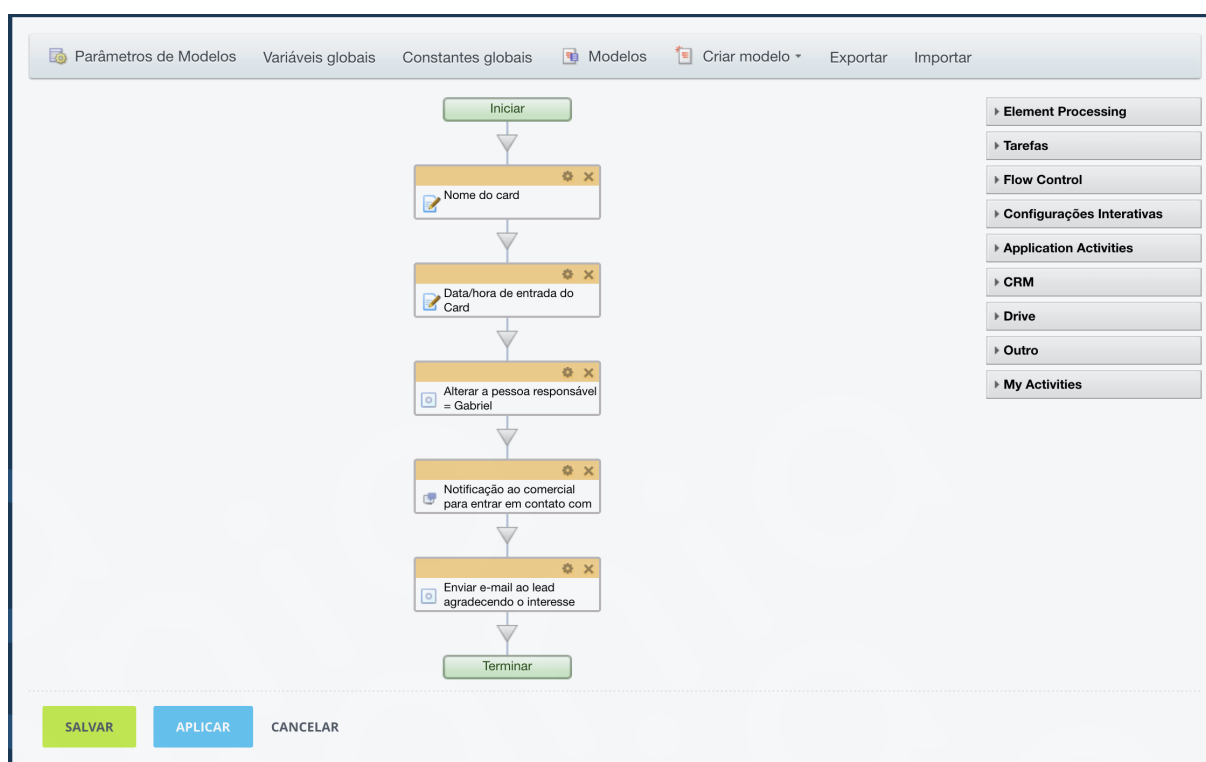
3.3 Tecnologias utilizadas

Visando cumprir a necessidade da empresa de criar um produto com agilidade para que o valor seja entregue para os clientes de forma mais rápida, sendo possível, por fim, validar e realizar alterações em curtos períodos de tempo, foi definido então que seria utilizado o software *Bitrix*, para desenvolver dentro desse sistema cada um dos fluxos citados acima. A escolha de um software já existente para desenvolver a solução foi baseada na agilidade do desenvolvimento, não sendo necessário desenvolver o *front-end*, isto é, a parte que o usuário interage. O *Bitrix* é um software russo, que foi desenvolvido para ser uma plataforma completa de gestão, com a possibilidade de automatizar processos diários e tornar a rotina de trabalho mais eficiente (DIGILANDIA, 2020). Dentre os principais recursos do software estão os bate-papos, gerenciamento

de dados e documentos, *e-mail* integrado, gestão de relacionamento com o cliente, e possibilidade de automatizar fluxos de trabalho.

O Nia utilizará do sistema *Bitrix* como um meio de entregar valor aos seus clientes, desenvolvendo inteligência por meio das automações, que são desenvolvidas por meio de programação em blocos específica do sistema. A figura 4 mostra um fluxo de automações de um processo, usado para exemplificar como é feito o desenvolvimento no sistema.

Figura 4 – Visão do *Bitrix* - Desenvolvimento



Fonte: do Autor(2022)

Para cada um dos fluxos citados previamente, são previstas fases em formato de *Kanban*, com cartões que se movem por essas fases de acordo com a etapa do processo que o negócio se encontra. Cada uma dessas fases contém automações específicas que foram desenvolvidas a partir das melhores práticas do setor imobiliário, visando automatizar os processos e dar mais clareza à operação das imobiliárias. Dentre as automações possíveis, é importante citar:

- Envio de *e-mail* e *Whatsapp* para os envolvidos no processo;
- Atribuição de responsáveis ao negócio;
- Geração de documentos, como contratos;

- Envio de formulários para anexo de documentos;
- Criação de tarefas para os responsáveis;
- Adição, exclusão, alteração de variáveis.

A figura 5 mostra uma visão do processo de Chamados no software *Bitrix*, com os cartões que representam os chamados abertos por proprietários e locatários.

Figura 5 – Visão do *Bitrix* - Chamados

Novo chamado (7)	Em análise (6)	Pendente locatário (4)	Pendente proprietário (1)
R\$0	R\$0	R\$0	R\$0
Roger 9 dezembro 2021 Código do imóvel 12354 Tipo de chamado Financeiro Tipo do cliente Locatário Atividades + Agenda	3 dezembro 2021 Código do imóvel 332211 Tipo de chamado Financeiro Tipo do cliente Proprietário Atividades + Agenda	9 dezembro 2021 Código do imóvel 54321 Tipo de chamado Manutenção Tipo do cliente Locatário Atividades + Agenda	3 dezembro 2021 Código do imóvel 8887 Tipo de chamado Financeiro Tipo do cliente Interno Atividades + Agenda
Paulo 3 dezembro 2021 Código do imóvel 312413 Tipo de chamado Financeiro Tipo do cliente Proprietário Atividades + Agenda	Ana 3 dezembro 2021 Código do imóvel 5645 Tipo de chamado Manutenção Tipo do cliente Proprietário Atividades + Agenda	Tatiana 3 dezembro 2021 Código do imóvel 2200 Tipo de chamado Financeiro Tipo do cliente Proprietário Atividades + Agenda	
negócio repetido 2 dezembro 2021 Código do imóvel 99977 Tipo de chamado Financeiro Tipo do cliente	negócio repetido Taynara 2 dezembro 2021 Código do imóvel 456789 Tipo de chamado Manutenção Tipo do cliente	Marcos 3 dezembro 2021 Código do imóvel 3453 Tipo de chamado Financeiro Tipo do cliente Interno	

Fonte: do Autor(2022)

Cada um dos cartões contém todas as informações daquele negócio, como o código do imóvel correspondente, o endereço, os dados do proprietário e locatário, documentos, e todo o histórico do negócio. Esses dados são armazenados internamente dentro do próprio sistema, como mostra a figura abaixo (figura 6).

Figura 6 – Visão do *Bitrix* - Cartões

SOLICITAÇÃO

Pontuação AI24

editar

ID

4693

Criado em

9 dezembro 2021 10:52

Cliente

Contato

Roger

+94 554 545 555, rogerroger@nia.tech



Imobiliária/Cliente

não selecionada

Prazo de atendimento

14/12/2021 10:52:59

Status do atendimento

Vencido

Código do imóvel

12354

Tipo do cliente

Locatário

Tipo de chamado

Financeiro

Categorias financeiro

Cobrança

Comentário

Cobrança indevida

Responsável



Nia Journey

Fonte: do Autor(2022)

O chamado mostrado na figura foi criado para ilustrar o funcionamento interno do *Bitrix*. Na figura, é possível visualizar os dados do cliente que solicitou um chamado, além de outras informações importantes para o processo. Entre essas informações importantes do fluxo de Chamados estão: prazo de atendimento, tipo de solicitação, categoria da solicitação, comentário do cliente e principalmente o responsável interno sobre a solicitação. Ao ser clicado no nome do cliente, o sistema redireciona para a página de contato, onde são encontrados os dados do cliente, como mostrado

na figura 7.

Figura 7 – Visão do *Bitrix* - Contatos

The image shows a screenshot of a contact form in the Bitrix system. It is divided into two main sections: 'SOBRE O CONTATO' (About the Contact) and 'ENDEREÇO' (Address). The 'SOBRE O CONTATO' section includes fields for ID (813), Name (Roger), WhatsApp (+94 554 545 555), and E-mail (rogerroger@nia.tech). Each field has a 'Trabalho' (Work) tag. There are also links for 'Selecionar campo', 'Criar campo', and 'Excluir seção'. The 'ENDEREÇO' section includes fields for Endereço (Rua), Número (1), Complemento (101), Bairro (Centro), Cidade (Florianópolis), Estado (SC), and CEP (88888888). Each field has an 'editar' (edit) link.

SOBRE O CONTATO		editar
ID	813	
Nome	Roger	
WhatsApp	+94 554 545 555	Trabalho
E-mail	rogerroger@nia.tech	Trabalho
Selecionar campo Criar campo		Excluir seção

ENDEREÇO		editar
Endereço	Rua	
Número	1	
Complemento	101	
Bairro	Centro	
Cidade	Florianópolis	
Estado	SC	
CEP	88888888	

Fonte: do Autor(2022)

Logo, por meio do sistema *Bitrix*, o Nia entrega um produto que abrange toda a jornada dos clientes (proprietários e locatários) na imobiliária, facilitando a operação da imobiliária por meio de automações e dando mais controle das etapas dos processos. Futuramente, após as validações necessárias que o produto atende as necessidades do público alvo, a *startup* pretende criar um sistema próprio e deixar de usar o software *Bitrix*

3.4 Aplicação do *Scrum*

Nessa seção, será explicitado como o *Scrum* é utilizado na empresa Nia, de forma que fique claro a aplicação da metodologia em ordem prática, no contexto da criação do produto citado acima.

3.4.1 Pessoas e responsabilidades

Como visto no capítulo de Fundamentação Teórica, os papéis e responsabilidades do *Scrum* são um ponto fundamental para a melhor aplicação do método, então é importante que o time esteja dividido e com funções claras de acordo com a teoria do *Scrum*.

O Nia contém 24 colaboradores, que são divididos nas equipes já citadas previamente. Para o projeto deste produto, foram alocados 2 colaboradores do time de processos, 6 colaboradores do time de CRM, 1 colaborador do time de operações, além da participação ativa dos fundadores da empresa no alinhamento de expectativas e do autor como gerente de produto e projeto. Então, para o projeto em questão, a divisão de acordo com as práticas do *Scrum* é:

- *Product Owner*: Autor, entendendo as demandas do produto e realizando o gerenciamento do *backlog*, de forma a priorizar e descrever as atividades que o time deverá realizar nas *Sprints*;
- *Scrum Master*: Fundador, cuja função é fazer a verificação se as melhores práticas do *Scrum* estão sendo seguidas e desimpedir as atividades do time de desenvolvimento, para que os objetivos sejam alcançados com mais agilidade;
- Time de desenvolvimento: 9 colaboradores dos times de processos, CRM e operações. Esse time irá efetivamente desenvolver o produto, desde a idealização dos fluxos de acordo com as práticas imobiliárias, desenvolvimento das comunicações e realização das automações dentro do sistema *Bitrix*.

3.4.2 Atividades e *backlog*

Após a definição dos fluxos, foi iniciado o desenvolvimento da lista de atividades por parte do *Product Owner*. As atividades do time de desenvolvimento consistiam em 3 categorias, que existiam para cada um dos 8 fluxos, e portanto, 24 atividades foram criadas. Para cada uma delas, existiam os stakeholders que deveriam ser acionados para validar as entregas e realizar pequenas alterações. As categorias são:

- Modelagem dos fluxos: o time de processos é responsável por essas atividades, visando criar o desenho dos fluxos de acordo com as experiências com as

imobiliárias. Nesses esboços estão contidos as fases e as principais automações que irão entregar valor para as imobiliárias;

- Criação das comunicações: uma colaboradora participante do time de operações realiza a atividade que é criar as comunicações que são enviadas durante os fluxos, mantendo padrões de linguagem e pensando na experiência do cliente;
- Desenvolvimento dos fluxos: esse grupo de atividades são de responsabilidade do time de CRM, que agrupa as informações criadas e recolhidas pelos outros times para desenvolver dentro do sistema *Bitrix* os fluxos com as automações. Essas atividades também compreende os testes que devem ser feitos pelos desenvolvedores para identificar possíveis problemas.

Cada uma das tarefas foi descrita de forma com que pudesse ser compreendido plenamente por toda a equipe, além de agrupar todos os documentos necessários para o desenvolvimento das atividades. Visando as melhores práticas, as descrições das atividades foram padronizadas, sendo necessário preencher claramente as seguintes informações:

- Descrição da atividade: Será feito uma breve descrição do que será feito na atividade;
- Objetivo: essa informação é importante para o time de desenvolvimento entender como sua atividade é importante para o objetivo final do projeto;
- Pontos de contato Nia: informação para que o responsável saiba qual colaborador pode ajudar a entender melhor ou desimpedir a atividade. Além disso, esse colaborador pode ser responsável também pela validação da atividade;
- Pontos de contato Cliente: quais são os clientes que estão nos ajudando a montar e que podem passar *feedbacks* valiosos para a construção do produto;
- Definição de entrega: essa informação tem o objetivo de alinhar as expectativas quanto ao que é esperado de resultado da atividade;
- *Links* importantes: opcional, caso exista alguma informação, documento, link que seja importante para o desenvolvimento da atividade.

Para cada uma das atividades, é estipulado juntamente do time de desenvolvimento, o tempo esperado de duração de cada tarefa e discutido sobre a complexidade esperada da atividade, para ser possível prever assistências e reuniões complementares. Como método de mensurar o tempo e a complexidade da atividade, a empresa

opta por utilizar a medida de pontos de história, onde um ponto de história convencionalmente é utilizado como duas horas, porém pode ser acrescentado um ponto dependendo da complexidade da tarefa. Caso uma atividade tenha a previsão de demorar 4 horas de desenvolvimento, porém é uma atividade complexa, pode ser atribuído 3 pontos de história, prevendo que possa levar até 6 horas para concluir a atividade.

3.4.3 Eventos

Os eventos adotados na empresa são os recomendados pelo método *Scrum*, com o intuito de ter mais eficiência e agilidade. Esses eventos se mostram mais importantes na realidade da *startup* em questão, visto que o time trabalha de forma remota, e logo, tem menos pontos de contato durante o dia. Por isso, é necessário manter uma padronização nos encontros, e que eles ocorram de forma recorrente para que sejam evitados retrabalhos e desalinhamentos.

Os *Sprints* têm a duração de 2 semanas, onde diariamente são feitas reuniões rápidas onde os colaboradores mostram sua página de atividades (que será mostrada a seguir), mostrando o que já foi realizado e o que não foi, apresentando também os impedimentos, para que os presentes na reunião possam auxiliar. No primeiro dia da *Sprint* também é feito uma reunião com todos os envolvidos onde o *Product Owner* apresenta as atividades que serão realizadas no *Sprint*, visando alinhar com toda a equipe o que está sendo produzido.

Durante o *Sprint*, o *Product Owner* reúne os envolvidos para realizar a reunião de planejamento de *Sprint*, onde todos os presentes podem ajudar a descrever as atividades com mais clareza, citar obstáculos e estimar a quantidade de tempo necessárias para as próximas atividades.

Ao fim do *Sprint*, são realizados dois encontros: um para mostrar o que foi realizado, e o que não foi realizado, apresentando os obstáculos que levaram a não completar a tarefa. O outro evento é feito por toda a empresa, onde são mostrados os pontos altos e baixos do *Sprint*, além de incentivar os colaboradores a citarem os aprendizados e passarem *feedbacks* de como obter melhores resultados.

3.4.4 Software utilizado

Para auxiliar a utilização da metodologia ágil, a empresa utiliza o software “*Jira*”. Esse sistema foi criado para ajudar as empresas na adoção de metodologias ágeis, permitindo o gerenciamento dos projetos e monitoramento das atividades de forma simples e visual. O *Jira* permite também a flexibilidade nos processos de acordo com as necessidades de cada equipe, ajudando a organizar o fluxo de trabalho e oferecendo relatórios de acordo com a metodologia ágil escolhida (4INFRA, 2021). No contexto do Nia, a metodologia utilizada dentro do software é o *Scrum*, e é pos-

sível também, por dentro do sistema realizar o planejamento de *sprints* e criação de cronogramas com as atividades.

As atividades citadas previamente foram então adicionadas com suas devidas descrições dentro do sistema, como mostra a figura 8. Na pagina de *backlog* do *Jira*, é possível ver todas as atividades, bem como sua priorização (duas barras amarelas na esquerda das tarefas, mostra que as atividades têm prioridade neutra), além dos responsáveis e da estimativa de tempo, que no caso da figura não estão especificados.

Figura 8 – Visão do Jira - Backlog

Projetos / Nia Journey / Nia Journey

Backlog

Compartilhar

...

Somente meus itens

Recentemente atualizados

Insights

VERSÕES

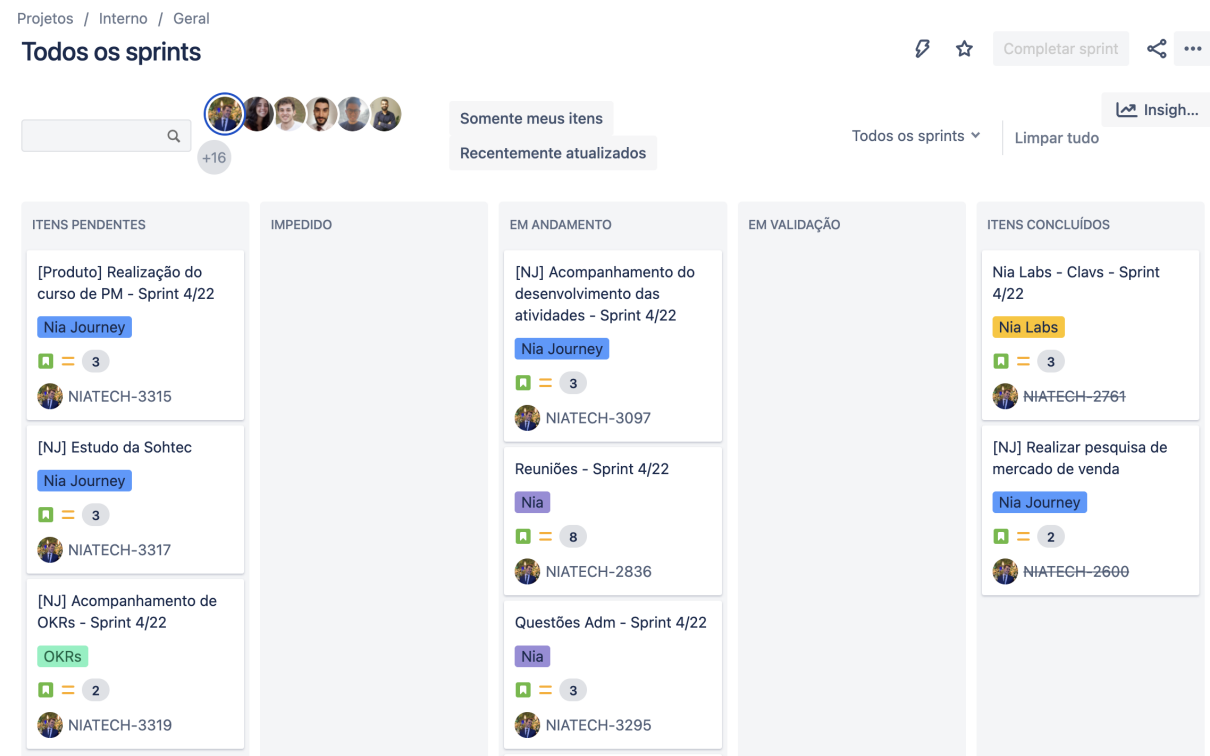
ÉPICOS

[Bitrix] Desenvolvimento do fluxo de pré vendas	Produto	NJ-64	=	-
[Bitrix] Desenvolvimento do fluxo de captação	Produto	NJ-65	=	-
[Bitrix] Desenvolvimento do fluxo de locação	Produto	NJ-66	=	-
[Bitrix] Desenvolvimento do fluxo de Onboarding	Produto	NJ-67	=	-
[Bitrix] Desenvolvimento do fluxo de Chamados	Produto	NJ-68	=	-
[Bitrix] Desenvolvimento do fluxo de Negociação	Produto	NJ-69	=	-
[Bitrix] Desenvolvimento do fluxo de Distrato	Produto	NJ-70	=	-
[Bitrix] Desenvolvimento do fluxo de Desocupação	Produto	NJ-71	=	-
[Processos] Modelagem do fluxo de pré venda	Produto	NJ-72	=	-
[Processos] Modelagem do fluxo de locação	Produto	NJ-73	=	-
[Processos] Modelagem do fluxo de captação	Produto	NJ-74	=	-
[Processos] Modelagem do fluxo de Onboarding	Produto	NJ-75	=	-
[Processos] Modelagem do fluxo de Chamados	Produto	NJ-76	=	-
[Processos] Modelagem do fluxo de Negociação	Produto	NJ-77	=	-
[Processos] Modelagem do fluxo de Distrato	Produto	NJ-78	=	-
[Processos] Modelagem do fluxo de Desocupação	Produto	NJ-79	=	-

Fonte: do Autor(2022)

Após a inserção das tarefas, bem como sua previsão de tempo, responsável e descrição, as atividades foram divididas em *sprints*. Para isso, foi feito um alinhamento com os fundadores para entender a quantidade de esforço que seria possível utilizar durante um ciclo, para conseguir fazer uma previsão de finalização do projeto. Quando as atividades são atribuídas no *Sprint*, elas aparecem em formato de *Kanban* para o responsável por aquela atividade, de modo que fique claro o andamento de suas atividades, como mostra a figura 9. As colunas representam os possíveis status das atividades: Não iniciado, impedido, em andamento, em validação e concluído. O objetivo ao final do *Sprint* é que todas as atividades planejadas estejam concluídas.

Figura 9 – Visão do Jira - Sprint



Fonte: do Autor(2022)

Ao selecionar uma atividade, é possível verificar os detalhes da tarefa, além de ser possível acompanhar a quantidade de tempo que o colaborador trabalhou na atividade. A figura 10 mostra a página da atividade de desenvolvimento do fluxo de chamados no *Bitrix*. A atividade em questão foi desenvolvida em 3 horas e 44 minutos, mostrando que a previsão de 2 pontos de história para a tarefa foi assertiva.

Figura 10 – Visão do Jira - Atividades

Projetos / Interno / NIATECH-119 / NIATECH-1162

[Bitrix] Desenvolver o pipeline de Chamados

AnexarCriar subtarefaVincular itemTrack time with Clockwork

Descrição

Descrição: Realizar o desenvolvimento do fluxo de chamados de acordo com a modelagem feita pelo time de processos, com as comunicações desenvolvidas.

Objetivo: Ter todos os fluxos do novo produto do Nia dentro do Bitrix.

Ponto de contato Nia: @Gabriel Diederichsen

Ponto de contato Nia: Arthur Marqueze Batista

Responsável pela validação: Geovane Silveira

Definição de entrega: Fluxo pronto, validado e testado.

Links: [Nia Journey] Chamados

Clockwork

3h 44m logged

3h 44m

Concluído 2Itens concluídos

Informações

Responsável	Geovane Silveira
Relator	Arthur Marqueze Batista
Categorias	Nenhum
Story Points	2
Tempo restante	0m
Epic Link	Nia Journey
Sprint	Nenhum +1
Prioridade	Medium
Clockwork	Abrir Clockwork

Mais campos Estimativa original, Componentes, Versõe...

Fonte: do Autor (2022)

4 RESULTADOS E DISCUSSÕES

Esse capítulo mostra o procedimento de coleta dos dados adotado para a avaliação do *Scrum* na empresa citada no capítulo 4, bem como os resultados dessa pesquisa.

4.1 Procedimentos de coleta dos dados

Para viabilizar a avaliação do uso dessa metodologia com os colaboradores da *startup* Nia, foi disponibilizado um formulário eletrônico na plataforma *Google Forms*, solicitado que eles respondam à perguntas acessando um *link*. Esse formulário ficou disponível do dia 14 de fevereiro ao dia 17 de fevereiro do ano 2022.

A ferramenta do *Google* também permite que sejam gerados gráficos, relatórios personalizados e o acompanhamento em tempo real do preenchimento, possibilitando então que o autor, juntamente do seu orientador, pudesse avaliar de diversas formas os resultados obtidos.

O questionário contém 29 perguntas que foram divididas em 6 seções, usando como base os estudos feitos sobre os principais pontos do *Scrum*:

- Planejamento;
- Artefatos;
- Papéis e Responsabilidades;
- Eventos;
- Pessoas; e
- Conclusão.

Para as primeiras 20 questões, as respostas possíveis são notas de 1 a 5, onde 1 representa "Discordo Totalmente" e 5 significa que o colaborador "Concorda totalmente". O restante das respostas corresponde formatos variados, como perguntas abertas, de perfil, e avaliação com nota 1 a 5. O formulário foi enviado para 10 colaboradores da empresa, onde apenas 8 responderam no período estipulado e suas respostas estão registradas de forma anônima. Todas as perguntas e respostas de cada um dos colaboradores estão em anexo para conhecimento. Na tabela ?? estão contidas todas as perguntas feitas.

Perguntas enumeradas:

1. No início do processo de desenvolvimento, é elaborado uma lista de atividades que serão desenvolvidas;

2. É feita uma priorização das atividades que são mais importantes e estratégicas;
3. Para cada atividade é estimada uma quantidade de esforço da equipe;
4. Toda a equipe envolvida participa da estimativa das atividades a serem desenvolvidas;
5. O projeto segue o modelo iterativo e incremental.
6. O software utilizado, "Jira", ajuda na organização e no acompanhamento do desenvolvimento;
7. A estimativa feita para a equipe é condizente com o que é possível de produzir no tempo estipulado;
8. O backlog é constantemente atualizado de acordo com os feedbacks recebidos;
9. As atividades do Sprint backlog são detalhados, de forma que todos os membros consigam entender a atividade a ser feita;
10. A equipe checa constantemente o andamento do projeto diariamente, para verificar se está indo dentro do prazo estipulado;
11. Existe uma pessoa na equipe responsável por resolver os impedimentos;
12. Existe uma pessoa na equipe responsável por checar se o Scrum está sendo corretamente aplicado;
13. Existe uma pessoa na equipe responsável por gerenciar os backlogs;
14. O cliente é visto como membro da equipe;
15. A equipe tem flexibilidade na tomada de decisões que dizem respeito a implementação;
16. São feitas reuniões diárias para realizar o acompanhamento das atividades;
17. Os impedimentos levantados nessas reuniões são rapidamente resolvidos;
18. Ao final de cada Sprint, há uma reunião da equipe para apresentar o que foi desenvolvido;
19. Ao final de cada Sprint, há uma reunião da equipe para apresentar os pontos positivos e negativos do Sprint;
20. São realizadas validações com os stakeholders após as entregas incrementais;
21. Tempo que você trabalha com desenvolvimento de software;

22. Liste quais metodologias de desenvolvimento de sistemas você conhece, indicando entre estas quais você já utilizou;
23. Quais os pontos fortes e fracos do Scrum você identifica em relação as outras metodologias que conhece?;
24. Como você avalia a transparência durante o projeto?;
25. Como você avalia a flexibilidade durante o projeto?;
26. Como você avalia a eficiência do time durante o projeto?;
27. Como você avalia a sua satisfação durante o projeto?;
28. Como você avalia o uso do Scrum como metodologia de desenvolvimento? ;
29. Para conclusão do questionário, caso você queira justificar alguma das suas respostas ou complementar a pesquisa com alguma informação;

4.2 Análise dos dados

Como primeiro passo da avaliação do *Scrum*, é necessário analisar o perfil das 8 pessoas que responderam. Como mostrado na figura 11, 12,5% dos respondentes trabalham com desenvolvimento de software a mais de 10 anos, 25% trabalha entre 2 e 5 anos e 37,5% trabalham a menos de dois anos na área. Assim, é possível entender que, dos respondentes, o colaborador com mais experiência é sênior, 2 colaboradores são plenos (de 2 a 5 anos de desenvolvimento) e 3 são juniores ou estagiários (menos de 2 anos). Além disso, os 25% restantes, ou 2 colaboradores, responderam que não desejam responder essa questão. Na avaliação do autor, juntamente de sua experiência na empresa, estas pessoas não são desenvolvedores, trabalhando no projeto em diferentes frentes. Para melhor compreensão das análises à seguir, estes colaboradores serão chamados de "Não Desenvolvedores"

Figura 11 – Tempo que o colaborador trabalha com desenvolvimento de software

Fonte: do autor (2022).

Traçados os perfis dos colaboradores, foi possível a partir de suas respostas (Tabela 3) que as pessoas com mais experiência (plenos e sênior) têm conhecimento e experiência em outras metodologias ágeis, já os juniores ou estagiários, têm conhecimento apenas do *Scrum*. Essas respostas indicam que aqueles colaboradores que estão iniciando sua trajetória profissional já estão em contato com essa metodologia, mostrando uma grande difusão desse método nas empresas de desenvolvimento. O colaborador 8 indicou que tem conhecimento sobre *Kanban*, porém esta não é considerada uma metodologia ágil, e sim uma forma de visualização dos projetos em andamento. Os não desenvolvedores não citaram metodologias.

Tabela 3 – Quais metodologias de gestão são conhecidas de acordo com experiência

Colaborador	Experiência	Metodologias
Colaborador 1	Sênior	PMBOK e <i>Scrum</i>
Colaborador 2	Pleno	Waterfall e <i>Scrum</i>
Colaborador 3	Não Desenvolvedor	-
Colaborador 4	Não Desenvolvedor	-
Colaborador 5	Júnior	<i>Scrum</i>
Colaborador 6	Júnior	<i>Scrum</i>
Colaborador 7	Pleno	XP e <i>Scrum</i>
Colaborador 8	Júnior	Kanban e <i>Scrum</i>

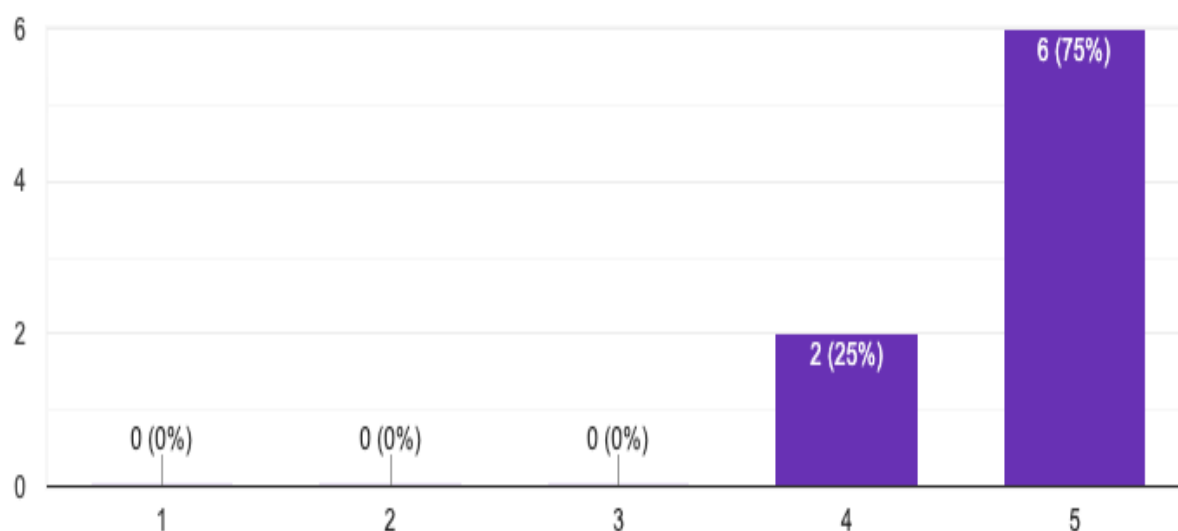
Fonte: do autor (2022).

As subseções abaixo serão divididas de acordo com os principais pontos do *Scrum*, para facilitar a análise e compreensão dos resultados.

4.2.1 Planejamento

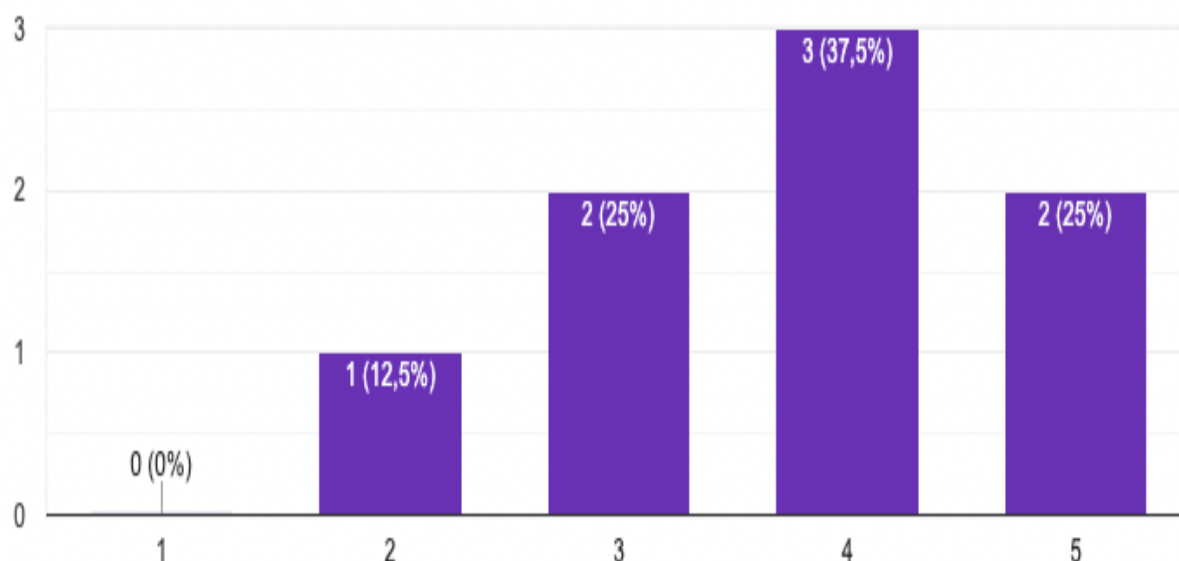
Na seção de planejamento, o objetivo era avaliar se o planejamento do *Scrum* estava de acordo com o que é recomendado na aplicação da metodologia. Nessa etapa, é possível então, visualizar e fazer algumas correlações das respostas com o perfil dos colaboradores. Então, como mostra na figura 12, a maioria dos respondentes concordou que são elaborados listas de atividades que serão desenvolvidas, e apenas 2 colaboradores responderam com nota 4. Estes, são de perfil júnior e não desenvolvedor, e muitas vezes podem não participar desse momento de planejamento, visto que atualmente as listas são feitas pelo gestor do projeto com os principais desenvolvedores e líderes, dependendo da demanda do projeto.

Figura 12 – No início do processo de desenvolvimento, é elaborado uma lista de atividades que serão desenvolvidas



Fonte: do autor (2022).

Essa tese se confirma na figura 13, onde foi possível ver diferentes respostas para a afirmação "Toda a equipe envolvida no projeto participa da estimativa das atividades a serem desenvolvidas", onde os dois colaboradores citados acima que deram nota 4, agora dão notas 2 e 4. Nesse gráfico, é possível notar como foram espalhadas as respostas dos colaboradores, mostrando que algumas pessoas, de perfis variáveis, não se sentem contempladas durante a estimativa das atividades a serem desenvolvidas. Portanto, esse seria um ponto de melhoria na aplicação *Scrum* dentro da empresa.

Figura 13 – Toda a equipe participa da estimativa das atividades a serem desenvolvidas

Fonte: do autor (2022).

Outro ponto interessante pode ser visto na tabela 4, onde os não desenvolvedores reportam não concordar nem discordar com a afirmação que o projeto segue o modelo iterativo e incremental, enquanto todos os desenvolvedores concordam com a afirmação. Como esses não desenvolvedores podem realizar tarefas mais administrativas ou específicas, é possível que não participem de todo o projeto, tendo uma visão mais pontual, não tendo contato com as continuações, validações e assim por diante.

Tabela 4 – Respostas dos colaboradores para a questão 5 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	5
Colaborador 3	Não Desenvolvedor	3
Colaborador 4	Não Desenvolvedor	3
Colaborador 5	Júnior	5
Colaborador 6	Júnior	5
Colaborador 7	Pleno	5
Colaborador 8	Júnior	5

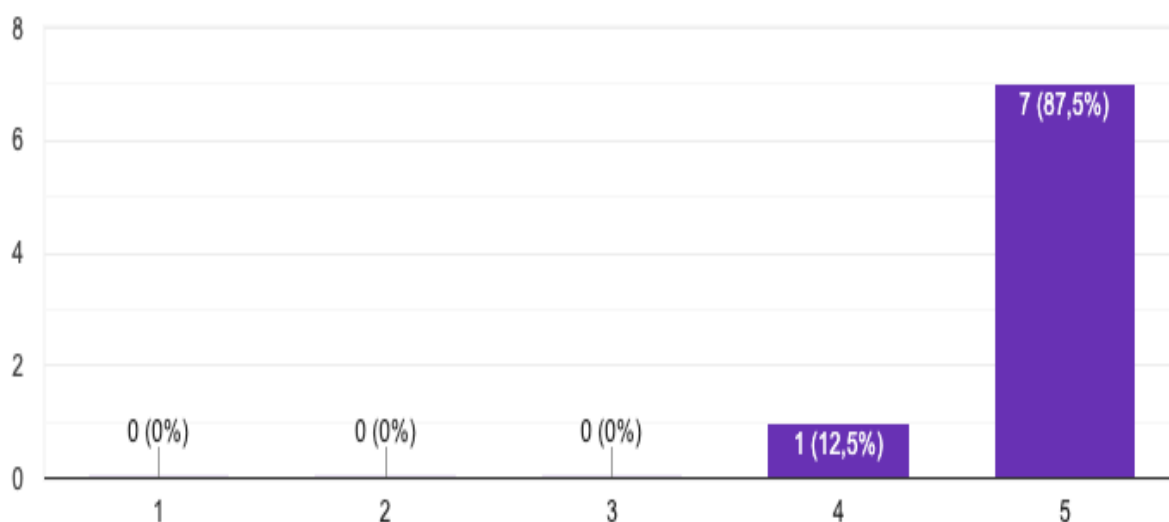
Fonte: do autor (2022).

Além disso, as demais respostas dessa seção tiveram resultados similares, mostrando que os colaboradores concordam que a priorização das atividades é feita e que para cada atividade realizada é estimado um esforço da equipe, como é possível ver no Anexo A.

4.2.2 Artefatos

Na seção de artefatos do questionário, o objetivo era avaliar os artefatos estudados no capítulo 3 e entender se a aplicabilidade destes tem sido efetiva no contexto da empresa estudada. A primeira pergunta dessa seção (Questão 6) é mostrado na figura 14 teve o objetivo de entender se o software utilizado e descrito no capítulo 4, o Jira, ajuda na organização e acompanhamento do desenvolvimento. Quase todos os colaboradores respondentes concordaram totalmente com essa afirmação, sendo apenas um deles, o colaborador 1, sênior, respondeu que concorda parcialmente (nota 4). Uma justificativa plausível a isso pode ser encontrada pareando esse perfil com as respostas na questão 23, onde é pedido para citar os pontos fortes e fracos do *Scrum* comparados com outras metodologias. Esse colaborador cita que um ponto fraco é a dificuldade na aplicação. Considerando que a plataforma utilizada serve justamente para aplicar o *Scrum* e que o colaborador que não concorda plenamente é o que têm mais experiência, conclui-se que alguns pontos da aplicação *Scrum* podem não estar contemplados na plataforma Jira.

Figura 14 – O software utilizado, "Jira", ajuda na organização e no acompanhamento do desenvolvimento

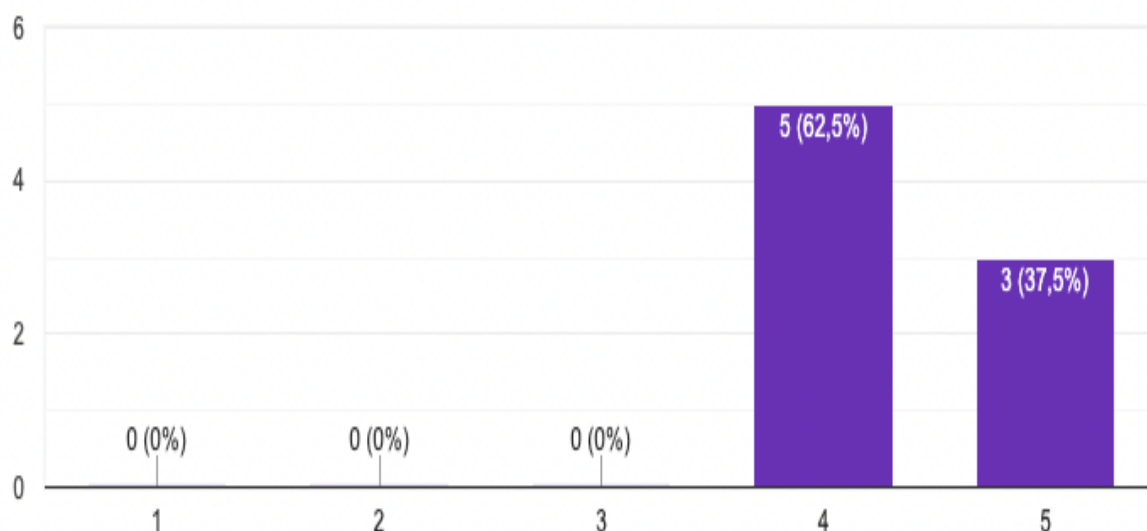


Fonte: do autor (2022).

Outras duas respostas chamaram a atenção do autor, são as questões 7 e 9, mostradas abaixo nas figuras 15 e 16 respectivamente. Na figura 15, relativa a questão 7, é possível ver que 5 colaboradores não concordam totalmente com a afirmação "A estimativa feita com a equipe condiz com o que é possível produzir no tempo estipulado". Como vimos na análise feita na subseção do planejamento, os respondentes se dividiram na afirmação que o time participa das estimativas (figura 13),

e pela análise e experiência do autor, a não participação dos colaboradores na hora de estimar o tempo de desenvolvimento das atividades pode ter sido um dos motivos pelo qual os respondentes não concordam totalmente com a afirmação da questão 7.

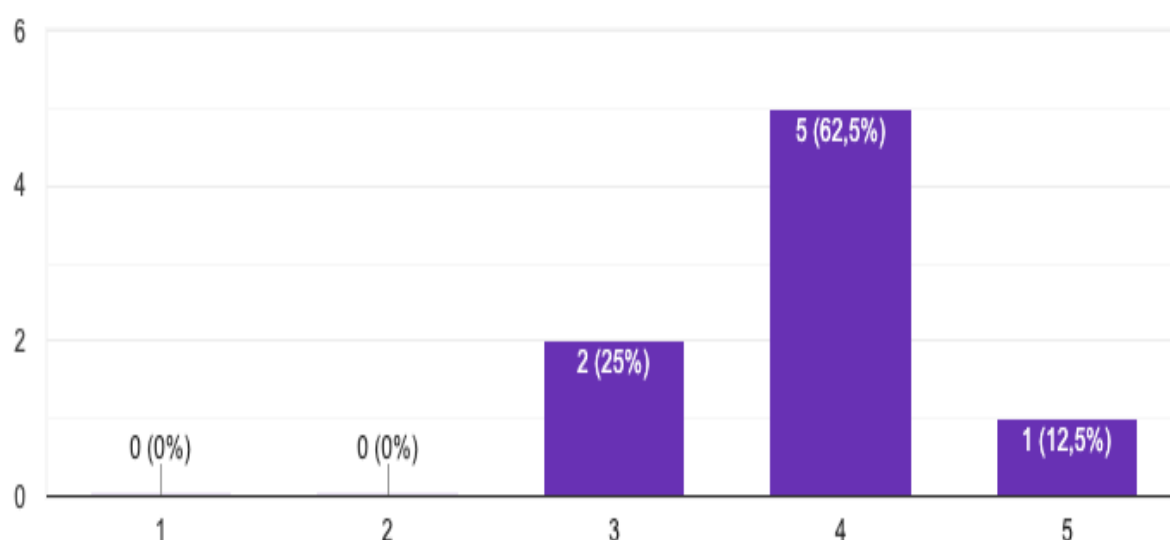
Figura 15 – A estimativa feita para a equipe é condizente com o que é possível de produzir no tempo estipulado



Fonte: do autor (2022).

Na questão 9 (figura 16), é possível ver que o time também se divide na afirmação "As atividades do *Sprint* Backlog são detalhados, de forma que todos os membros consigam entender a atividade a ser feita". É possível ver esses resultados de forma mais explícita comparando com o perfil dos respondentes na tabela 5, onde os dois colaboradores que responderam com nota 3 são de dois perfis distintos, um é desenvolvedor júnior, e outro não é desenvolvedor. O motivo pela nota mais baixa dada por esses colaboradores se deve à um ter pouca experiência e outro não trabalhar diretamente com o desenvolvimento, tornando o entendimento das atividades limitada pela falta de experiência. Porém, autor avalia que este é um ponto que pode ser melhorado dentro da empresa. Além disso, o colaborador 7, que citou que já teve experiência com a metodologia XP, concordou totalmente (nota 5) com essa afirmação. Como foi visto na fundamentação teórica, a metodologia XP não compreende um *backlog* como o *Scrum*, e portanto, é possível afirmar que, visto as experiências passadas desse colaborador, o *Scrum* se mostra melhor que a metodologia XP quanto a descrição e entendimento das tarefas.

Figura 16 – As atividades do *Sprint* Backlog são detalhados, de forma que todos os membros consigam entender a atividade a ser feita"



Fonte: do autor (2022).

Tabela 5 – Respostas dos colaboradores para a questão 9 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	4
Colaborador 3	Não Desenvolvedor	3
Colaborador 4	Não Desenvolvedor	4
Colaborador 5	Júnior	3
Colaborador 6	Júnior	4
Colaborador 7	Pleno	5
Colaborador 8	Júnior	4

Fonte: do autor (2022).

4.2.3 Papéis e Responsabilidades

Na seção sobre Papéis e Responsabilidades, o objetivo era avaliar o entendimento do time sobre as funções exercidas pelos colaboradores dentro do time. Estas funções foram estudadas de forma mais aprofundada no capítulo 3. As questões foram formuladas para evitar usar os nomes utilizados na metodologia (*Scrum Master*, *Product Owner*), mas citar as funções que exercem. Dessa forma, é possível garantir que todos os respondentes entendam e respondam com clareza.

A questão 11 e 12 avaliam a presença do *Scrum Master*, com as afirmações "Existe uma pessoa responsável por resolver os impedimentos" e "Existe uma pessoa

responsável por checar se o *Scrum* está sendo corretamente aplicado". Para a primeira afirmação, mostrado na tabela 6, vemos que aqueles que têm mais experiência no desenvolvimento (um sênior e um pleno), respondem com nota 4, mais baixo que o restante da equipe. Isso se deve a complexidade dos impedimentos que estes têm, que resolvem problemas com um alto nível de sofisticação. Estes, muitas vezes servem para retirar os impedimentos existentes dos outros colaboradores, por sua alta experiência.

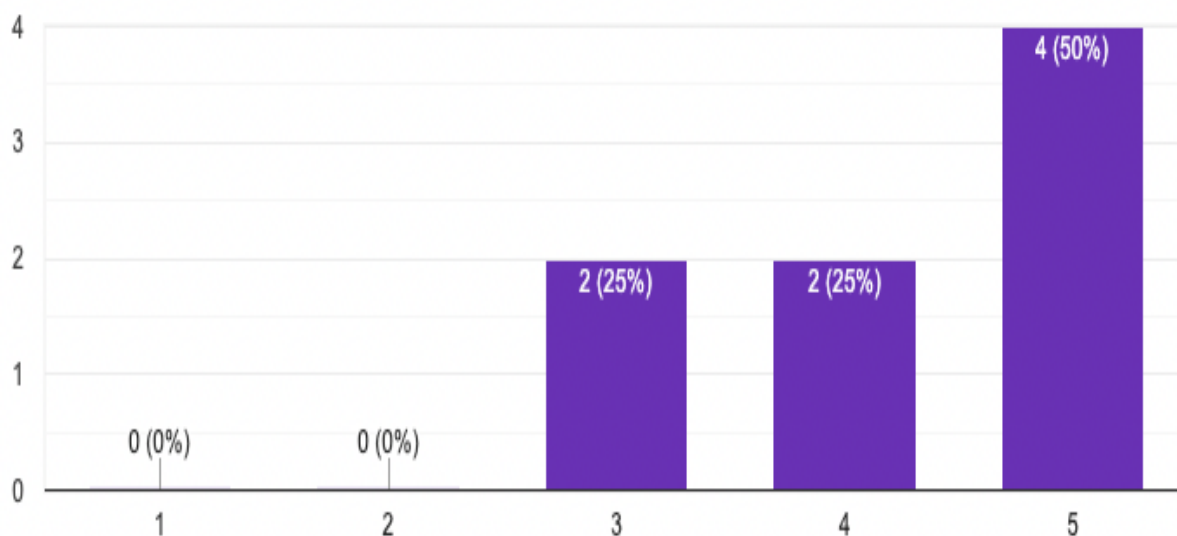
Tabela 6 – Respostas dos colaboradores para a questão 11 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	5
Colaborador 3	Não Desenvolvedor	5
Colaborador 4	Não Desenvolvedor	5
Colaborador 5	Júnior	5
Colaborador 6	Júnior	5
Colaborador 7	Pleno	4
Colaborador 8	Júnior	5

Fonte: do autor (2022).

Para a questão 12 (figura 17), é visto que 2 respondentes de perfil júnior e pleno, não têm clareza sobre a existência de um colaborador responsável por checar as boas práticas do *Scrum*. Na visão do autor, estes não concordam nem discordam pelo alto conhecimento de todo o time sobre *Scrum*, que se auto-organiza constantemente. Esta característica foi citada na fundamentação teórica ao citar as principais características do time *Scrum*.

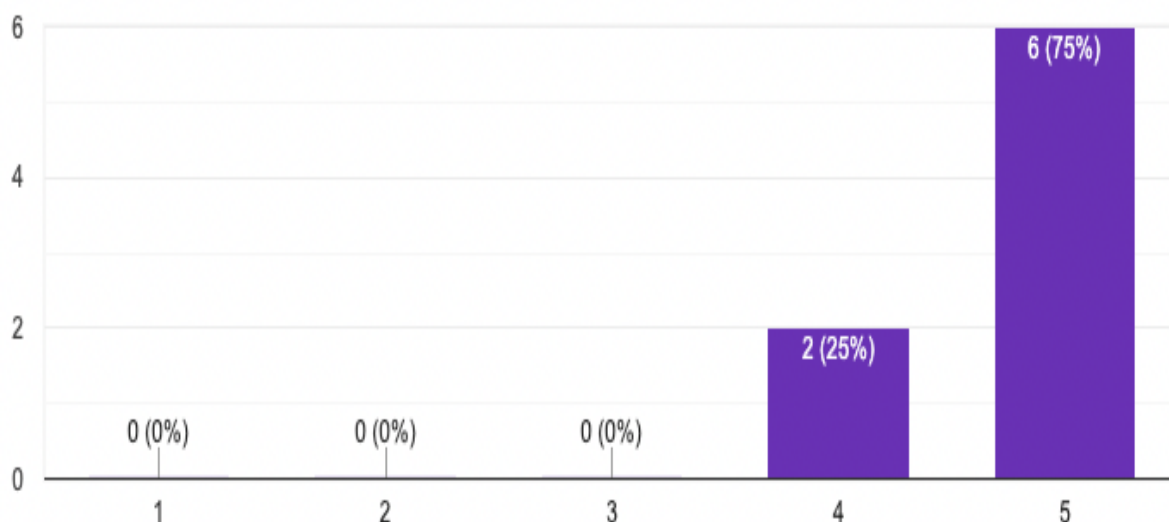
Figura 17 – Existe uma pessoa na equipe responsável por checar se o *Scrum* está sendo corretamente aplicado



Fonte: do autor (2022).

A questão 13, "Existe uma pessoa na equipe responsável por gerenciar os backlogs", é usada para avaliar a função do *Product Owner*, e é mostrado na figura 18 que o time concorda com a existência desse colaborador.

Figura 18 – Existe uma pessoa na equipe responsável por gerenciar os backlogs



Fonte: do autor (2022).

Como foi visto no capítulo 3, o cliente tem uma grande participação no desenvolvimento do projeto utilizando *Scrum*. A questão 14, então, tem o objetivo de

avaliar se os colaboradores veem a participação dos clientes durante o desenvolvimento. Porém, os resultados se mostraram dispersos, como visto na tabela 7. Os não desenvolvedores e desenvolvedores mais experientes (plenos e sênior) responderam com notas negativas ou neutras (2 ou 3). Já os juniores, são os responsáveis pelas notas mais altas (4 ou 5). Neste caso, o autor acredita que o contato com o cliente e compreensão do mercado imobiliário foi um fator essencial para as notas dadas. O autor observou empiricamente que os desenvolvedores com menos experiência são aqueles que menos têm contato com os clientes, e por isso, acreditam estar fazendo algo embasado nas demandas do cliente. Já aqueles com mais experiência de mercado, entendem que a construção do produto não necessariamente seguem as demandas urgentes das imobiliárias.

Tabela 7 – Respostas dos colaboradores para a questão 14 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	3
Colaborador 2	Pleno	2
Colaborador 3	Não Desenvolvedor	3
Colaborador 4	Não Desenvolvedor	3
Colaborador 5	Júnior	5
Colaborador 6	Júnior	4
Colaborador 7	Pleno	2
Colaborador 8	Júnior	5

Fonte: do autor (2022).

Outro ponto importante que foi verificado por meio da resposta da última questão dessa seção, foi a flexibilidade na tomada de decisões que dizem respeito à implementação, onde 7 dos 8 colaboradores concordam plenamente com a afirmação. No *Scrum*, uma das principais características é a flexibilidade que o time de desenvolvimento têm para implementar as atividades.

4.2.4 Eventos

Nessa seção do questionário, o objetivo era checar se a realização dos eventos é feito de acordo com as boas práticas do *Scrum*. A questão 16 (tabela 8) aborda um dos eventos mais importantes do método *Scrum*, e o 6 de 8 colaboradores concordam que são feitas reuniões diárias para o acompanhamento das atividades. De acordo com a experiência do autor, os outros 2 colaboradores, um sênior e um não desenvolvedor, podem não ter participado de todas as reuniões diárias, pois podem não ter tido atividades destinadas em algum dos *Sprints* que foram desenvolvidos.

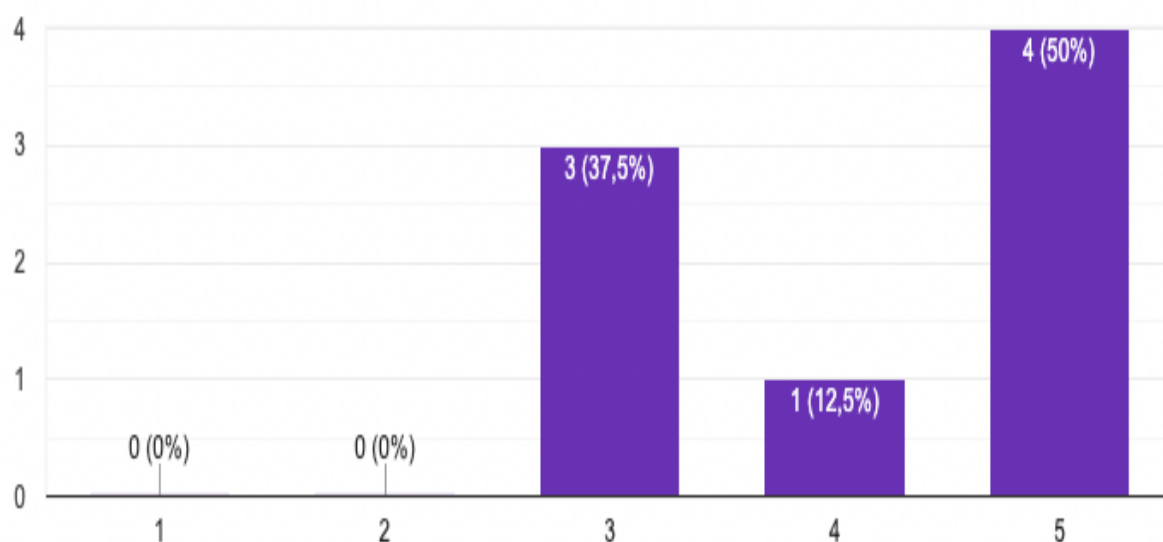
A questão 17 aborda o tópico *Scrum* que fala sobre a resolução dos impedimentos nessas reuniões diárias. Como mostra a figura 19, os resultados foram

Tabela 8 – Respostas dos colaboradores para a questão 16 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	5
Colaborador 3	Não Desenvolvedor	5
Colaborador 4	Não Desenvolvedor	4
Colaborador 5	Júnior	5
Colaborador 6	Júnior	5
Colaborador 7	Pleno	5
Colaborador 8	Júnior	5

Fonte: do autor (2022).

distintos, e diversos perfis responderam que não concordam nem discordam com essa afirmação. Considerando a análise feita previamente, na questão 11 (tabela 6), que os colaboradores concordam que existe uma pessoa que realiza os impedimentos, é possível afirmar que este responsável não está sendo efetivo no desimpedimento das atividades, sendo mais um ponto de melhoria para a empresa quanto à aplicação do *Scrum*.

Figura 19 – Os impedimentos levantados nessas reuniões são rapidamente resolvidos

Fonte: do autor (2022).

Assim como nas questões relacionadas aos papéis e responsabilidades, para as questões 18 e 19, não foi utilizado o nome teórico dos outros eventos *Scrum*, que são a Revisão de *Sprint* e Retrospectiva de *Sprint*. Dessa forma, é possível garantir que se compreenda as reuniões por suas funções, e não pelo nome. Para

estas questões, foram vistas respostas distintas por diferentes perfis, porém, apenas 1 colaborador discorda que existem essas reuniões. O autor não conseguiu chegar a nenhuma análise que justifique de forma plausível essa resposta, visto que, pela experiência do autor na empresa, essas reuniões existem.

A questão 20 afirma que são realizadas validações com os stakeholders após as entregas incrementais, e é visto que 6 de 8 colaboradores concordam em algum nível com essa afirmação, porém, 2 colaboradores de perfil júnior e não desenvolvedor não concordam nem discordam com essa afirmação. Pela análise das respostas do não desenvolvedor, na questão 5 (tabela 3), que afirma que o projeto segue o modelo incremental e iterativo, foi visto que sua resposta foi a mesma (nota 3). Como foi analisado previamente, esse colaborador pode não participar de todas as partes do projeto e portanto, ter pouco contato com os stakeholders. Além disso, pode-se utilizar da mesma análise para o desenvolvedor júnior, que como citado previamente (tabela 7), pode não ter contato direto com os clientes quanto os outros desenvolvedores.

Tabela 9 – Respostas dos colaboradores para a questão 20 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	4
Colaborador 3	Não Desenvolvedor	4
Colaborador 4	Não Desenvolvedor	3
Colaborador 5	Júnior	3
Colaborador 6	Júnior	4
Colaborador 7	Pleno	4
Colaborador 8	Júnior	5

Fonte: do autor (2022).

4.2.5 Pessoas

Essa seção, além de ter perguntas usadas para diferenciar o perfil dos colaboradores da empresa, conta também com perguntas sobre a opinião dos respondentes quanto a transparência, flexibilidade, eficiência e satisfação durante o projeto.

Na questão 24, foi possível obter a opinião dos colaboradores quanto a transparência. Como visto na tabela abaixo, 7 dos 8 colaboradores deram notas boas (4 ou 5) e apenas 1 avaliou com nota 3. Este colaborador é um não desenvolvedor e, como dito previamente, pode não participar de algumas partes do projeto, prejudicando no entendimento completo do projeto, e cita também na questão aberta 23 que um ponto fraco do *Scrum* é justamente a transparência.

Como é possível ver no Anexo B, os colaboradores 5, 6 e 8, todos júniores citam a transparência como um ponto forte do *Scrum*. Porém, nenhum destes citou

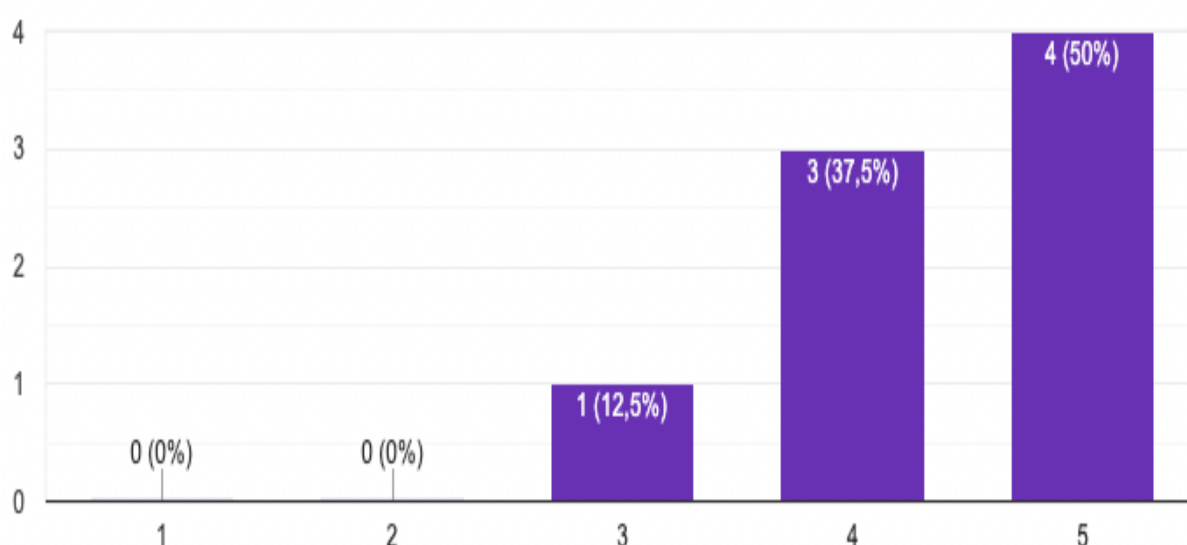
Tabela 10 – Respostas dos colaboradores para a questão 24 de acordo com a experiência

Colaborador	Experiência	Resposta
Colaborador 1	Sênior	4
Colaborador 2	Pleno	5
Colaborador 3	Não Desenvolvedor	5
Colaborador 4	Não Desenvolvedor	3
Colaborador 5	Júnior	5
Colaborador 6	Júnior	5
Colaborador 7	Pleno	5
Colaborador 8	Júnior	5

Fonte: do autor (2022).

conhecer ou ter experiência com alguma outra metodologia, e por isso, não é possível afirmar que o *Scrum* é melhor nesse quesito que outros métodos.

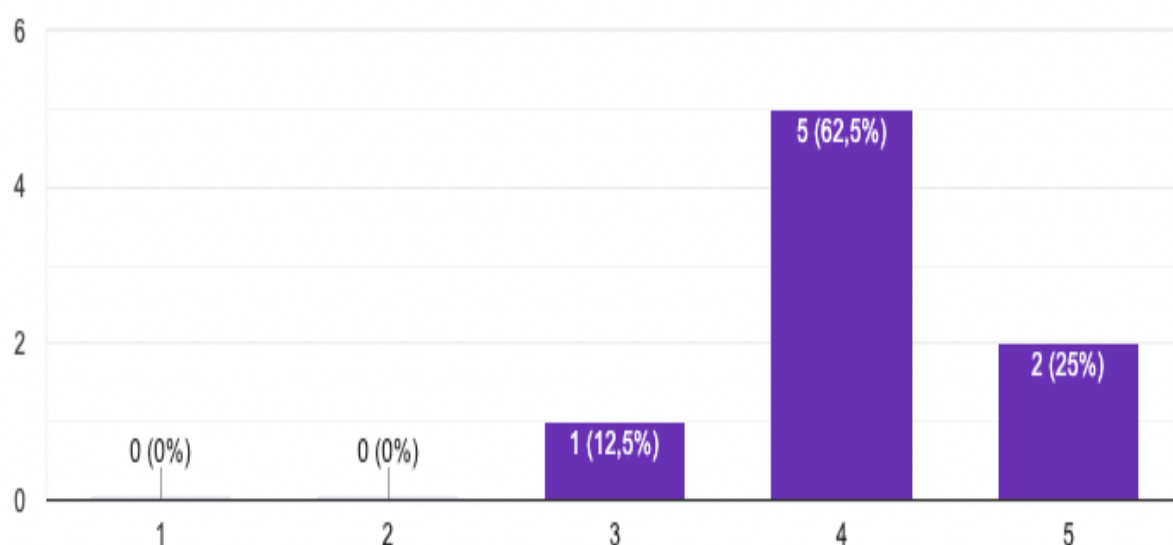
A questão 25 (figura 20) avalia a opinião dos colaboradores quanto a flexibilidade do projeto. As resposta foram variadas, mas ainda sim em sua maioria, 7 de 8 são notas boas. Apenas um colaborador, também não desenvolvedor, deu nota 3 para a flexibilidade do projeto. Esse mesmo colaborador concordou totalmente na questão 15 que a afirmação que o time têm flexibilidade na tomada de decisão quanto a implementação. O autor entende, partindo de ambas respostas, que na visão desse colaborador, as implementações são flexíveis e podem ser feitas da forma que o time acredita ser mais efetivo, porém, o escopo do projeto não é tão flexível.

Figura 20 – Como o colaborador avalia a flexibilidade durante o projeto?

Fonte: do autor (2022).

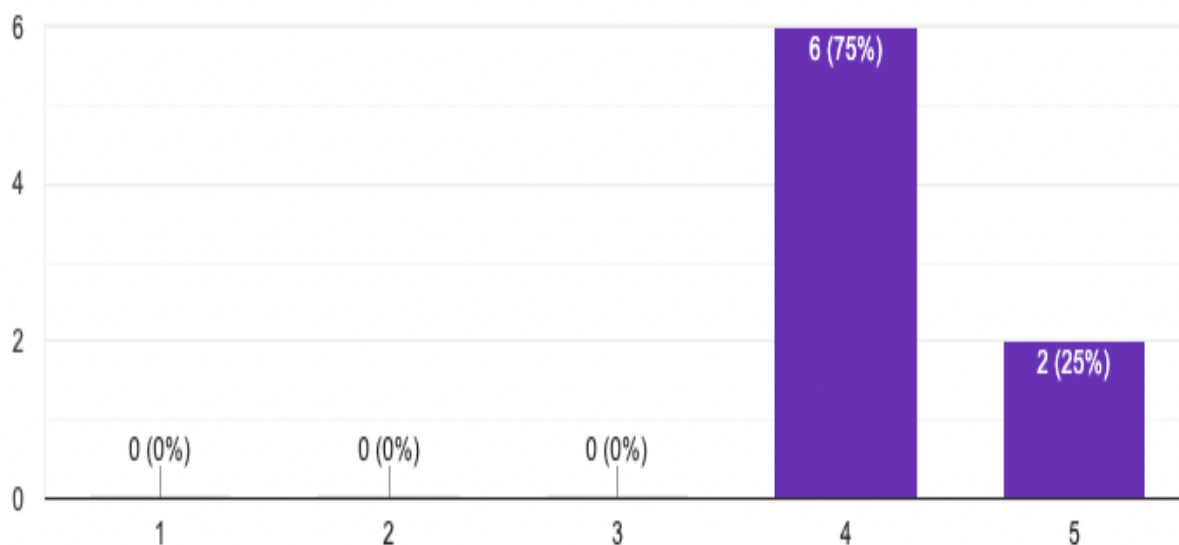
A questão 26 (figura 21) pergunta aos colaboradores sobre a eficiência do time durante o projeto. Nessa questão, apenas um colaborador votou com uma nota mediana (nota 3). A partir das análises feitas do perfil deste colaborador presentes no anexo B, este é um desenvolvedor pleno que já obteve experiências com a metodologia XP e cita que o ponto fraco do *Scrum* é a falta de programação pareada, uma das características do método citado. O autor então, acredita que a programação pareada do método XP pode gerar mais eficiência no desenvolvimento.

Figura 21 – Como o colaborador avalia a eficiência durante o projeto?



Fonte: do autor (2022).

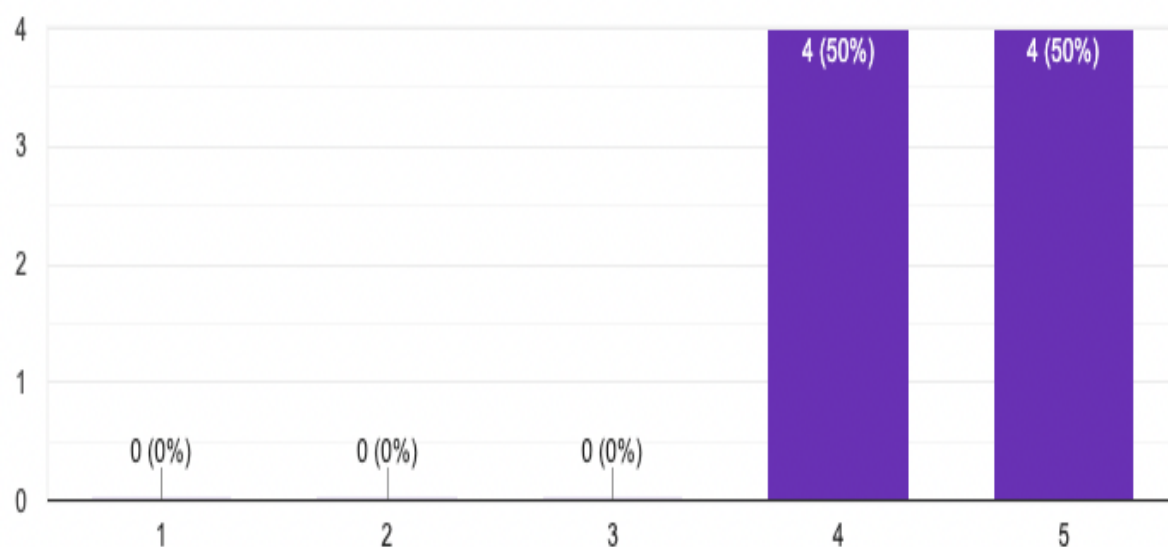
Por fim desta seção, é perguntado aos colaboradores (questão 27) como avaliam a satisfação durante o projeto, e 6 de 8 deram nota 4, correspondente a "Bom", como pode ser visto na figura 22. Os outros 2 colaboradores, atribuíram nota máxima, mesmo que um destes respondentes (5) cite como ponto fraco do *Scrum* (Anexo B) que as mudanças rápidas e frequentes no projeto possam causar estresse e burnout.

Figura 22 – Como o colaborador avalia a própria satisfação durante o projeto?

Fonte: do autor (2022).

4.2.6 Finalização

Para finalizar o questionário, é questionado para os colaboradores como eles avaliam o uso do *Scrum* como metodologia de desenvolvimento de software. Como mostra na figura 23, as respostas foram positivas, com uma nota média de 4,5.

Figura 23 – Como o colaborador avalia o *Scrum* como metodologia de desenvolvimento de projeto?

Fonte: do autor (2022).

Além disso, foi incluído no formulário um campo aberto opcional como abertura para que o respondente pudesse justificar alguma resposta, porém, nenhum dos colaboradores registrou nenhuma resposta nesse campo.

5 CONCLUSÃO

O presente trabalho teve como objetivo estudar e avaliar a aplicabilidade da metodologia ágil *Scrum* no contexto de uma *startup* que atua no mercado imobiliário. Além disso, era de interesse do estudo verificar se as características sugeridas de planejamento, papéis, artefatos, eventos e pessoas do método estavam sendo cumpridas de forma aderente pela empresa em questão.

Então, de acordo com os dados coletados com essa pesquisa, pode-se concluir que os objetivos foram atingidos de forma satisfatória, pois verificou que o time compreende e adere as práticas do método. Foi visto, ao fim do questionário, que os envolvidos avaliam positivamente o uso do *Scrum* como metodologia ágil, além de aprovarem que o uso do método traz mais flexibilidade, transparência, eficiência e satisfação no decorrer dos projetos. Como pontos positivos apresentados na pesquisa, destaca-se a importância de ter um sistema para acompanhar o desenvolvimento das atividades.

Porém, foi visto que mesmo que a empresa adote as práticas mais comuns, é importante que alguns pontos sejam otimizados para que se busque o uso ideal da metodologia. Como principais pontos, pode ser citado que as reuniões diárias devem buscar a realização das atividades com mais agilidade, adotar mais momentos de contato com o cliente ao longo dos *Sprints*, maior comprometimento dos envolvidos para mensurar e detalhar melhor as atividades do *backlog*, além de centralizar um colaborador na equipe para checar se o *Scrum* está sendo aplicado. Portanto, o autor acredita que a empresa em questão pode olhar os resultados obtidos nesse estudo para trazer mais clareza sobre pontos de melhoria.

Por fim, mesmo que o estudo se limitou a estudar a realidade de apenas uma instituição, as contribuições deste estudo são importantes nos contextos práticos de empresas de desenvolvimento de sistemas, contribuindo para gerar mais transparência, eficiência e satisfação nos projetos realizados. Como trabalhos futuros, a sugestão é avaliar a metodologia *Scrum* em empresas de outros contextos, fazendo comparativo entre elas para entender melhor a aplicabilidade desse método no gerenciamento de projetos.

REFERÊNCIAS

4INFRA. **Jira Software: o que é e para que serve?** 2021. Disponível em: <<https://4infra.com.br/jira-software-o-que-e-e-para-que-serve/>>. 42

DIGILANDIA. **Bitrix24: o que é, como funciona, os planos e como usar.** 2020. Disponível em: <<https://digilandia.io/ferramentas-para-trabalho-remoto/bitrix24/#:~:text=A%20Bitrix24%20foi%20desenvolvida%20para,s%C3%A3o%20diversos%20os%20recursos%20dispon%C3%ADveis.>> 35

EMPRESÔMETRO. **Scrum: o método ágil mais utilizado no mundo corporativo.** 2020. Disponível em: <<https://blog.empresometro.com.br/scrum-o-metodo-agil-mais-utilizado-no-mundo-corporativo/>>. Acesso em: 25 jun 2020. 25, 28

FADEL, A. **Metodologias ágeis no contexto de desenvolvimento de software: XP, Scrum e Lean.** 2010. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/3307430/Lean_Agil_v8-with-cover-page-v2.pdf?Expires=1643232112&Signature=LUFypOguheM211yiqpJDgluOjRji4gxa5RzSZNRw8xMgwbNOqViHu9xSXxz4noIBiTtohOWpk-2v__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. 22

FILHO, D. L. B. **Experiências com Desenvolvimento ágil.** 2008. Disponível em: <<https://www.ime.usp.br/~dairton/files/Dissertacao-DairtonBassi.pdf>>. 22

FRANCO, E. F. **Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta.** 2007. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/3/3141/tde-09012008-155823/pt-br.php>>. 23

JUSTO, A. S. **Conheça 3 metodologias ágeis que vão transformar o seu jeito de conduzir projetos.** 2019. Disponível em: <<https://www.euax.com.br/2019/04/metodologias-ageis/>>. 23

KEELING, R. **GESTÃO DE PROJETOS.** Saraiva Educação S.A., 2017. ISBN 9788502227118. Disponível em: <<https://books.google.com.br/books?id=TDtnDwAAQBAJ>>. 16

KERZNER, H. **Gestão de Projetos - 4.ed.: As Melhores Práticas.** Bookman Editora, 2020. ISBN 9788582605301. Disponível em: <<https://books.google.com.br/books?id=piTKDwAAQBAJ>>. 17, 18, 19

LEAL, L. de Q. **Uma abordagem Ágil ao Gerenciamento de Projetos de Software baseada no PMBOK Guide.** 2008. 18

LOBÃO LUIS; ZILLI, C. **A jornada da transformação digital.** [S.l.: s.n.], 2020. 13

LOURENÇO, L. A. E. **Aplicabilidade das práticas do framework Scrum na gestão dos projetos de desenvolvimento de software do exército brasileiro.** 2019. Disponível em: <<https://bdex.eb.mil.br/jspui/bitstream/123456789/4537/1/18%20-%20TCC%20Leonardo%20Arthur%20Louren%C3%A7o.pdf>>. 31

NATHAN-REGIS, B. N. **EVALUATION OF THE MOST USED AGILE METHODS (XP, LEAN, SCRUM)**. 1. ed. [S.l.]: LAP Lambert Academic Publishing, 2012. 23

NOLETO, C. **Modelo cascata: saiba o que é e porque é um modelo ultrapassado**. 2020. Disponível em: <<https://blog.betrybe.com/tecnologia/modelo-cascata/>>. 21

OLIVEIRA, J. F. **A utilização da metodologia Scrum sob a percepção da equipe de desenvolvimento em uma empresa privada de software: Um estudo de caso**. 2014. Disponível em: <<https://si.dcx.ufpb.br/wp-content/uploads/2015/12/Jessyca-Ferreira-de-Oliveira.pdf>>. 19, 30

PEREIRA, G. V. **Metodologia Lean de Desenvolvimento de Software: Uma visão geral**. 2012. 24

PMI (Ed.). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. 5. ed. Newtown Square, PA: Project Management Institute, 2013. ISBN 978-1-935589-67-9. 16, 17, 19

POPPENDIECK, M. **Implementando o Desenvolvimento Lean de Software: Do conceito ao Dinheiro**. 2011. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=FJZ1mq5Vjo4C&oi=fnd&pg=PT1&dq=desenvolvimento+de+software+lean&ots=LQVkNJg_NE&sig=mDIEsYioijqWoCBQX_9Pp0MPwgY#v=onepage&q=desenvolvimento%20de%20software%20lean&f=false>. 24

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. [S.l.]: AMGH Editora Ltda., 2011. 20, 21

SANTIAGO, A. M. **Desenvolvimento de software: Aplicabilidade de metodologias ágeis no mercado de software em Vitória da Conquista - BA**. 2012. Disponível em: <<http://www2.uesb.br/computacao/wp-content/uploads/2014/09/DESENVOLVIMENTO-DE-SOFTWARE-APLICABILIDADE-DE-METODOLOGIAS-%C3%81GEIS-NO-MERCADO-DE-SOFTWARE-EM-VIT%C3%93RIA-DA-CONQUISTA-Monografia-Alan-Moitinho-Santiago.pdf>>. 19, 20, 31

SCHWABER, K.; SUTHERLAND, J. *The scrum guide*. 2014. 25, 27, 28, 30

SEBRAE. **O que é uma startup**. 2014. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/artigos/o-que-e-uma-startup,6979b2a178c83410VgnVCM1000003b74010aRCRD>>. 34

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. [S.l.]: São Paulo: Pearson Prentice Hall, 2011. 19

WIKIPEDIA. **Modelo em cascata**. 2021. Disponível em: <https://pt.wikipedia.org/wiki/Modelo_em_cascata#/media/Ficheiro:Modelo_em_cascata.png>. 21

APÊNDICES

APÊNDICE A – QUESTIONÁRIO E RESPOSTAS



Questionário de avaliação do Scrum como metodologia ágil

Perguntas Respostas 8 Configurações

8 respostas



Não está aceitando respostas



Mensagem para os participantes

Este formulário não aceita mais respostas

Resumo

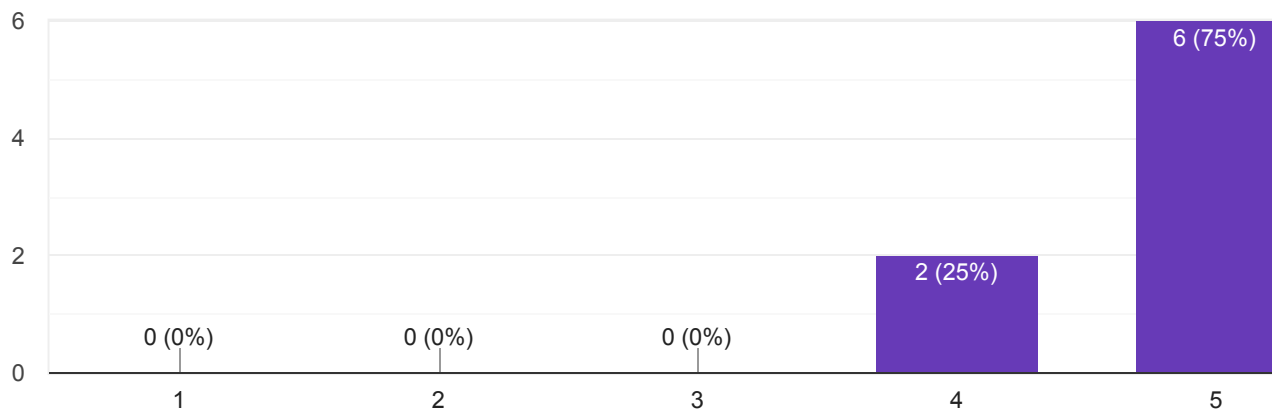
Pergunta

Individual

Seção 1 - Planejamento

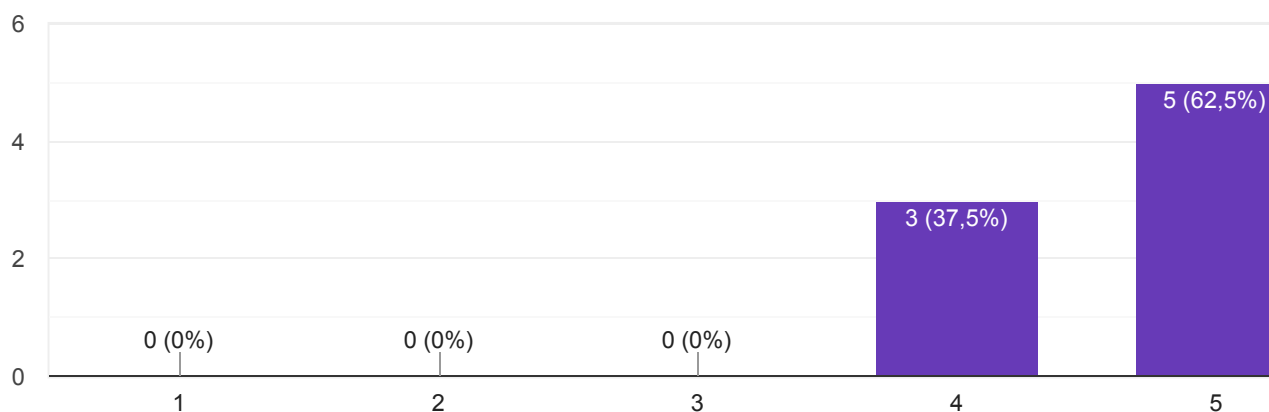
Q1: No início do processo de desenvolvimento, é elaborado uma lista de atividades que serão desenvolvidas

8 respostas

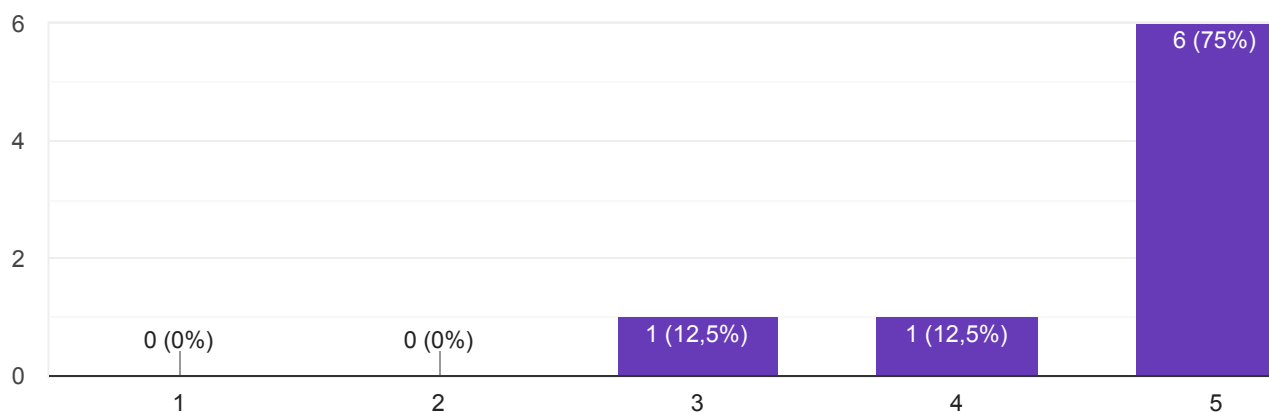


Q2: É feita uma priorização das atividades que são mais importantes e estratégicas

8 respostas

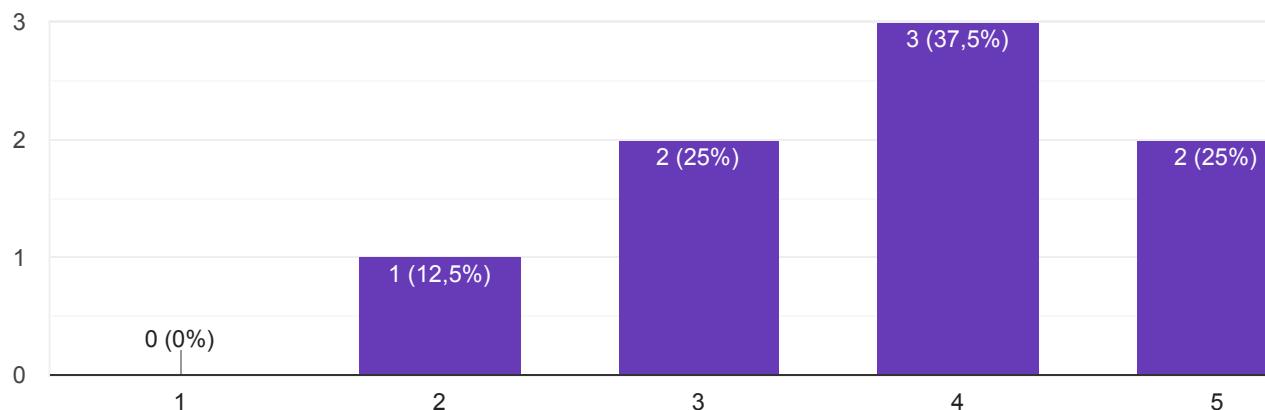
**Q3: Para cada atividade é estimada uma quantidade de esforço da equipe**

8 respostas



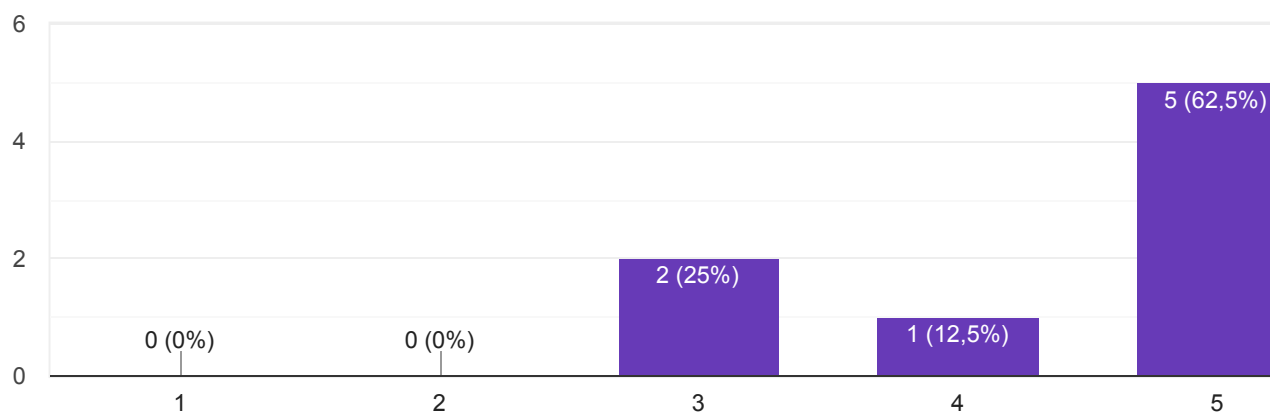
Q4: Toda a equipe envolvida participa da estimativa das atividades a serem desenvolvidas

8 respostas



Q5: O projeto segue o modelo iterativo e incremental

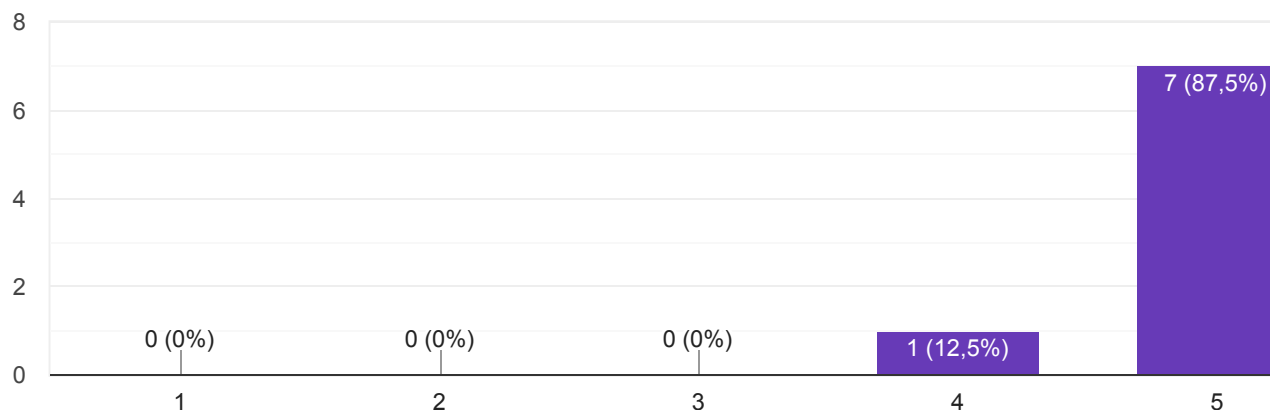
8 respostas



Seção 2 - Artefatos

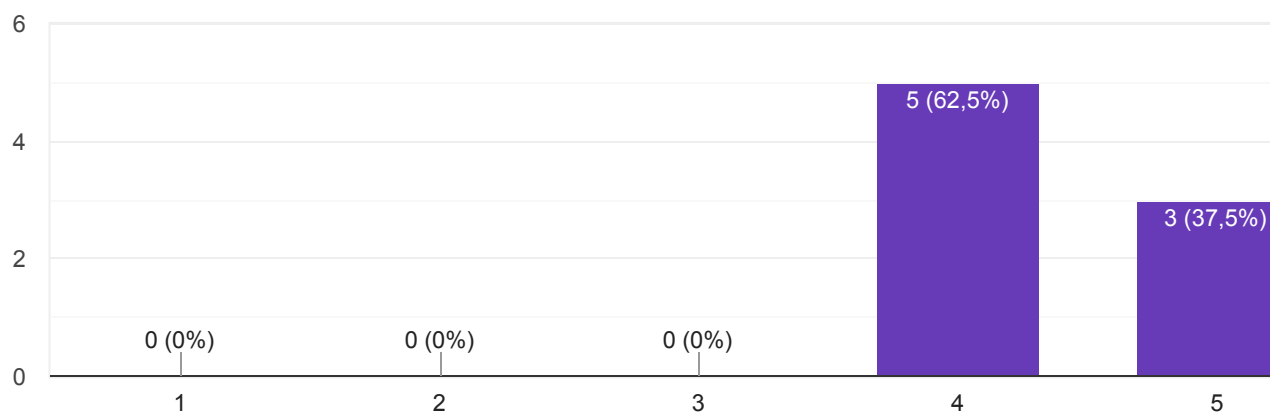
Q6: O software utilizado, "Jira", ajuda na organização e no acompanhamento do desenvolvimento

8 respostas



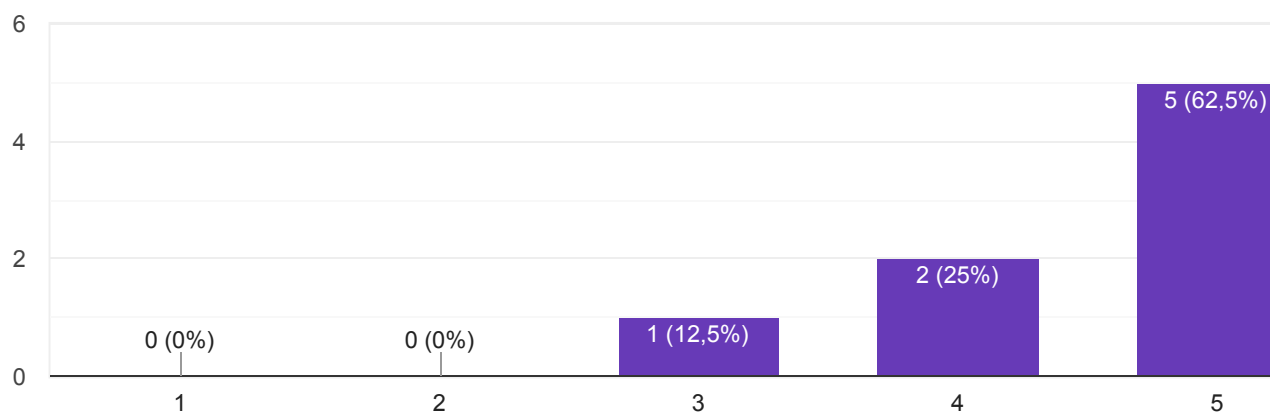
Q7: A estimativa feita para a equipe é condizente com o que é possível de produzir no tempo estipulado

8 respostas

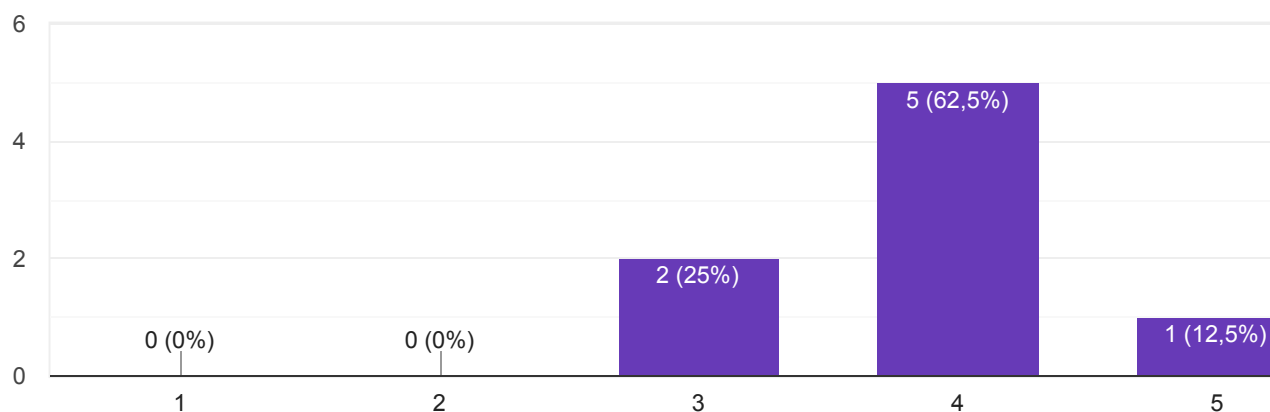


Q8: O backlog é constantemente atualizado de acordo com os feedbacks recebidos

8 respostas

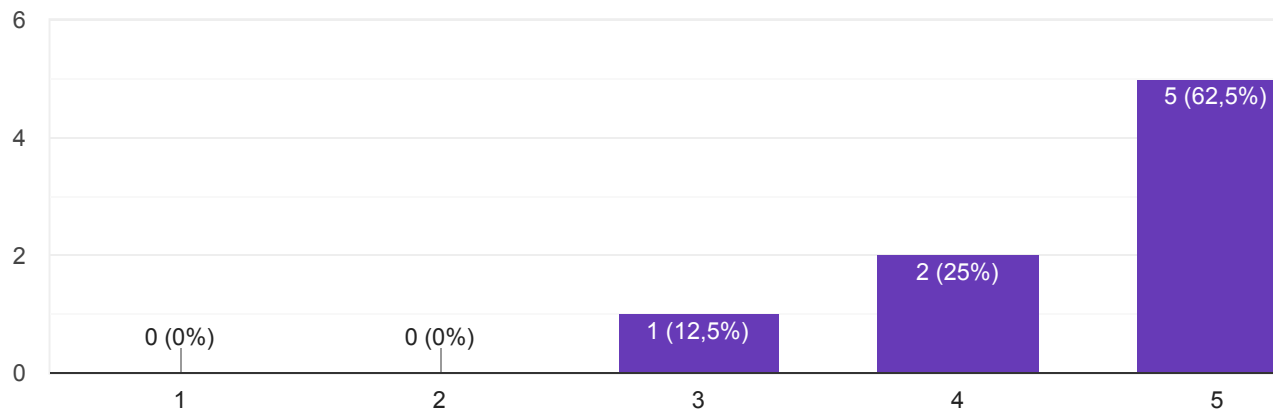
**Q9: As atividades do Sprint Backlog são detalhados, de forma que todos os membros consigam entender a atividade a ser feita**

8 respostas



Q10: A equipe checa constantemente o andamento do projeto diariamente, para verificar se está indo dentro do prazo estipulado

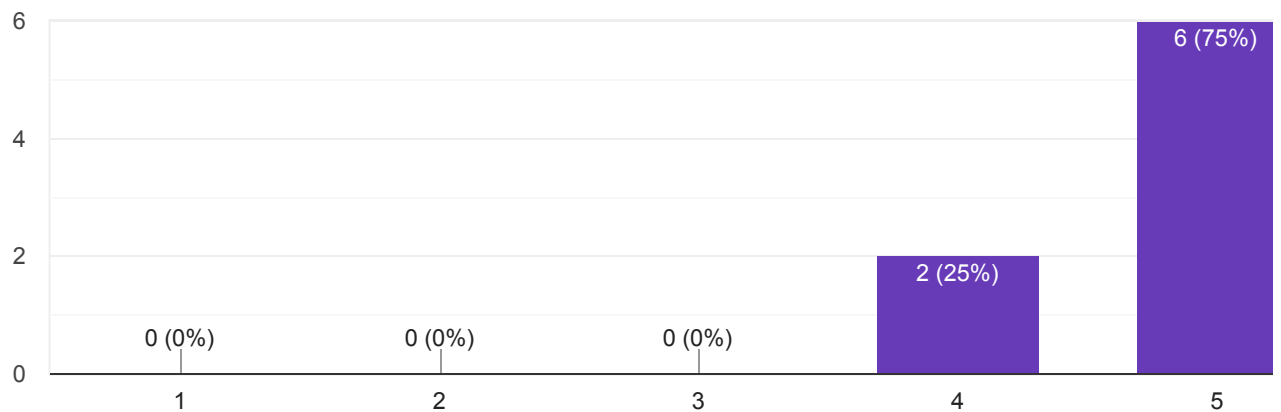
8 respostas



Seção 3 - Papéis e responsabilidades

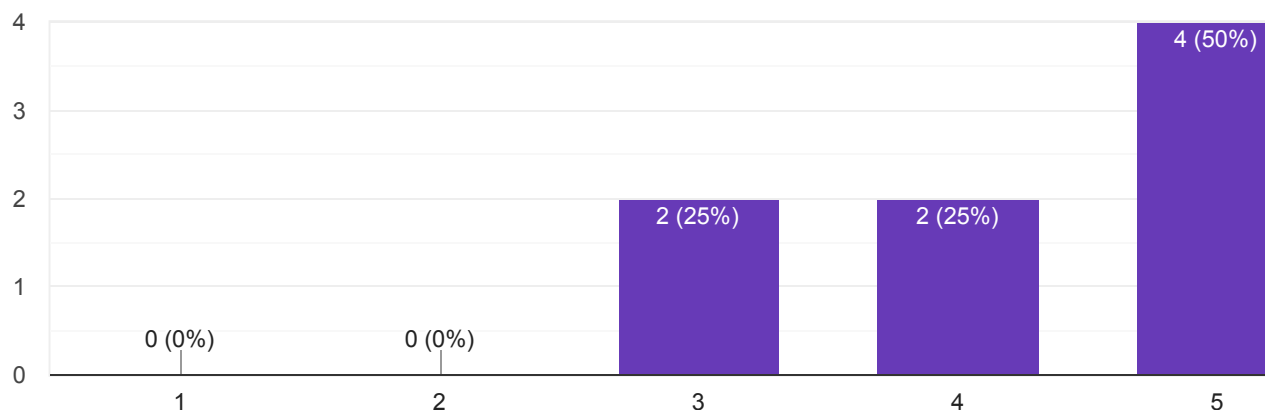
Q11: Existe uma pessoa na equipe responsável por resolver os impedimentos

8 respostas

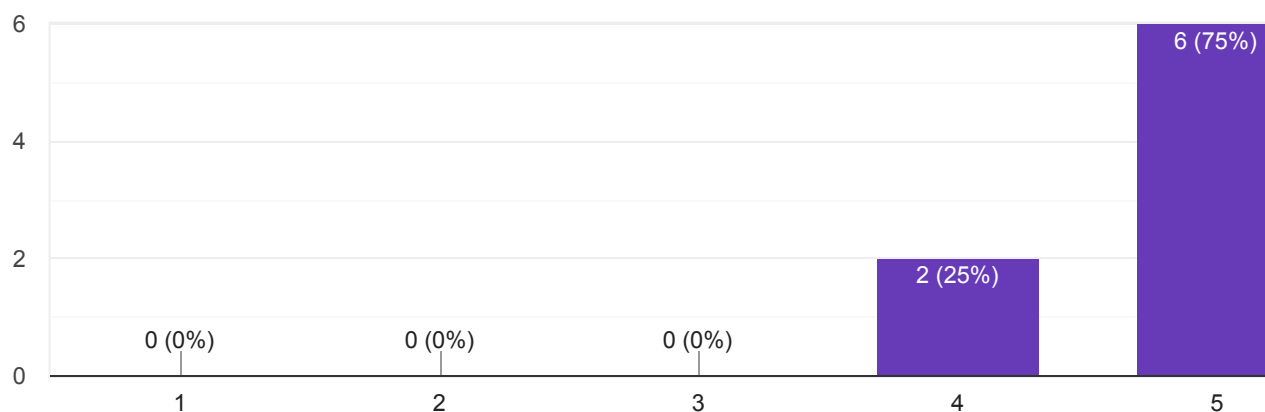


Q12: Existe uma pessoa na equipe responsável por checar se o Scrum está sendo corretamente aplicado

8 respostas

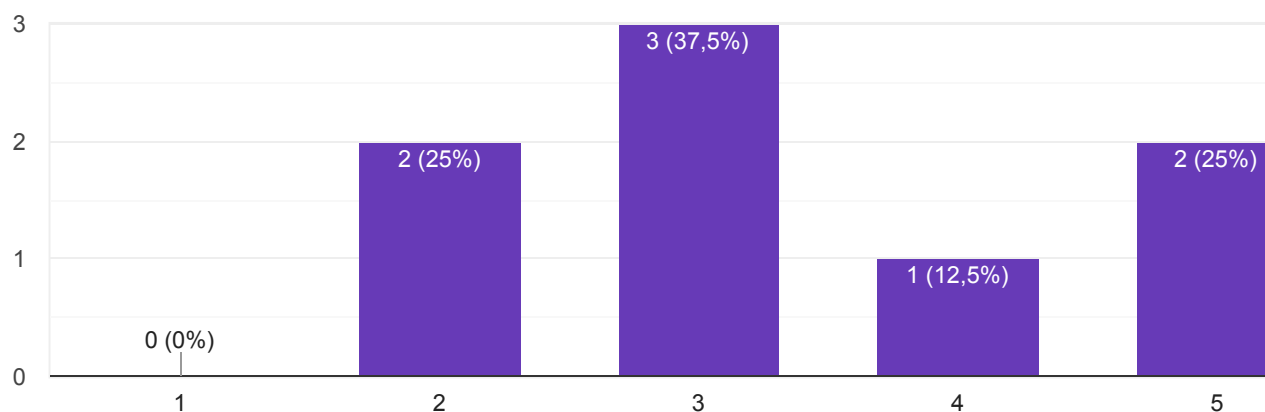
**Q13: Existe uma pessoa na equipe responsável por gerenciar os backlogs**

8 respostas

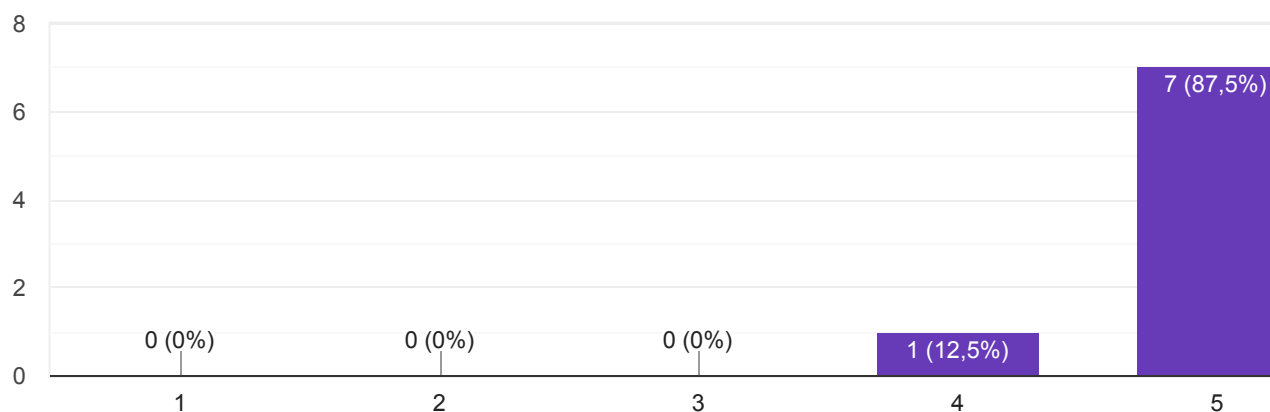


Q14: O cliente é visto como membro da equipe

8 respostas

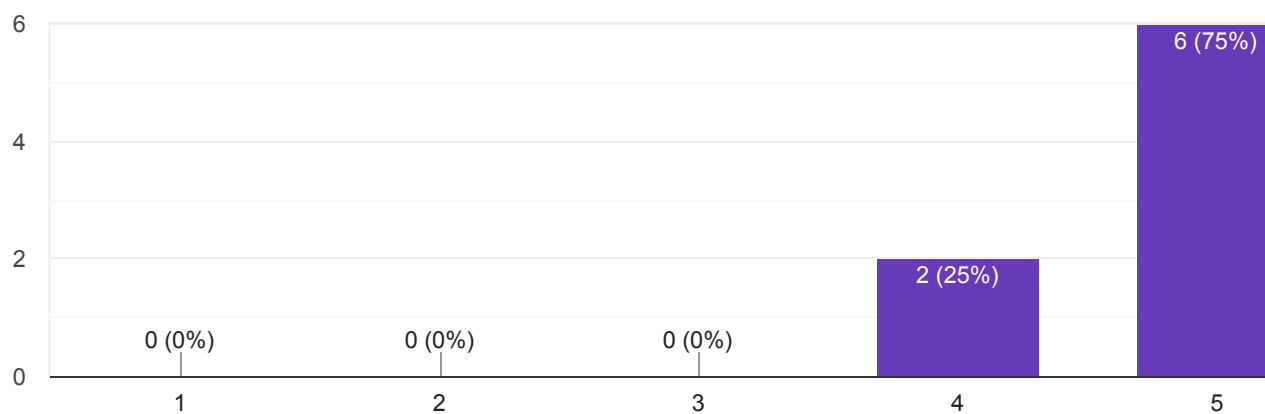
**Q15: A equipe tem flexibilidade na tomada de decisões que dizem respeito a implementação**

8 respostas

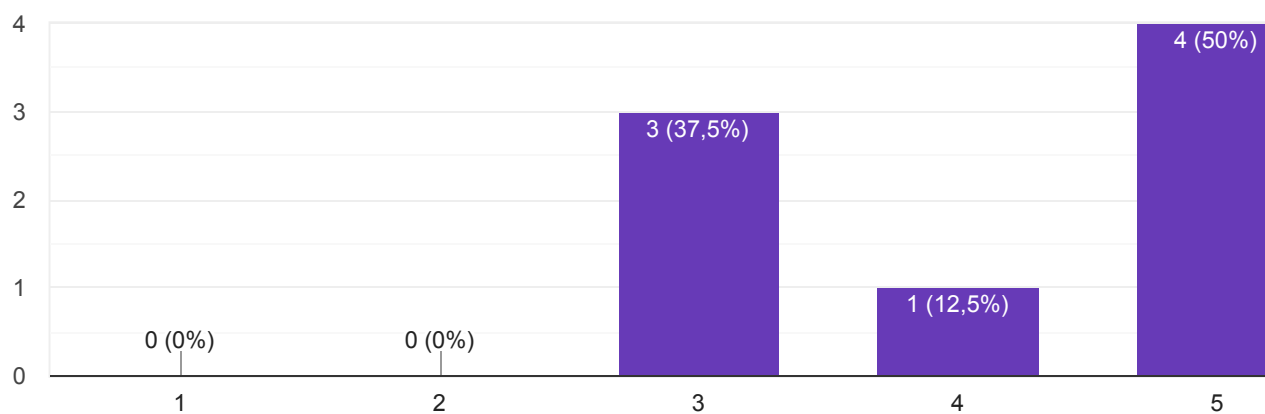
**Seção 4 - Eventos**

Q16: São feitas reuniões diárias para realizar o acompanhamento das atividades

8 respostas

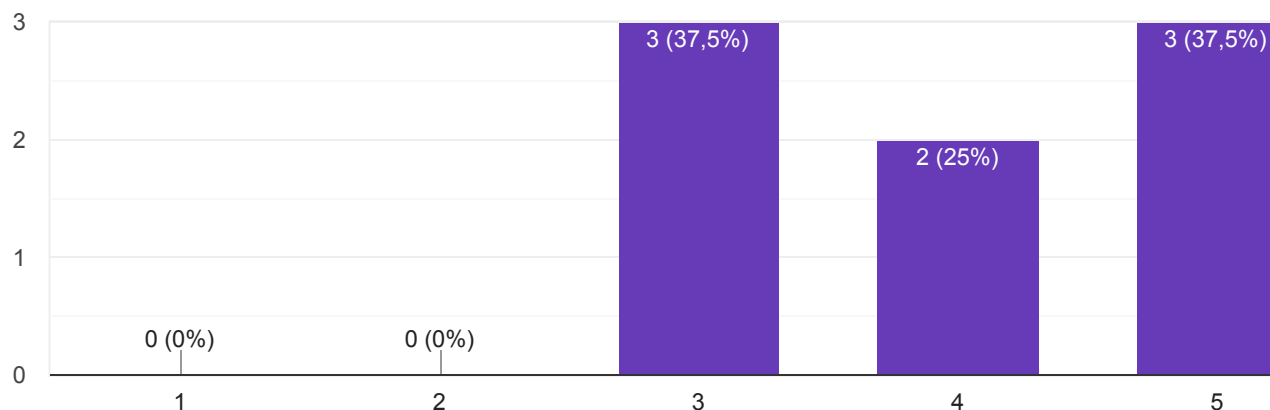
**Q17: Os impedimentos levantados nessas reuniões são rapidamente resolvidos**

8 respostas



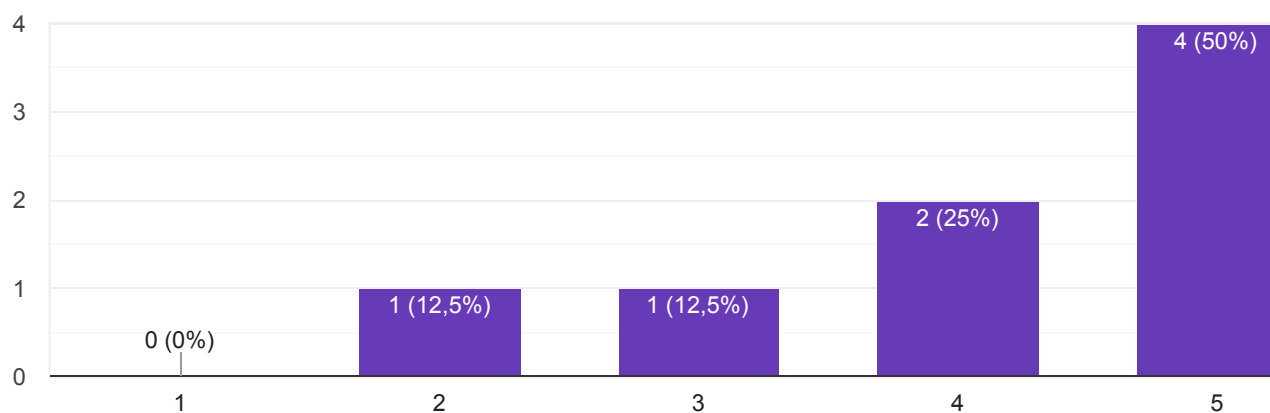
Q18: Ao final de cada Sprint, há uma reunião da equipe para apresentar o que foi desenvolvido

8 respostas



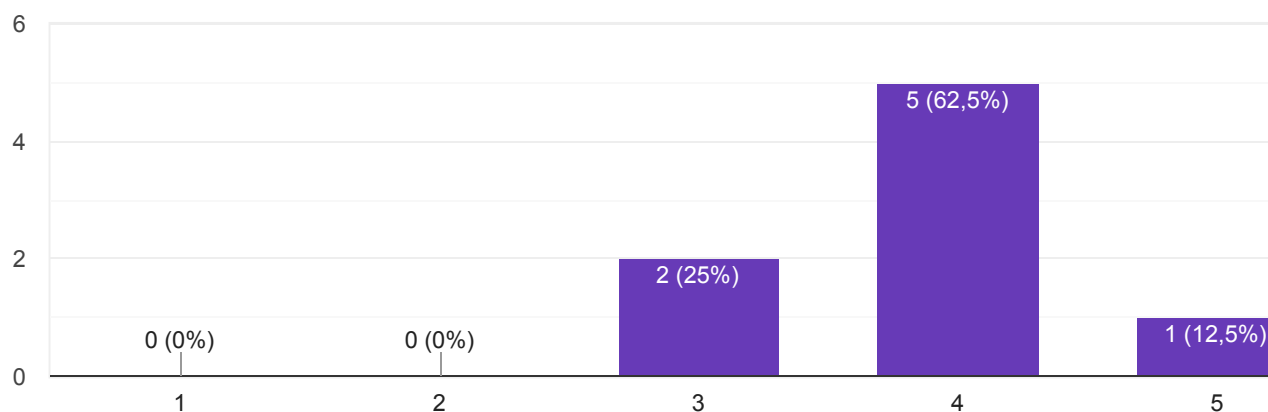
Q19: Ao final de cada Sprint, há uma reunião da equipe para apresentar os pontos positivos e negativos do Sprint

8 respostas



Q20: São realizadas validações com os stakeholders após as entregas incrementais

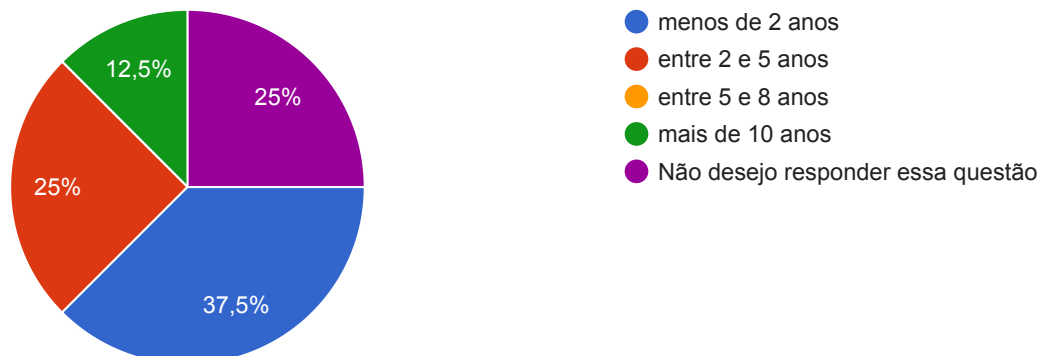
8 respostas



Seção 5 - Pessoas

Q21: Tempo que você trabalha com desenvolvimento de software

8 respostas



Q22: Liste quais metodologias de desenvolvimento de sistemas você conhece, indicando entre estas quais você já utilizou

8 respostas

pmbok e Scrum. Scrum a mais utilizada

Waterfall e Scrum

-

0

Scrum

Apenas o scrum

Scrum e programação extrema (xp)

Kanban, Scrum.

Q23: Quais os pontos fortes e fracos do Scrum você identifica em relação as outras metodologias que conhece?

8 respostas

Pontos fortes Agilidade e eficiencia. Fracos- dificuldade na aplicação

Fortes: flexibilidade, rapidez às mudanças, alto contato com os clientes, prioridades claras e definidas

Fracos: atividades fora do escopo prejudicam o andamento dos projetos (demandas inesperadas)

-

Acompanhamento frequente e transparência no decorrer do desenvolvimento.

Pontos forte: Processo iterativo e com bastante clareza do que está sendo desenvolvido.

Pontos fracos: Processo com mudanças rápidas e frequentes podem causar burnout e estresse

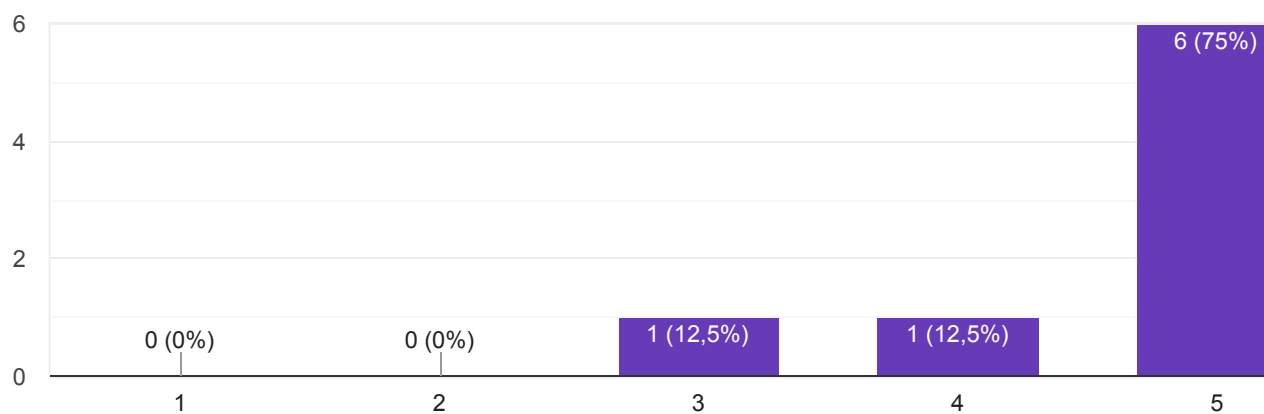
Pontos fortes: transparência; ter uma visão geral do que será e está sendo feito;

Pontos fracos: ideia de utilizar pouca documentação, pode levar a falta de informações

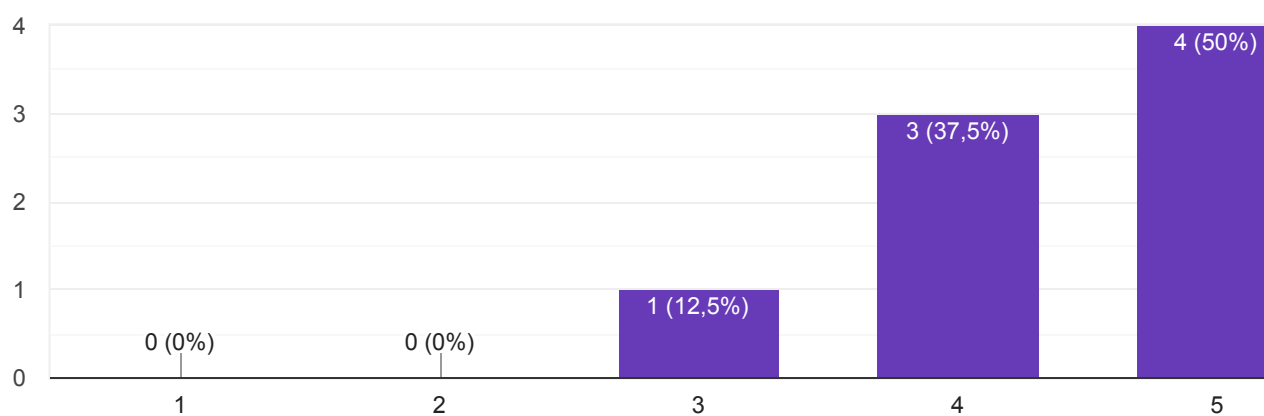
Falta de programação pareada

Q24: Como você avalia a transparência durante o projeto?

8 respostas

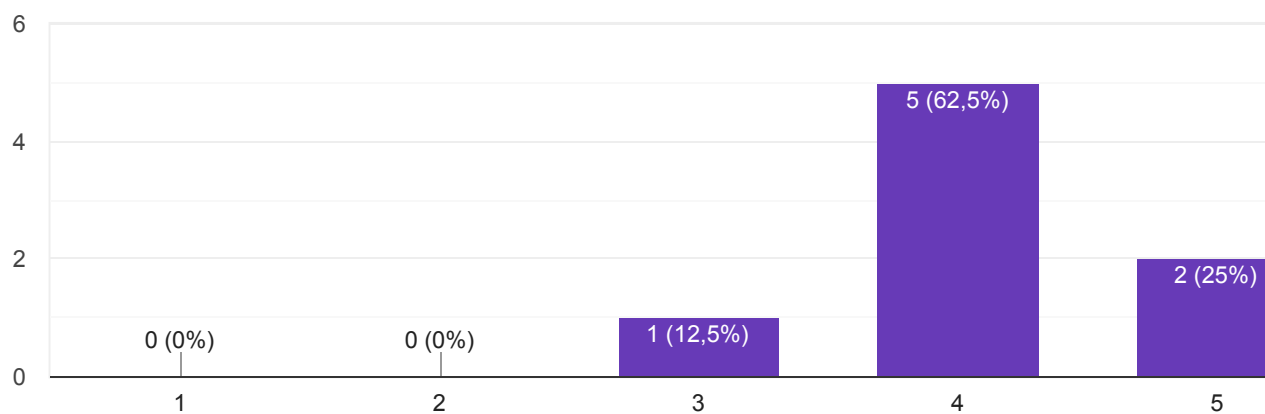
**Q25: Como você avalia a flexibilidade durante o projeto?**

8 respostas

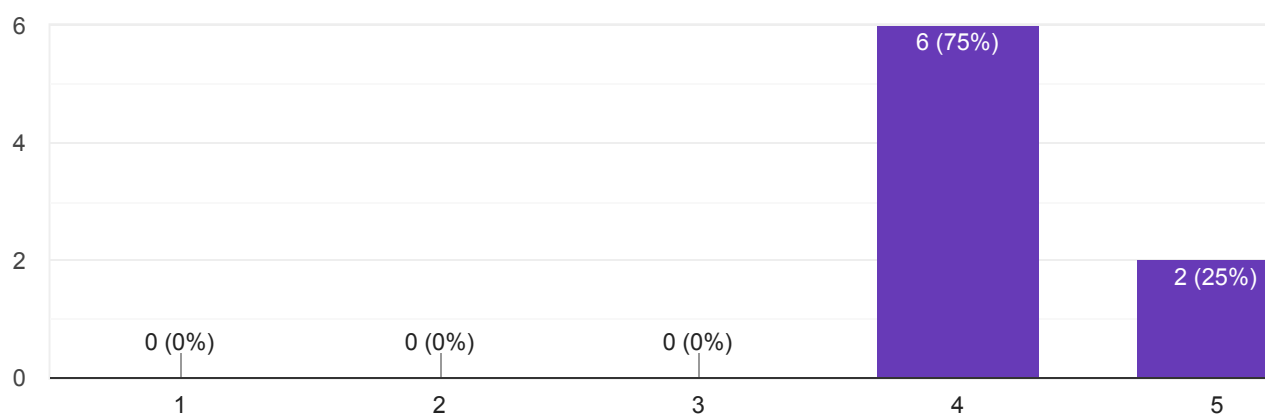


Q26: Como você avalia a eficiência do time durante o projeto?

8 respostas

**Q27: Como você avalia a sua satisfação durante o projeto?**

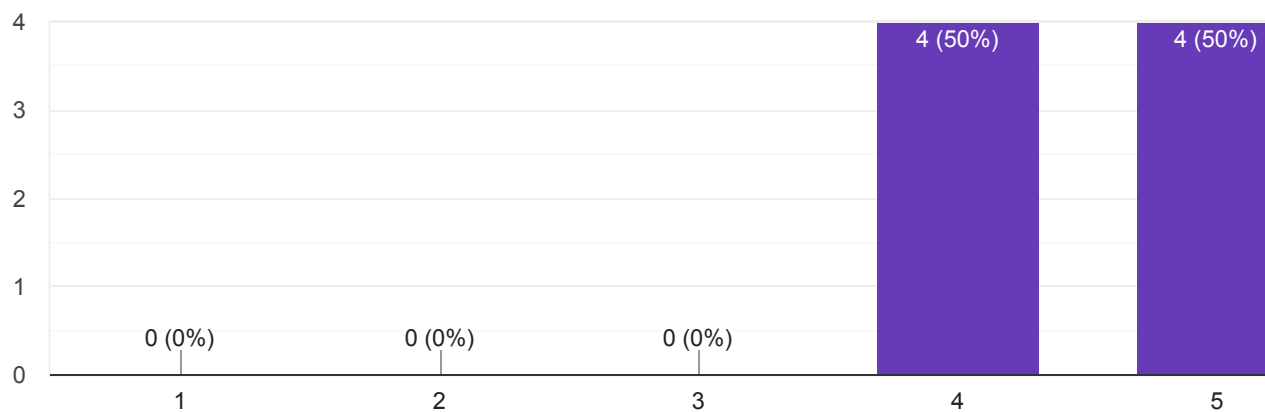
8 respostas



Seção final

Q28: Como você avalia o uso do Scrum como metodologia de desenvolvimento?

8 respostas

**Q29: Para conclusão do questionário, caso você queira justificar alguma das suas respostas ou complementar a pesquisa com alguma informação**

0 resposta

Ainda não há respostas para esta pergunta.

APÊNDICE B – RESPOSTAS INDIVIDUAIS

Tabela 11 – Colaborador 1

Questão	Resposta
Q1	5
Q2	4
Q3	4
Q4	4
Q5	4
Q6	4
Q7	4
Q8	4
Q9	4
Q10	4
Q11	4
Q12	4
Q13	4
Q14	3
Q15	4
Q16	4
Q17	3
Q18	4
Q19	4
Q20	4
Q21	mais de 10 anos
Q22	PMBOK e Scrum. Scrum a mais utilizada
Q23	Pontos fortes Agilidade e eficiência, e Pontos fracos dificuldade na aplicação
Q24	4
Q25	4
Q26	4
Q27	4
Q28	4
Q29	-

Tabela 12 – Colaborador 2

Questão	Resposta
Q1	5
Q2	5
Q3	5
Q4	3
Q5	5
Q6	5
Q7	4
Q8	5
Q9	4
Q10	5
Q11	5
Q12	5
Q13	5
Q14	2
Q15	5
Q16	5
Q17	5
Q18	5
Q19	5
Q20	4
Q21	entre 2 e 5 anos
Q22	Waterfall e Scrum
Q23	Pontos fortes flexibilidade, rapidez às mudanças, alto contato com os clientes, prioridades claras e definidas, e Pontos fracos atividades fora do escopo prejudicam o andamento dos projetos (demandas inesperadas)
Q24	5
Q25	5
Q26	4
Q27	5
Q28	5
Q29	-

Tabela 13 – Colaborador 3

Questão	Resposta
Q1	5
Q2	5
Q3	5
Q4	5
Q5	3
Q6	5
Q7	5
Q8	4
Q9	5
Q10	4
Q11	5
Q12	5
Q13	5
Q14	3
Q15	5
Q16	5
Q17	3
Q18	3
Q19	5
Q20	4
Q21	Não desejo responder essa questão
Q22	-
Q23	-
Q24	5
Q25	3
Q26	4
Q27	4
Q28	5
Q29	-

Tabela 14 – Colaborador 4

Questão	Resposta
Q1	4
Q2	4
Q3	3
Q4	4
Q5	3
Q6	5
Q7	4
Q8	5
Q9	4
Q10	3
Q11	5
Q12	4
Q13	4
Q14	3
Q15	5
Q16	4
Q17	4
Q18	4
Q19	4
Q20	3
Q21	Não desejo responder essa questão
Q22	0
Q23	Acompanhamento frequente e transparência no decorrer do desenvolvimento.
Q24	3
Q25	4
Q26	4
Q27	4
Q28	4
Q29	-

Tabela 15 – Colaborador 5

Questão	Resposta
Q1	4
Q2	5
Q3	5
Q4	2
Q5	5
Q6	5
Q7	4
Q8	3
Q9	3
Q10	5
Q11	5
Q12	5
Q13	5
Q14	5
Q15	5
Q16	5
Q17	5
Q18	3
Q19	3
Q20	3
Q21	menos de 2 anos
Q22	Scrum
Q23	Pontos forte: Processo iterativo e com bastante clareza do que está sendo desenvolvido. Pontos fracos: Processo com mudanças rápidas e frequentes podem causar burnout e estresse
Q24	5
Q25	5
Q26	5
Q27	5
Q28	5
Q29	-

Tabela 16 – Colaborador 6

Questão	Resposta
Q1	5
Q2	4
Q3	5
Q4	3
Q5	5
Q6	5
Q7	5
Q8	4
Q9	4
Q10	5
Q11	5
Q12	3
Q13	5
Q14	4
Q15	5
Q16	5
Q17	3
Q18	5
Q19	2
Q20	4
Q21	menos de 2 anos
Q22	Apenas o Scrum
Q23	Pontos fortes: transparência; ter uma visão geral do que será e está sendo feito; Pontos fracos: ideia de utilizar pouca documentação, pode levar a falta de informações
Q24	5
Q25	4
Q26	4
Q27	4
Q28	5
Q29	-

Tabela 17 – Colaborador 7

Questão	Resposta
Q1	5
Q2	5
Q3	5
Q4	4
Q5	5
Q6	5
Q7	5
Q8	5
Q9	5
Q10	4
Q11	4
Q12	3
Q13	5
Q14	2
Q15	5
Q16	5
Q17	5
Q18	3
Q19	5
Q20	4
Q21	entre 2 e 5 anos
Q22	Scrum e Programação extrema
Q23	Falta de programação pareada
Q24	5
Q25	5
Q26	3
Q27	4
Q28	4
Q29	-

Tabela 18 – Colaborador 8

Questão	Resposta
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	4
Q8	5
Q9	4
Q10	5
Q11	5
Q12	5
Q13	5
Q14	5
Q15	5
Q16	5
Q17	5
Q18	5
Q19	5
Q20	5
Q21	menos de 2 anos
Q22	Kanban e Scrum
Q23	Como pontos fortes vejo transparência do projeto e redução de falhas. Como pontos fracos vejo principalmente a visão do processo segmentada deixando algumas vezes difícil entender como o processo como todo se desenrolou
Q24	5
Q25	5
Q26	5
Q27	4
Q28	4
Q29	-