

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
MBA EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO**

**RICARDO HEYN JUNIOR**

**A INFLUÊNCIA DA CULTURA DEVOPS NO ENGAJAMENTO DE  
PROFISSIONAIS DE TECNOLOGIA EM CURITIBA**

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**CURITIBA**

**2018**

**RICARDO HEYN JUNIOR**

**A INFLUÊNCIA DA CULTURA DEVOPS NO ENGAJAMENTO DE  
PROFISSIONAIS DE TECNOLOGIA EM CURITIBA**

Monografia de Especialização, apresentada ao Curso de MBA em Gestão da Tecnologia da Informação e Comunicação, do Departamento Acadêmico de Eletrônica – DAELN, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Kleber Kendy Horikawa Nabas

**CURITIBA**

**2018**



## **TERMO DE APROVAÇÃO**

### **A INFLUÊNCIA DA CULTURA DEVOPS NO ENGAJAMENTO DE PROFISSIONAIS DE TECNOLOGIA EM CURITIBA**

por

**RICARDO HEYN JUNIOR**

Esta monografia foi apresentada em 28 de novembro de 2018 como requisito parcial para a obtenção do título de Especialista: MBA em Gestão de Tecnologia da Informação e Comunicação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

**Prof. Dr. Kleber Kendy Horikawa Nabas**  
Orientador

---

**Prof. Alexandre Jorge Miziara**  
Membro titular

---

**Prof. Dr. Kleber Kendy Horikawa Nabas**  
Membro titular

## RESUMO

HEYN JUNIOR, Ricardo. **A influência da cultura DevOps no engajamento de profissionais de tecnologia em Curitiba**. 2018. 40 f. Trabalho de Conclusão de Curso: MBA em Gestão da Tecnologia da Informação e Comunicação, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A presente monografia aborda o estudo das boas práticas DevOps e suas influências aos profissionais de tecnologia da informação na cidade de Curitiba. Com a utilização da metodologia DevOps, amplia-se a visão dos gestores que precisam controlar as entregas, melhorando o ciclo de vida dos serviços e minimizando falhas, como por exemplo, durante um processo de atualização do sistema. Portanto, o projeto inicia-se com a pesquisa sobre a metodologia DevOps aplicada, seguido de um estudo de campo feito com profissionais de tecnologia da informação.

**Palavras-chave:** Desafios ágeis. Entrega ágil. DevOps. Sistemas de gerenciamento de informações.

## ABSTRACT

HEYN JUNIOR, Ricardo. **The influence of the DevOps culture on the engagement of technology professionals in Curitiba**. 2018. 40 f. Trabalho de Conclusão de Curso: MBA em Gestão da Tecnologia da Informação e Comunicação, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This monograph deals with the study of DevOps good practices and their influence on information technology professionals in the city of Curitiba. By the use of the DevOps methodology, the vision of the managers who need to control deliveries is improved, improving the service life cycle and minimizing failures, for example: during a system upgrade process. Therefore, the project begins with the research on the applied DevOps methodology, followed by a case study performed with information technology professionals.

**Keywords:** Agile challenges. Agile delivery. DevOps. Information management systems.

## LISTA DE FIGURAS

Figura 1. Custos em função do tempo de desenvolvimento.....	12
Figura 2. Processo de desenvolvimento da metodologia Programação Extrema .....	15
Figura 3. Processo de desenvolvimento da metodologia Scrum.....	16
Figura 4. Marketplace para aplicativos DevOps .....	19
Figura 5. O crescimento de empresas com soluções Cloud entre o ano de 2017 e 2018 .....	20
Figura 6. Fluxo de execução do estudo de campo.....	22
Figura 7. Pesquisa de campo: Área de atuação na tecnologia .....	24
Figura 8. Pesquisa de campo: Quantidade de integrantes da equipe.....	25
Figura 9. Pesquisa de campo: Alinhamento/explicação do gestor sobre os processos utilizados .....	25
Figura 10. Pesquisa de campo: Disponibilidade do gestor para conversar e tirar dúvidas.....	26
Figura 11. Pesquisa de campo: Possibilidade de propor melhorias ao gestor .....	27
Figura 12. Pesquisa de campo: Fornecimento de feedback constante .....	27
Figura 13. Pesquisa de campo: Reuniões individuais com o gestor .....	28
Figura 14. Pesquisa de campo: Duração da reunião diária do time .....	28
Figura 15. Pesquisa de campo: Dificuldade de interação/comunicação entre as equipes de desenvolvimento, suporte, QA e infraestrutura.....	29
Figura 16. Pesquisa de campo: Conhecimento sobre metodologias ágeis .....	29
Figura 17. Pesquisa de campo: Utiliza ou pretende utilizar alguma metodologia ágil .....	30
Figura 18. Pesquisa de campo: Time de qualidade para teste de novas funcionalidades adicionadas ao software .....	31
Figura 19. Pesquisa de campo: Existência do Product Owner ou divisão de papel entre os gestores.....	31
Figura 20. Pesquisa de campo: Utilização de Sprints .....	32
Figura 21. Pesquisa de campo: Conhecimento sobre a cultura DevOps .....	32
Figura 22. Pesquisa de campo: Utilização de soluções Cloud Computing nas empresas.....	33
Figura 23. Pesquisa de campo: Utilização de provedores de Cloud computing nas empresas.....	33
Figura 24. Pesquisa de campo: Softwares para tratamento de casos/tickets/incidentes/bugs .....	34
Figura 25. Pesquisa de campo: Utilização de centralizadores de códigos.....	34
Figura 26. Pesquisa de campo: Utilização de frameworks para desenvolvimento .....	35

Figura 27. Pesquisa de campo: Utilização de gerenciadores de repositório de software.....	35
Figura 28. Pesquisa de campo: Utilização de ferramentas de integração contínua ..	36
Figura 29. Pesquisa de campo: Utilização de ferramentas de entrega e/ou deploy contínuo .....	36
Figura 30. Pesquisa de campo: Utilização de ferramentas de testes de automação	37
Figura 31. Pesquisa de campo: Utilização de ferramentas de monitoramento de servidores e rede.....	37

## LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CRC	Classe, Responsabilidade, Colaboração
IBM	<i>International Business Machines</i>
IU	Interface do Usuário
KISS	<i>Kepp It Simple, Stupid</i> (mantenha simples, idiota)
PaaS	<i>Platform as a Service</i>
QA	<i>Quality Assurance</i>
SaaS	<i>Software as a Service</i>



## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>9</b>
1.1 TEMA .....	9
1.2 OBJETIVOS .....	9
1.2.1 Objetivo Geral .....	9
1.2.2 Objetivos Específicos .....	10
1.3 JUSTIFICATIVA .....	10
1.4 PROCEDIMENTOS METODOLÓGICOS .....	10
<b>2 REFERENCIAIS TEÓRICOS .....</b>	<b>12</b>
2.1 METODOLOGIAS ÁGEIS .....	12
2.1.1 PROGRAMAÇÃO EXTREMA .....	13
2.1.2 SCRUM .....	15
2.2 DEVOPS NA PRÁTICA .....	17
2.3 PRINCÍPIOS DO TIME SCRUM .....	18
2.4 TECNOLOGIAS EM DEVOPS E AS MAIS UTILIZADAS .....	19
<b>3 ESTUDO DE CAMPO .....</b>	<b>22</b>
<b>4 RESULTADOS .....</b>	<b>24</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>38</b>
<b>REFERÊNCIAS .....</b>	<b>39</b>

## 1 INTRODUÇÃO

Neste capítulo serão tratados os elementos introdutórios relacionados ao estudo de boas práticas utilizando a metodologia DevOps.

### 1.1 TEMA

Com a competitividade no setor de tecnologia, os profissionais de tecnologia da informação precisam se atualizar tecnicamente cada vez mais, com maior eficiência, pois assim garantem a qualidade técnica, utilizando as últimas tecnologias do mercado, provendo um melhor trabalho para a empresa e inclusive garantindo maiores chances de empregabilidade. Além disso, com o aumento da robustez das aplicações nas empresas, existe a necessidade de profissionais cada vez mais especializados. Para que as empresas e colaboradores possam continuar evoluindo com base nas novas tecnologias, o ideal é iniciar um estudo das metodologias aplicáveis que podem melhorar os processos e que suportem todas as demandas. No caso da metodologia DevOps, as empresas e profissionais devem ter ciência de que várias mudanças poderão ocorrer a fim de se obter o êxito na inclusão da cultura DevOps.

Aplicando uma metodologia ágil, diversas equipes de uma determinada empresa que antes trabalhavam de forma separada se unem formando uma corrente organizada para vencer os obstáculos diários.

### 1.2 OBJETIVOS

Nesta sessão será abordado o objetivo geral e objetivos específicos.

#### 1.2.1 Objetivo Geral

As influências da cultura DevOps aos profissionais de tecnologia da informação.

### 1.2.2 Objetivos Específicos

Realizar uma pesquisa com no mínimo 30 e no máximo 100 profissionais de tecnologia da informação situadas na cidade de Curitiba e capturar os dados atuais, realizar comparações, e apresentá-las em um gráfico. Demonstrando, a partir de pesquisa, as vantagens de profissionais que utilizam a cultura DevOps, os ganhos para os mesmos e relacionando com os ganhos diretos e indiretos nas empresas e possíveis dificuldades que possam ser encontradas nas mudanças que devem ser levadas em consideração. Criar um gráfico de comparação com grau de dificuldade, grau de ganho e mostrar os principais resultados.

### 1.3 JUSTIFICATIVA

Normalmente as empresas possuem mais de um serviço ou produto e, geralmente após a sua disponibilização, as melhorias são requisitadas, e essas por sua vez dependem de profissionais cada vez mais capacitados, que consigam suprir as demandas, e se organizando para que as empresas tenham ganhos e até mesmo vantagens competitivas, porém sabemos que podem ocorrer contratemplos até a publicação no ambiente de produção, levando prejuízo a empresa e ao cliente. Os problemas podem ser evitados e resolvidos mais rapidamente por meio da implantação de processos ágeis e utilização da cultura DevOps.

### 1.4 PROCEDIMENTOS METODOLÓGICOS

Baseando-se no conhecimento de Gil (2002) sobre a classificação e os objetivos de cada pesquisa, esta monografia tem embasamento nos procedimentos técnicos de pesquisa bibliográfica e materiais já elaborados em livros e artigos científicos.

Segundo Gil (2002), a principal vantagem da pesquisa bibliográfica está no fato de permitir ao investigador a cobertura de uma gama de fenômenos muito mais ampla do que aquela que poderia pesquisar diretamente (GIL, 2002, p. 44-45). Em relação ao estudo de campo, por ser definido procura muito mais o aprofundamento das questões propostas do que a distribuição das características da população segundo determinadas variáveis. Como consequência, o planejamento do estudo de

campo apresenta uma flexibilidade muito maior, podendo ocorrer mesmo que seus objetivos sejam reformulados ao longo da pesquisa. Além disso, o fato de demonstrar essas informações, possibilita a caracterização precisa de seus segmentos (GIL, 2002, p. 53).

## 2 REFERENCIAIS TEÓRICOS

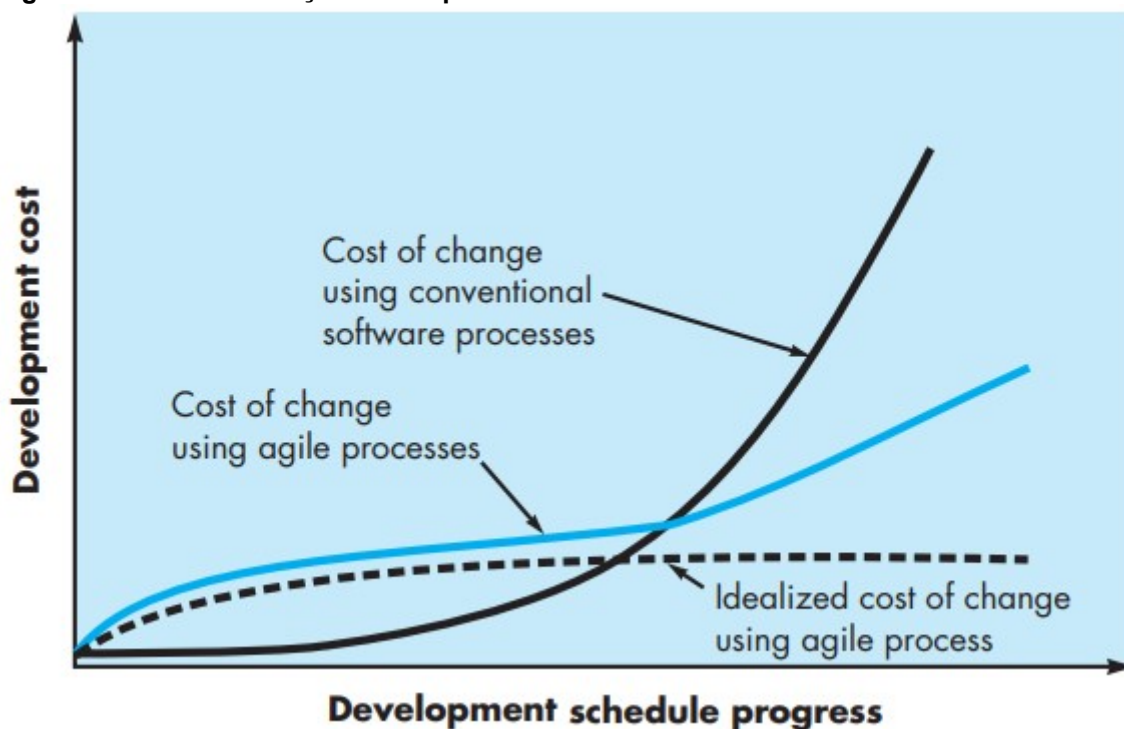
No presente capítulo serão apresentados os conceitos relacionados às metodologias ágeis, conceitos e práticas cruciais sobre a cultura DevOps, princípios do time Scrum e tecnologias DevOps.

### 2.1 METODOLOGIAS ÁGEIS

As empresas buscam inovação no desenvolvimento de seus softwares e um dos fatores para que obtenham sucesso é a busca pelas palavras “Ágil” e “DevOps”, são essas as tendências globais que estão em constante adaptação e melhoria. A engenharia de software ágil começou a ser utilizada devido a velocidade de implantação e ao aumento na satisfação do cliente. O planejamento é melhor aplicado e existe uma facilidade no desenvolvimento geral, se tornando por muitas vezes uma melhor alternativa ao método tradicional (PRESSMAN; MAXIM, 2014).

No quesito mudança de custo de desenvolvimento, a Figura 1 representa a diferença entre o método tradicional e método ágil, conforme observa-se nas linhas.

Figura 1. Custos em função do tempo de desenvolvimento



Fonte: Pressman e Maxim (2014).

O custo no desenvolvimento tradicional é elevado pois existe um grande prazo de desenvolvimento até a entrega do produto, já no caso do método ágil, como a entrega ocorre de semanas em semanas, o custo é reduzido.

### 2.1.1 PROGRAMAÇÃO EXTREMA

Uma das metodologias utilizadas hoje em dia em empresas é a Programação Extrema, definida primeiramente pelo planejamento, iniciando a coleta de requisitos para a compreensão da necessidade comercial do software, recursos e funcionalidades. A estória então é escrita pelo cliente, atribuindo um valor de prioridade, sempre baseando no valor geral do negócio, recurso e função. Em seguida, o membro da Programação Extrema estima um custo médio de desenvolvimento em semanas. Caso o desenvolvimento da estória esteja estimado em mais de três semanas, o cliente é convidado a dividir em tarefas menores e então novamente ocorre a análise de estimativa por parte da equipe de Programação Extrema e assim por diante. Dessa forma, os desenvolvedores trabalham juntamente com os clientes para definir os próximos lançamentos de recursos e funcionalidades do software a ser desenvolvido, lembrando de sempre incluir a data de entrega e detalhes do projeto. Seguindo essa linha, as estórias serão desenvolvidas das seguintes maneiras:

- Todas as estórias serão implementadas em algumas semanas;
- As estórias com maior valor serão movidas no horário e implementadas primeiro;
- As estórias mais arriscadas serão movidas no cronograma e implementadas separadamente.

Então, calcula-se a velocidade da implementação do projeto com base no número de estórias entregues no primeiro lançamento, modificando o planejamento para que o trabalho de desenvolvimento continue, inclusive o cliente pode adicionar novas estórias ou retirá-las.

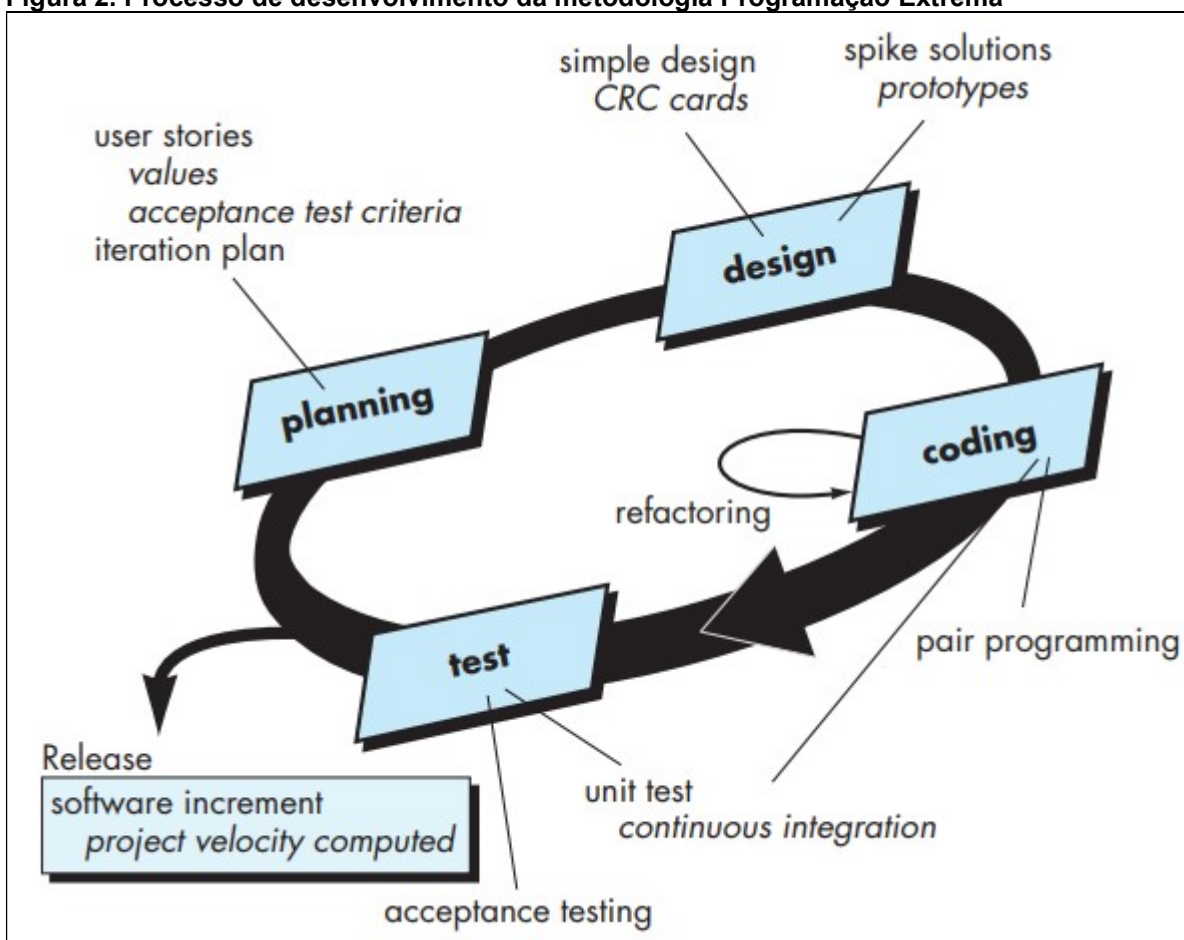
A fase do desenho segue rigorosamente o princípio KISS (*Keep It Simple, Stupid*), pois é sempre preferido em uma representação menos complexa, fornecendo representação para a estória que está em implementação. A Programação Extrema incentiva o uso de cartões CRC (Classe, Responsabilidade,

Colaboração), incentivando o pensamento sobre o software em um contexto orientado a objetos, sendo parte do processo de Programação Extrema. Em uma determinada estória, caso um problema de design for de difícil solução, a Programação Extrema recomenda criar imediatamente um protótipo operacional dessa parcela do design, implementado e avaliado, com o intuito de reduzir o risco não prejudicando as validações e estimativas originais.

Uma parte importante no desenvolvimento do sistema é que a refatoração é incentivada pelo time, o qual não altera o comportamento do programa, melhorando sua estrutura interna, em outras palavras, limpando o código e minimizando as chances de erro que ocorrem por meio das modificações realizadas no código. Esse processo deve ocorrer de maneira contínua conforme o desenvolvimento do sistema. Dando continuidade no processo, ocorrem os testes unitários no sistema de cada estória desenvolvida, que será incluída na versão nova de incremento do software. Após os testes unitários terem sucesso, proporciona um feedback instantâneo para os desenvolvedores. Outro conceito importante na Programação Extrema é a programação em pares, que é o trabalho de duas pessoas juntas em uma estação de trabalho para criar código para uma estória, fornecendo um mecanismo para resolução de problemas em tempo real, como diz o ditado “duas cabeças pensam melhor que uma”, garantindo uma ação mais rápida dos desenvolvedores, pois o código é revisado pelo par à medida que o programa é criado, mantendo os desenvolvedores focados no problema.

A Figura 2 ilustra o processo da Programação Extrema no desenvolvimento até a entrega do incremento de software, conforme descrito anteriormente.

Figura 2. Processo de desenvolvimento da metodologia Programação Extrema



Fonte: Pressman e Maxim (2014).

### 2.1.2 SCRUM

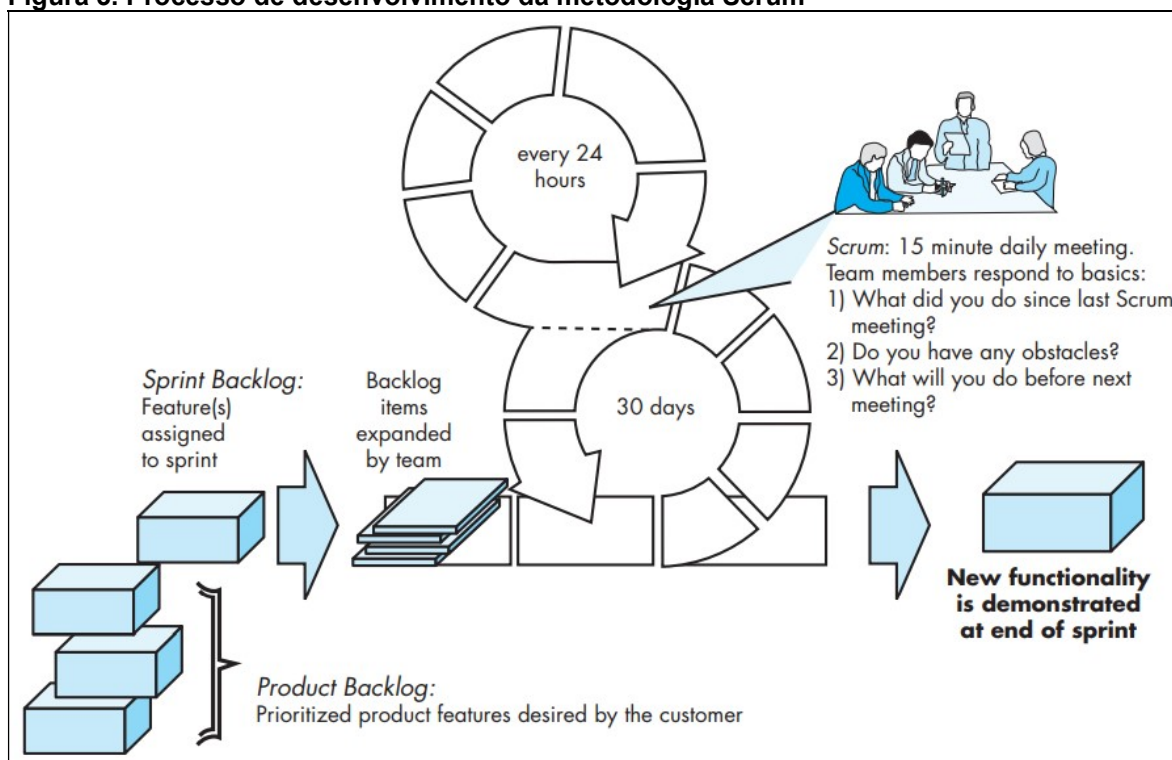
No caso do Scrum, é um método ágil de desenvolvimento de software que foi concebido por Jeff Sutherland e sua equipe de desenvolvimento no início da década de 1990. Nos últimos anos, um maior desenvolvimento nos métodos Scrum foi realizado por Sutherland e Schwaber (2016). Os princípios do Scrum são consistentes ao manifesto ágil e são usados para orientar atividades de desenvolvimento dentro de um processo que incorpora as atividades de requisitos, análise, design, evolução e entrega.

Conforme ilustra a Figura 3, os processos que ocorrem em cada uma dessas atividades são chamados de Sprints, que dependerão da complexidade e tamanho do produto a ser desenvolvido. Utiliza-se processos de software padronizados que são eficazes em projetos e cronogramas apertados, e cada uma dessas atividades fazem parte, primeiramente citando o *backlog*, que é uma lista de requisitos ou



recursos do projeto que contém priorização e fornece valor comercial para o cliente e pode ser atualizado a qualquer momento pelo gerente de produto que prioriza os itens do *backlog*. Para organizar o *backlog* é utilizado o sistema de Sprints, que é a divisão em semanas de trabalho do time de desenvolvimento e qualidade, garantindo a entrega em determinado período, permitindo inclusive que a equipe trabalhe em entregas de curto prazo e com ambiente estável. Nesse período são realizadas também as reuniões de Scrum, geralmente com duração de 15 minutos diários pelo time Scrum, e nestas reuniões são feitas as seguintes perguntas a todos os integrantes da equipe: O que você fez desde a última reunião da equipe? Quais obstáculos você está encontrando? O que você planeja realizar na próxima reunião da equipe? Essas perguntas são respondidas de forma rápida para que outro membro da equipe possa responder às mesmas perguntas. Um líder da equipe, chamado de mestre Scrum, conduz a reunião avaliando as respostas de cada participante da reunião diária. O encontro auxilia na descoberta dos problemas o mais cedo possível e promove uma estrutura de equipe auto organizada. Por fim, o software demo, que é a entrega do software, contém os requisitos e funcionalidades para que o cliente avalie, ou seja, é uma demonstração de um incremento do software desejado pelo cliente (PRESSMAN; MAXIM, 2014).

**Figura 3. Processo de desenvolvimento da metodologia Scrum**



Fonte: Pressman e Maxim (2014).

Além dos processos acima, destaca-se a primeira iniciativa popular, o Manifesto Ágil, visão de Alistair Cockburn, Jim Highsmith, Craig Larman dentre outras 17 pessoas, que se reuniram nos Estados Unidos e definiram alguns valores adicionais e superiores no desenvolvimento do software: Indivíduos e interações mais que processos e ferramentas, software em funcionamento mais que documentação abrangente, colaboração com o cliente mais que negociação de contratos e responder a mudanças mais que seguir um plano (AGILE MANIFESTO, 2001). Não significando que processos, documentação, contratos e planos sejam descartados pelas metodologias ágeis, mas sim que eles têm uma importância secundária no processo de desenvolvimento. Os métodos ágeis podem ser vistos como uma referência no mercado de TI, porém empresas de outros ramos estão utilizando em seus projetos, ou seja, já está definido que é uma resposta natural de mercado. Com isso, as empresas podem se concentrar em prover mais valor aos seus clientes, concluindo em poucos dias projetos que antes demoravam meses. Essa eficiência é visível nos tempos modernos e a cada ano aumenta a utilização dos métodos ágeis (BASTOS, 2013).

Uma grande vantagem das metodologias ágeis é levar o conceito de feedback constante adiante, pois a qualidade do software aumenta a cada ciclo, conforme a comunicação com os *stakeholders* dos projetos, que são os clientes, colaboradores e gestores. Diferentemente da metodologia tradicional, quando o sistema só era entregue após estar 100% concluído, com as metodologias ágeis pequenas entregas são realizadas, assim o cliente consegue testar a aplicação e propor melhorias.

## 2.2 DEVOPS NA PRÁTICA

Para se obter uma ampliação dos produtos, serviços e a necessidade de entregar rapidamente as atualizações dos sistemas de tecnologia, a cultura DevOps tem um papel importante na metodologia ágil e deve ser implantada gradativamente, conduzindo a união entre desenvolvimento e operações de tecnologia da informação, sendo vantajosa a automatização de tarefas, a transformação de ambientes em ambientes escaláveis, a melhora dos indicadores de eficiência e até mesmo trazer redução de custos para as empresas (GAEA CONSULTING, 2017).

Sendo um processo importante a ser definido, a entrega e a melhoria contínua são possíveis graças a nuvem, e é uma prática que é desenvolvida e aplicada quando ocorre qualquer alteração de código, liberando automaticamente para o ambiente de produção após testes do time DevOps (GILL et al., 2017).

A estratégia de integração contínua é de muita importância, pois ajuda a evitar incompatibilidade, e para acelerar os testes unitários no sistema, recomenda-se que a estrutura do código seja feita de forma que permita automatização, para que o sistema seja testado mais rapidamente. Esses testes podem descobrir erros no sistema implementado, assim o time pode rapidamente corrigir o erro no mesmo dia, conforme Wells afirma, “Reparar pequenos problemas a cada poucas horas leva menos tempo do que reparar problemas enormes antes do prazo final”. E em relação aos testes de aceitação conforme a programação extrema, são especificados pelo cliente e o foco é sempre em recursos e funcionalidades do sistema, com base nas estórias desenvolvidas e testadas pelo time. Os artefatos criados podem ser implantados e utilizados no processo de teste padronizado, descobrindo problemas e erros antes de ser disponibilizado para produção, incluindo também os testes de IU (Interface do Usuário), carga, integração, API (*Application Programming Interface*), entre outros (SMITH, 2017).

### 2.3 PRINCÍPIOS DO TIME SCRUM

Os papéis de cada integrante do time Scrum são de importante definição, sendo eles: dono do produto, time de desenvolvimento e Scrum Master.

O papel do dono do produto é maximizar o valor do produto e trabalhar com o time de desenvolvimento, porém isto pode variar dependendo da empresa. É um trabalho de gerenciamento individual, e envolve detalhar, categorizar e priorizar o *backlog* do produto, e principalmente deixá-lo o mais claro possível para o time de desenvolvimento, respondendo aos desejos do comitê.

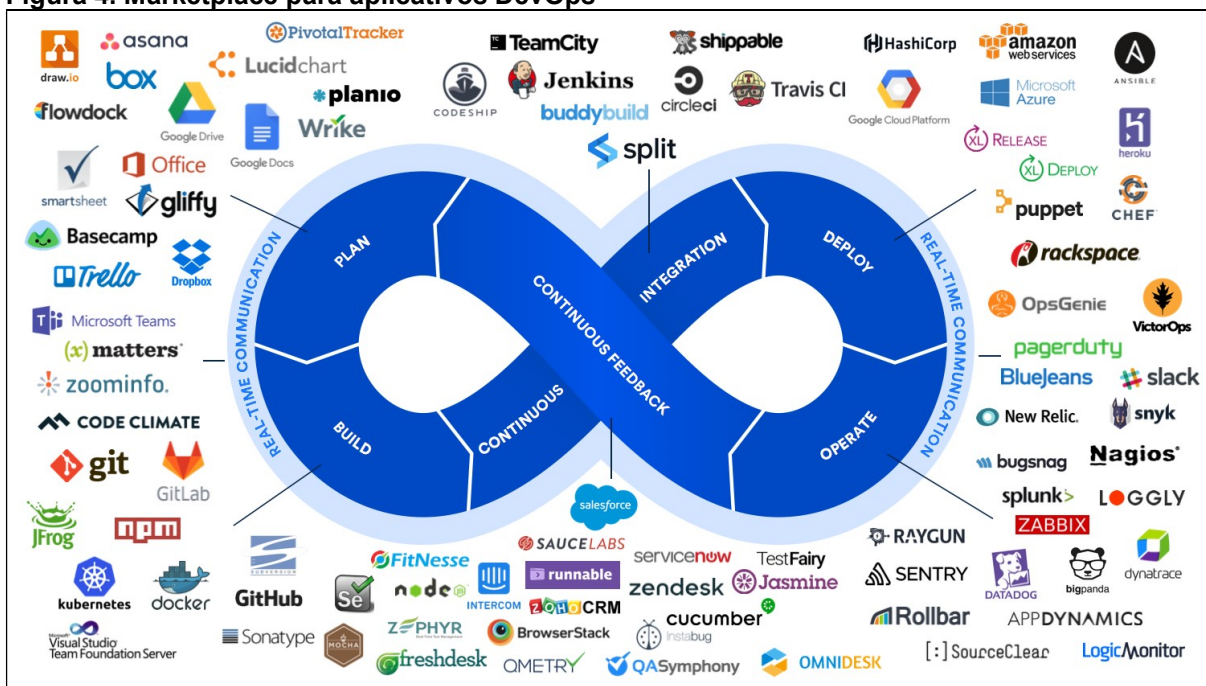
A equipe de desenvolvimento inclui pessoas capacitadas tecnicamente na criação do incremento de *backlog*, criando os releases no final de cada Sprint. Elas devem ser treinadas para gerenciar seu próprio trabalho, pois no Scrum o time de desenvolvimento é autogerenciável e multifuncional, ou seja, cada integrante tem a habilidade necessária para trabalhar em um item do *backlog* de produto.

Já no caso do Scrum Master, ele é responsável por garantir que o Scrum seja entendido de forma correta, incluindo as teorias, práticas e regras do Scrum. Esta pessoa mostra às outras equipes fora do time Scrum quais suas relações com o time Scrum, com o intuito de maximizar os valores apresentados (SUTHERLAND; SCHWABER, 2016).

## 2.4 TECNOLOGIAS EM DEVOPS E AS MAIS UTILIZADAS

Existe uma enorme variedade de softwares disponíveis utilizados diariamente pelo time de tecnologia, ferramentas complexas ou de simples utilização e configuração, algumas possuem livre licença e outras são pagas. Conforme a Figura 4, cada ferramenta faz parte de um ou mais processos no desenvolvimento e entrega do software.

Figura 4. Marketplace para aplicativos DevOps

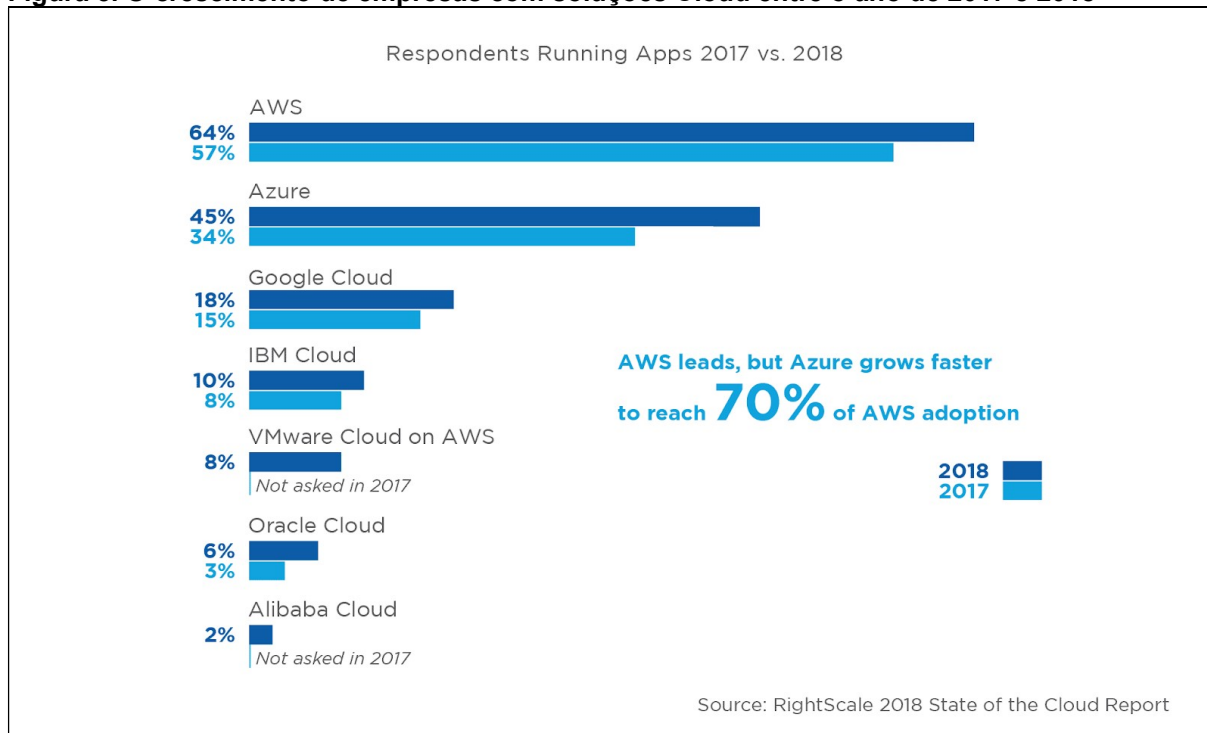


Fonte: Atlassian (2018).

Todas as ferramentas devem ser muito bem estudadas antes de serem utilizadas em produção, até mesmo especialistas devem ser cogitados para configuração. Sendo assim, os times unidos devem conhecer os fundamentos das metodologias ágeis e as ferramentas utilizadas na empresa (SIMPLILEARN SOLUTIONS, 2018). Além disso, existem muitas empresas que vendem soluções *Cloud Computing* (Figura 5), ajudando na implantação de soluções PaaS (*Plataforma*

as a Service) e SaaS (*Software as a Service*), como é o caso da Amazon, Azure, Google Cloud, IBM Cloud.

**Figura 5. O crescimento de empresas com soluções Cloud entre o ano de 2017 e 2018**



Fonte: Right Scale (2018).

Além de decidir sobre os provedores e ferramentas, é importante definir com a equipe qual a linguagem ou framework de desenvolvimento que será utilizado. Alguns frameworks prontos têm o objetivo de otimizar, acelerar o desenvolvimento e estruturar o código deixando-o mais legível e organizado, quebrando a complexidade do código (ZANINOTTO; POTENCIER, 2007). Existem inúmeros frameworks e a escolha depende da linguagem de programação utilizada e os benefícios ganhos. Alguns exemplos: Node / Express, Laravel, Django, ASP.NET Core, Elixir + Phoenix, dentre outros (SIMONS et al., 2018).

Uma das ferramentas em DevOps mais utilizadas é o Git, que é um sistema de revisão distribuído com um conjunto de vários comandos que fornece uma operação de alto nível e acesso total ao time de desenvolvimento (GIT-SCM, 2018). É possível integrar o Git com ferramentas de controle, como por exemplo o GitLab, que fornece soluções para todos os estágios do ciclo de vida DevOps: planejar, criar, verificar, empacotar, liberar, configurar, monitorar (GITLAB, 2018).

A ferramenta Jenkins é utilizada para integração contínua e automatização de software para entrega futura. Com esse software é possível automatizar tarefas no

processo de desenvolvimento de software, automatizando tarefas como: compilar código-fonte e executar testes simples, além disso, é possível integrá-lo com outras ferramentas de testes. Existem outras ferramentas como o CircleCI, Apache Maven e Microsoft Team Foundation Server (G2CROWD, 2018).

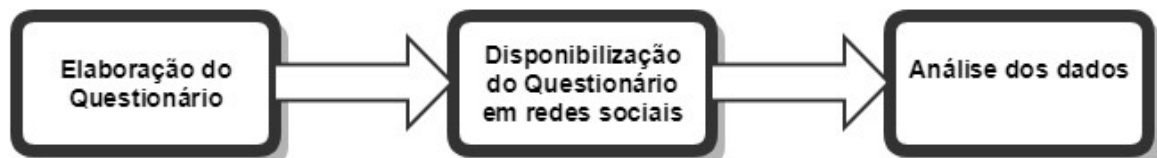
A ferramenta Selenium, utilizada para testes de automação na web, é uma das mais populares, também sendo de licença livre, suportando múltiplos sistemas operacionais e navegadores (SELENIUMHQ, 2018). Dando sequência aos testes, temos a ferramenta de entrega contínua SaltStack, personalizável de acordo com a necessidade da aplicação, esse software também é utilizado como gerenciamento de configurações, mantendo as configurações dos servidores organizadas e padronizadas (SALTSTACK, 2018).

No próximo capítulo é apresentado o método e o questionário elaborados para os profissionais de tecnologia da informação de Curitiba.

### 3 ESTUDO DE CAMPO

Com base no capítulo referenciais teóricos, foi elaborada uma pesquisa de campo que possui como objetivo demonstrar a realidade dos profissionais de tecnologia da informação de Curitiba, a respeito do engajamento sobre as metodologias ágeis e principalmente sobre a cultura DevOps. Além disso, buscar como as empresas levam em consideração os processos ágeis e a cultura DevOps de forma prática, respondendo à questão: “Profissionais de tecnologia da informação de Curitiba estão engajados sobre a cultura DevOps e utilizam processos ágeis nas empresas?”. A Figura 6 ilustra o fluxo das atividades.

Figura 6. Fluxo de execução do estudo de campo



Fonte: Autoria própria.

A primeira etapa do trabalho, elaboração do questionário, foi realizada levando em consideração a pesquisa conceitual sobre Metodologias Ágeis e cultura DevOps, dividida em seções:

- Perguntas iniciais: obtenção de informações dos dados dos participantes contendo questões relativas ao nome, e-mail, área de atuação no mercado de tecnologia, quantidade de integrantes na equipe, cidade em que trabalha e o nome da empresa. As informações pessoais e os nomes das empresas não foram apresentados.
- Interação com gestor e time: obtenção de informações sobre o comportamento dos gestores, interação entre o funcionário e o gestor, interação entre os membros da equipe e se processos são seguidos na empresa.
- Metodologia ágil: obtenção de informações sobre o conhecimento dos participantes a respeito de metodologias ágeis, se a empresa utiliza alguma metodologia ágil, se possui time de qualidade para testes em softwares, se existe algum *Product Owner* e especificamente se a equipe de desenvolvimento e qualidade trabalham com Sprints.

- Desenvolvimento e operações: apresentado a Figura 4 sobre o Marketplace DevOps como exemplos de ferramentas DevOps. Obtenção de informações dos participantes sobre os conhecimentos da cultura DevOps, se é utilizado algum provedor de *Cloud Computing*, sistema para tratamento de casos/incidentes, centralizador de código, *framework* para desenvolvimento, gerenciador de repositório de software, ferramenta de integração contínua, ferramenta de entrega e *deploy* contínuo, ferramenta de teste de automação, monitoramento de servidores e rede nas empresas.

Após o término da criação das perguntas, foram escolhidos três participantes para analisar o tempo de entrega das respostas, evitar desistências, e resultando em um tempo aceitável de aproximadamente 05 minutos para cada participante finalizar o questionário.

O link do questionário foi disponibilizado nas redes sociais com uma quantidade de aceitação de análise mínima de 30 e máxima de 100 participantes. A aceitação de respostas ficou aberta por uma semana.

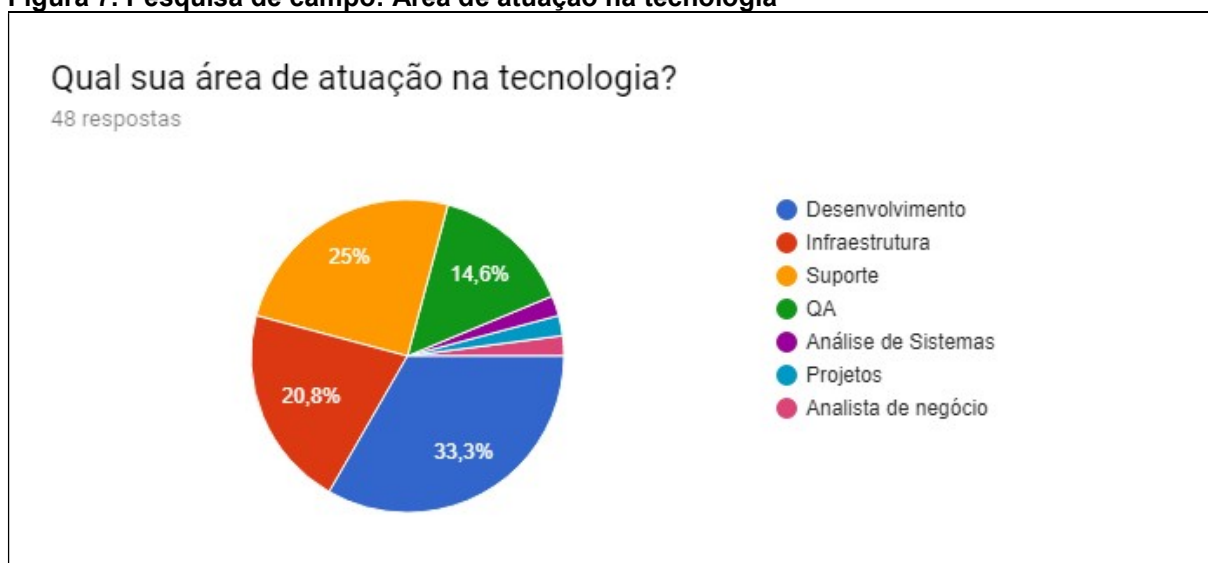


## 4 RESULTADOS

Com o objetivo de responder a problemática “Profissionais de tecnologia da informação de Curitiba estão engajados sobre a cultura DevOps e utilizam processos ágeis nas empresas?”, foram analisados os dados de 54 participantes e excluídos 06 devido ao limite de escopo de pesquisa regional, totalizando 48 participantes. Os conteúdos das respostas são apresentados nas Figuras de número 7 até 31.

Os resultados para a pergunta “Qual sua área de atuação na tecnologia?”, apresentados na Figura 7, demonstram que com 33,3% das respostas, a maioria dos participantes da pesquisa atuam na área de Desenvolvimento, seguido de Suporte com 25%, Infraestrutura com 20,8%, QA (*Quality Assurance*) com 14,6%, Análise de Sistemas com 2,1%, Projetos com 2,1% e Analista de negócio com 2,1%.

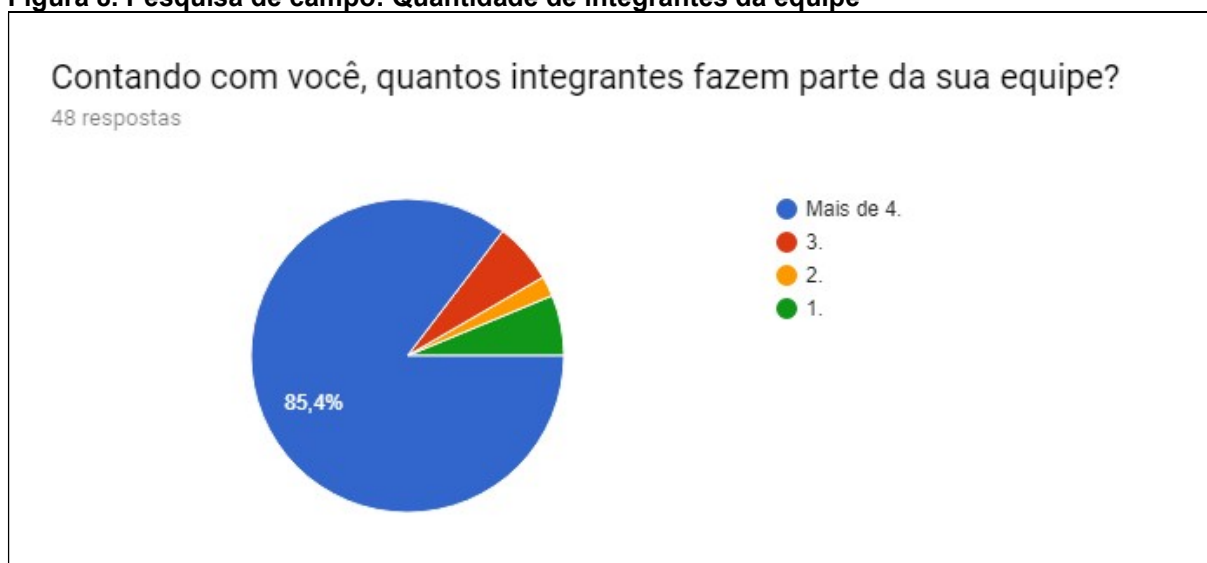
**Figura 7. Pesquisa de campo: Área de atuação na tecnologia**



Fonte: Autoria própria.

Os resultados para a pergunta “Contando com você, quantos integrantes fazem parte da sua equipe?”, apresentados na Figura 8, demonstram que com 85,4% das respostas, a maioria dos participantes trabalha com uma equipe de mais de 4 pessoas, seguido de 6,3% que trabalham com 3 integrantes na equipe, 6,3% que trabalham com 1 integrante na equipe e 2,1% que trabalham com 2 integrantes na equipe.

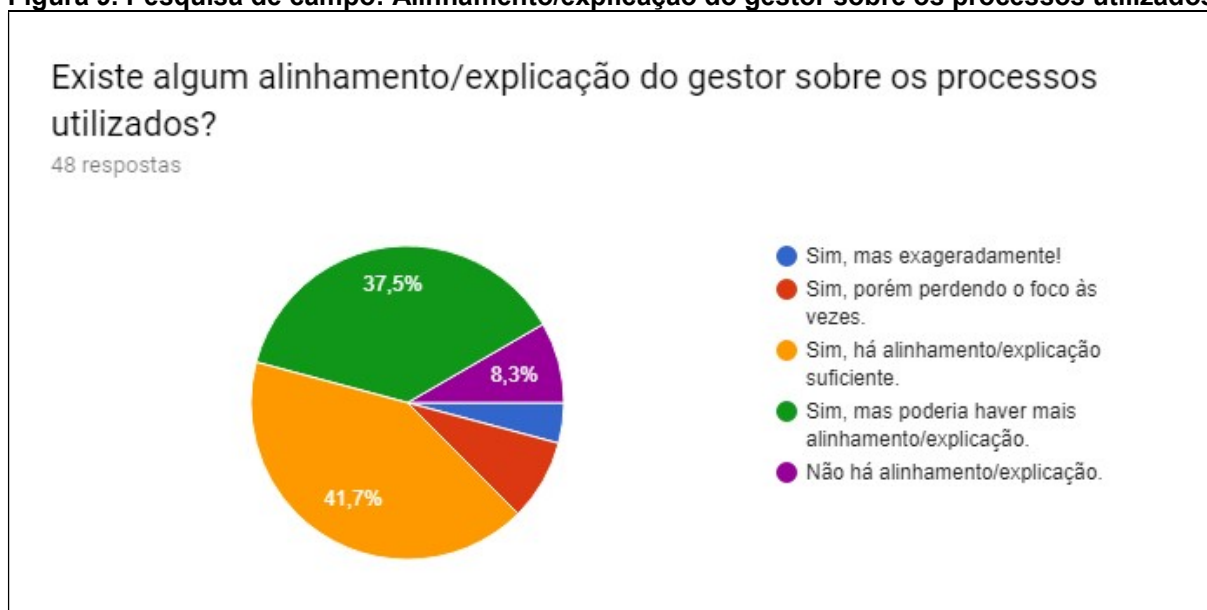
**Figura 8. Pesquisa de campo: Quantidade de integrantes da equipe**



Fonte: Autoria própria.

Os resultados para a pergunta “Existe algum alinhamento/explicação do gestor sobre os processos utilizados?”, apresentados na Figura 9, demonstram que com 41,7% das respostas, a maioria dos participantes consideram que “Sim, há alinhamento/explicação suficiente”, seguido de 37,5% considerando que “Sim, mas poderia haver mais alinhamento/explicação”, 8,3% consideram que “Não há alinhamento/explicação”, 8,3% consideram que “Sim, porém perdendo o foco às vezes” e 4,2% consideram que “Sim, mas exageradamente!”.

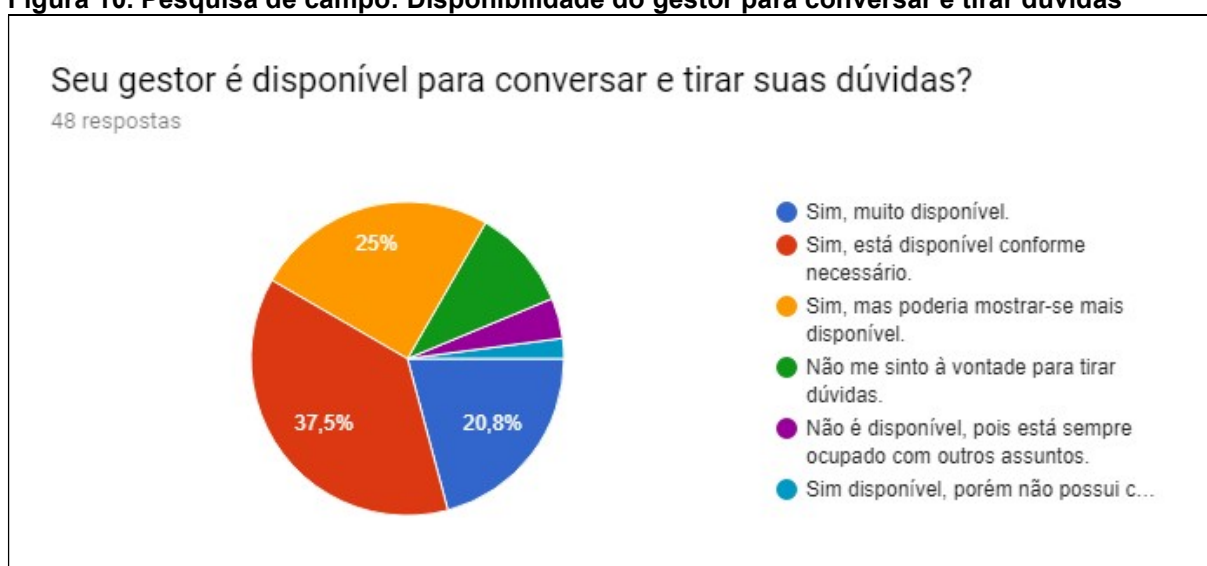
**Figura 9. Pesquisa de campo: Alinhamento/explicação do gestor sobre os processos utilizados**



Fonte: Autoria própria.

Os resultados para a pergunta “Seu gestor é disponível para conversar e tirar suas dúvidas?”, apresentados na Figura 10, demonstram que com 37,5% das respostas, a maioria dos participantes consideram que “Sim, está disponível conforme necessário”, 25% consideram que “Sim, mas poderia mostrar-se mais disponível”, 20,8% consideram que “Sim, muito disponível”, 10,4% consideram que “Não me sinto à vontade para tirar dúvidas”, 4,2% consideram que “Não é disponível, pois está sempre ocupado com outros assuntos” e 2,1% consideram que “Sim disponível, porém não possui conhecimento para tirar dúvidas”.

**Figura 10. Pesquisa de campo: Disponibilidade do gestor para conversar e tirar dúvidas**



Fonte: Autoria própria.

Os resultados para a pergunta “Seu gestor lhe possibilita propor melhorias?”, apresentados na Figura 11, demonstram que com 60,4% das respostas, a maioria dos participantes consideram que “Sim, possibilita propor melhorias conforme necessário”, 20,8% consideram que “Sim, mas não auxilia muito na implantação de melhorias”, 16,7% consideram que “Não me sinto à vontade para propor melhorias”, 2,1% consideram que “Ele não gosta de melhorias pois gera mais trabalho” e nenhum participante respondeu “Não consigo propor melhorias, ele está sempre muito ocupado”.

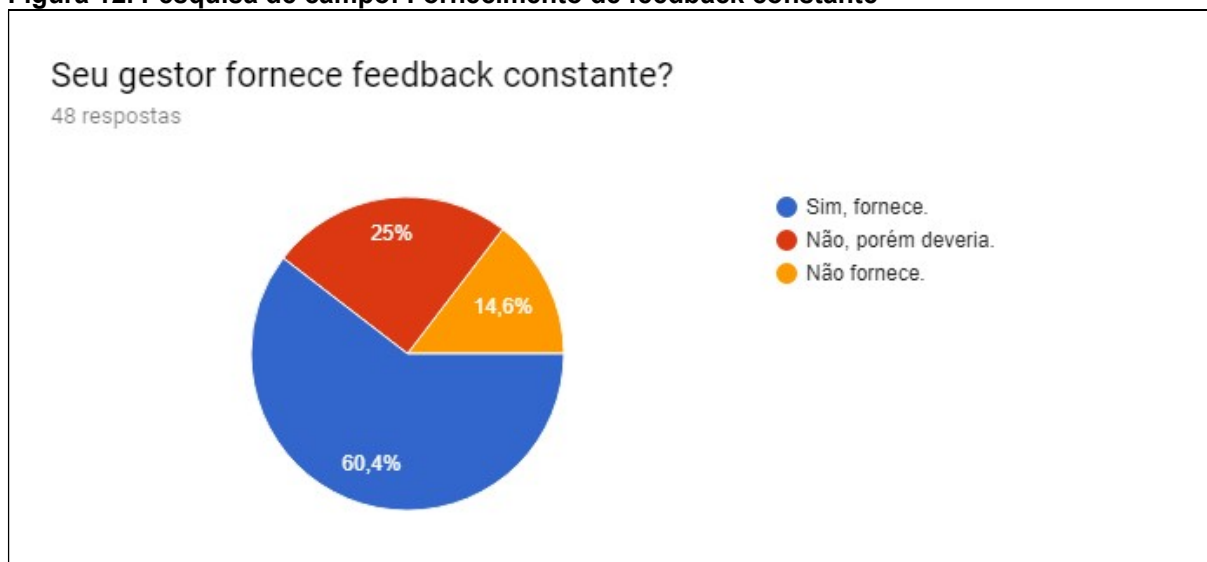
**Figura 11. Pesquisa de campo: Possibilidade de propor melhorias ao gestor**



Fonte: Autoria própria.

Os resultados para a pergunta “Seu gestor fornece feedback constante?”, apresentados na Figura 12, demonstram que com 60,4% das respostas, a maioria dos participantes consideram que “Sim, fornece”, 25% consideram que “Não, porém deveria” e 14,6% consideram que “Não fornece”.

**Figura 12. Pesquisa de campo: Fornecimento de feedback constante**

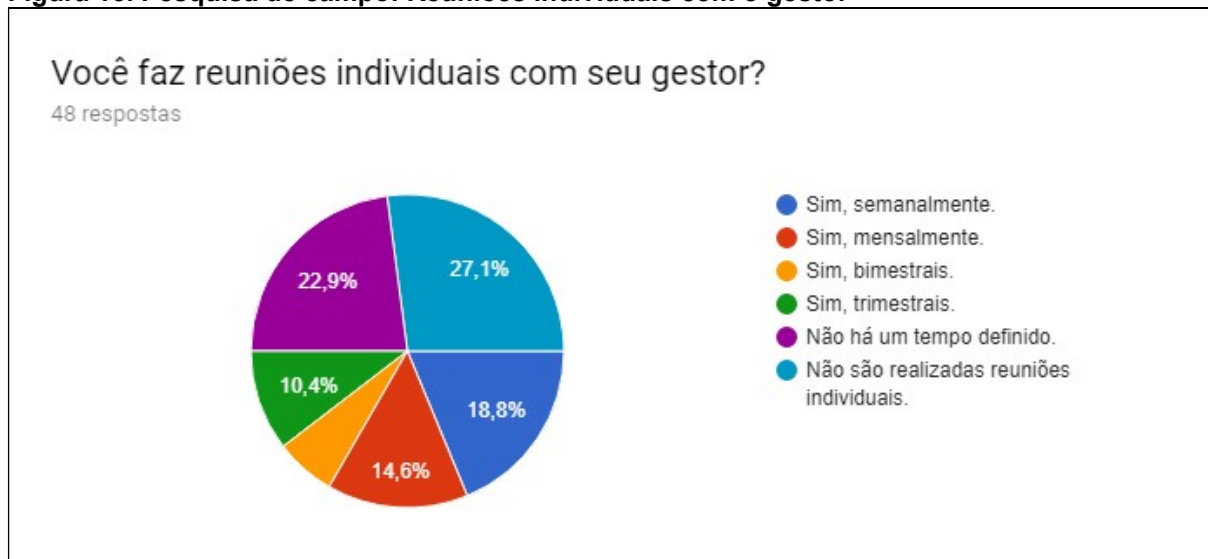


Fonte: Autoria própria.

Os resultados para a pergunta “Você faz reuniões individuais com seu gestor?”, apresentados na Figura 13, demonstram que com 27,1% das respostas, a maioria dos participantes consideram que “Não são realizadas reuniões individuais”, 22,9% consideram que “Não há um tempo definido”, 18,8% consideram que “Sim,

semanalmente”, 14,6% consideram que “Sim, mensalmente”, 10,4% consideram que “Sim, trimestrais” e 6,3% consideram que “Sim, bimestrais”.

**Figura 13. Pesquisa de campo: Reuniões individuais com o gestor**



Fonte: Autoria própria.

Os resultados para a pergunta “Qual é a duração da reunião diária de seu time?”, apresentados na Figura 14, demonstram que com 37,5% das respostas, a maioria dos participantes consideram que “15 minutos”, 29,2% consideram que “Não são realizadas reuniões diárias”, 18,8% consideram que “Não há um tempo definido”, 12,5% consideram que “30 minutos” e 2,1% consideram que “01 hora”.

**Figura 14. Pesquisa de campo: Duração da reunião diária do time**

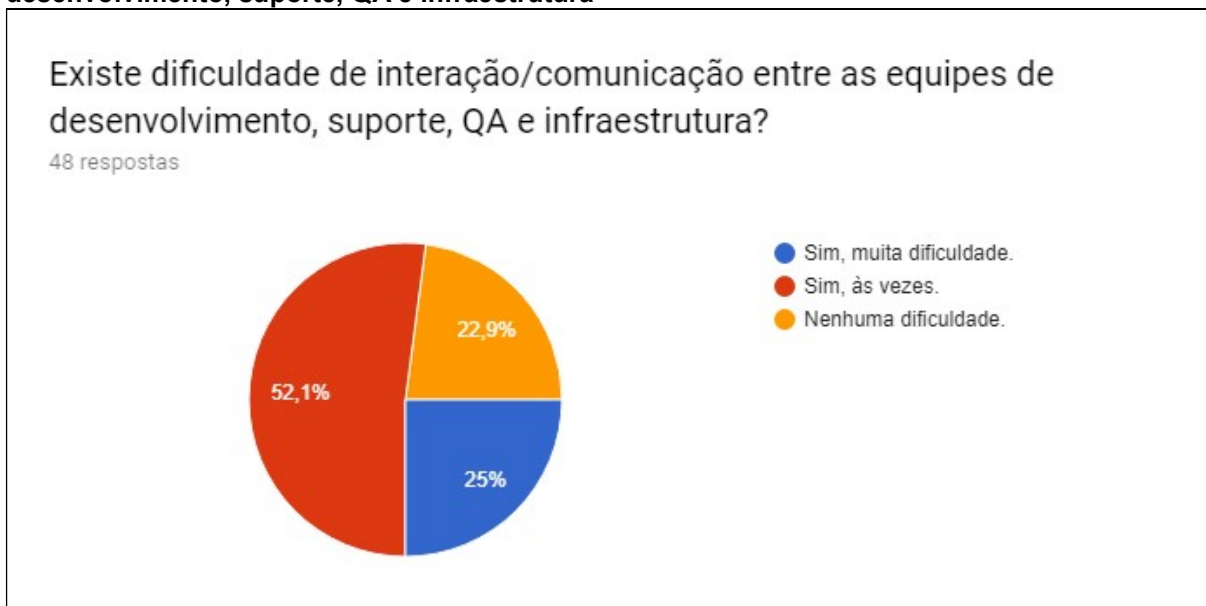


Fonte: Autoria própria.

Os resultados para a pergunta “Existe dificuldade de interação/comunicação entre as equipes de desenvolvimento, suporte, QA e infraestrutura?”, apresentados

na Figura 15, demonstram que com 52,1% das respostas, a maioria dos participantes consideram que “Sim, às vezes”, 25% consideram que “Sim, muita dificuldade” e 22,9% consideram que “Nenhuma dificuldade”.

**Figura 15. Pesquisa de campo: Dificuldade de interação/comunicação entre as equipes de desenvolvimento, suporte, QA e infraestrutura**



Fonte: Autoria própria.

Os resultados para a pergunta “Você conhece as metodologias ágeis?”, apresentados na Figura 16, demonstram que com 81,3% das respostas, a maioria dos participantes consideram que “Sim”, 16,7% consideram que “Não” e 2,1% consideram que “Não tenho certeza”.

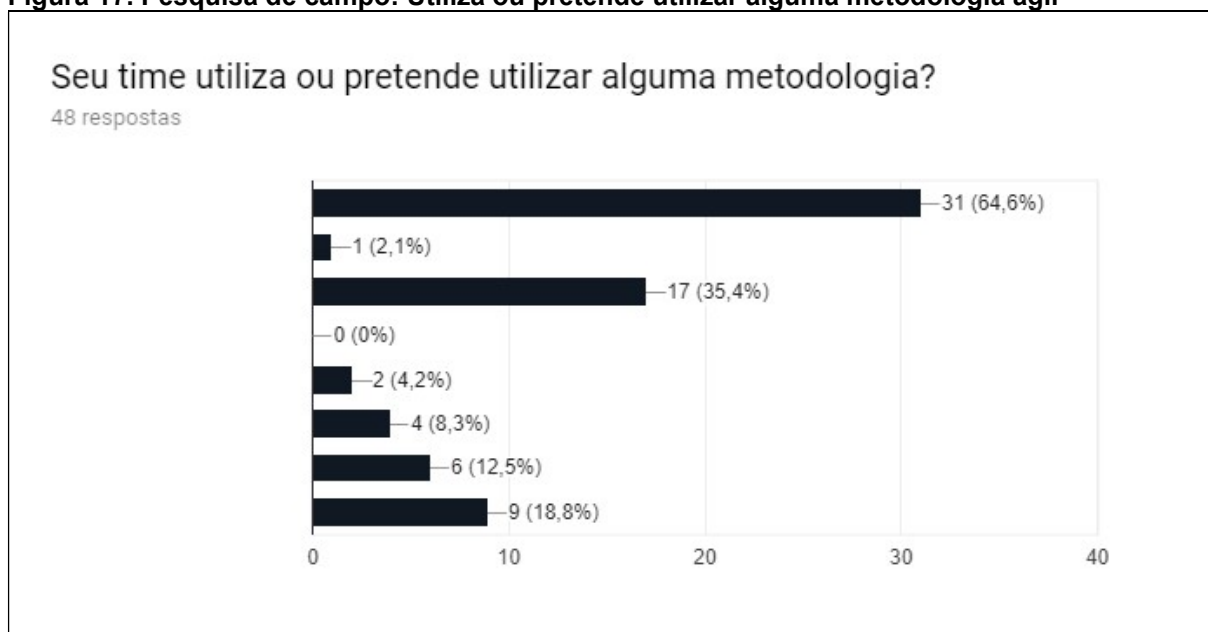
**Figura 16. Pesquisa de campo: Conhecimento sobre metodologias ágeis**



Fonte: Autoria própria.

Os resultados para a pergunta “Seu time utiliza ou pretende utilizar alguma metodologia?”, apresentados na Figura 17, demonstram que com 64,6% das respostas, a maioria dos participantes utilizam “Scrum”, 35,4% utilizam “Kanban”, 18,8% consideram que “Não tenho certeza”, 12,5% consideram que “Não”, 8,3% utilizam “Lean Software Development (LSD)”, 4,2% utilizam “Feature-Driven Development (FDD)” e 2,1% utilizam “Programação Extrema”.

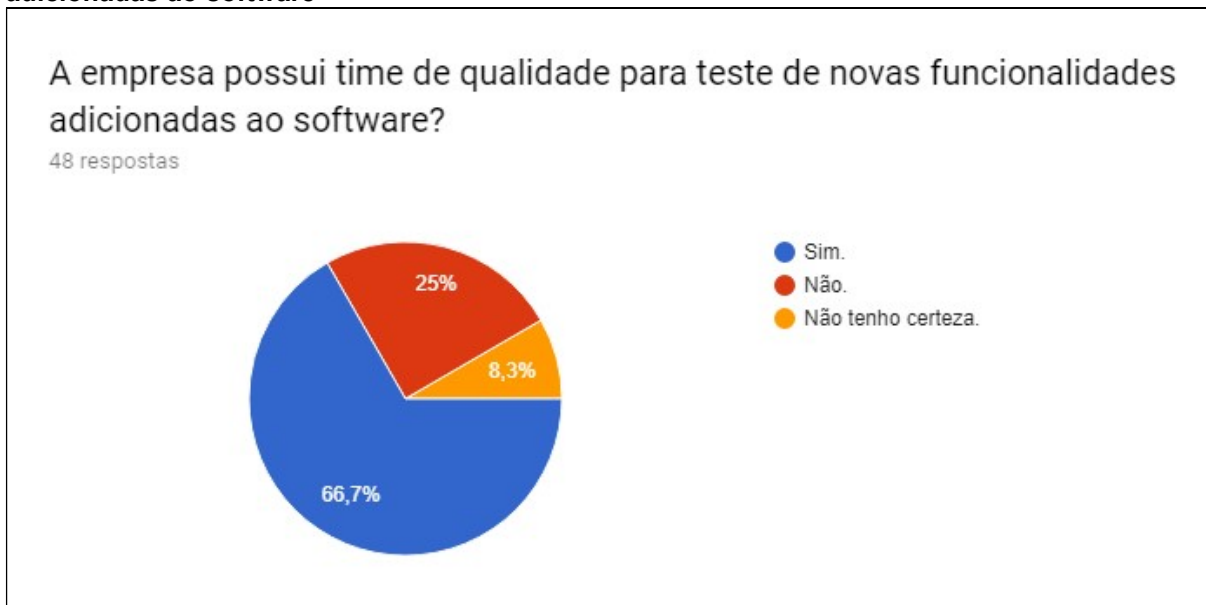
**Figura 17. Pesquisa de campo: Utiliza ou pretende utilizar alguma metodologia ágil**



Fonte: Autoria própria.

Os resultados para a pergunta “A empresa possui time de qualidade para teste de novas funcionalidades adicionadas ao software?”, apresentados na Figura 18, demonstram que com 66,7% das respostas, a maioria dos participantes consideram que “Sim”, 25% consideram que “Não” e 8,3% consideram que “Não tenho certeza”.

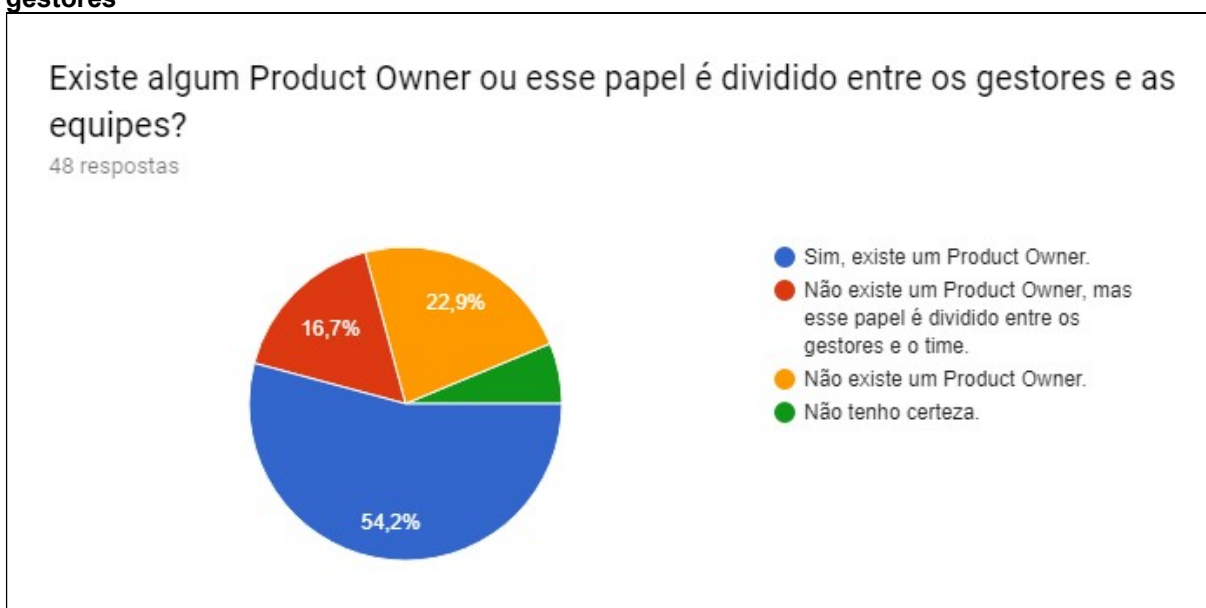
**Figura 18. Pesquisa de campo: Time de qualidade para teste de novas funcionalidades adicionadas ao software**



Fonte: Autoria própria.

Os resultados para a pergunta “Existe algum Product Owner ou esse papel é dividido entre os gestores e as equipes?”, apresentados na Figura 19, demonstram que com 54,2% das respostas, a maioria dos participantes consideram que “Sim, existe um Product Owner”, 22,9% consideram que “Não existe um Product Owner”, 16,7% consideram que “Não existe um Product Owner, mas esse papel é dividido entre os gestores e o time” e 6,3% consideram que “Não tenho certeza”.

**Figura 19. Pesquisa de campo: Existência do Product Owner ou divisão de papel entre os gestores**



Fonte: Autoria própria.



Os resultados para a pergunta “A equipe de desenvolvimento e qualidade trabalham com Sprints?”, apresentados na Figura 20, demonstram que com 31,3% das respostas, a maioria dos participantes consideram que “Sim, mesma Sprint para as equipes de desenvolvimento e qualidade”, 27,1% consideram que “Sim, Sprints separadas, uma para a equipe de desenvolvimento e outra para a equipe de qualidade”, 22,9% consideram que “Não tenho certeza” e 18,8% consideram que “Não”.

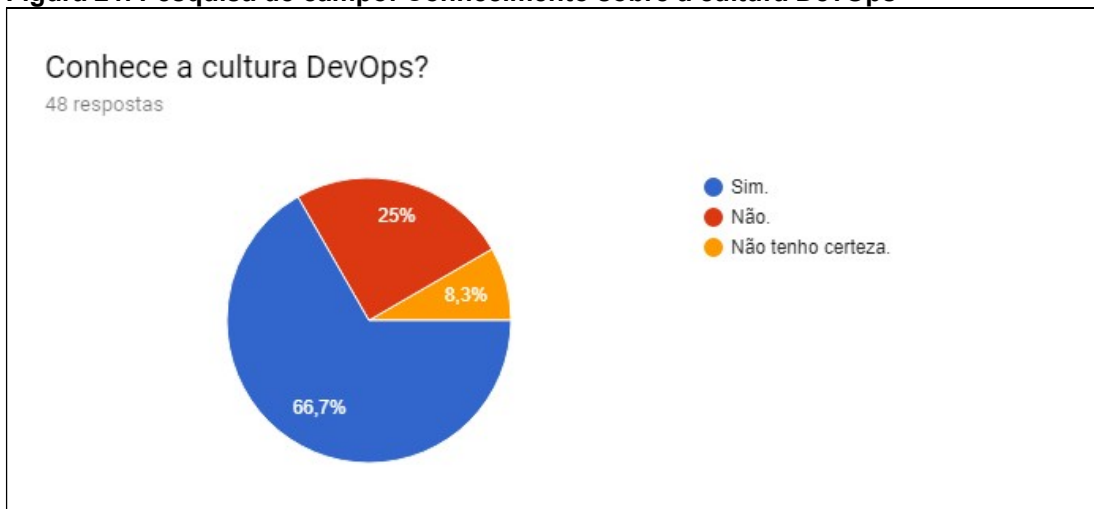
**Figura 20. Pesquisa de campo: Utilização de Sprints**



Fonte: Autoria própria.

Os resultados para a pergunta “Conhece a cultura DevOps?”, apresentados na Figura 21, demonstram que com 66,7% das respostas, a maioria dos participantes consideram que “Sim”, 25% consideram que “Não” e 8,3% consideram que “Não tenho certeza”.

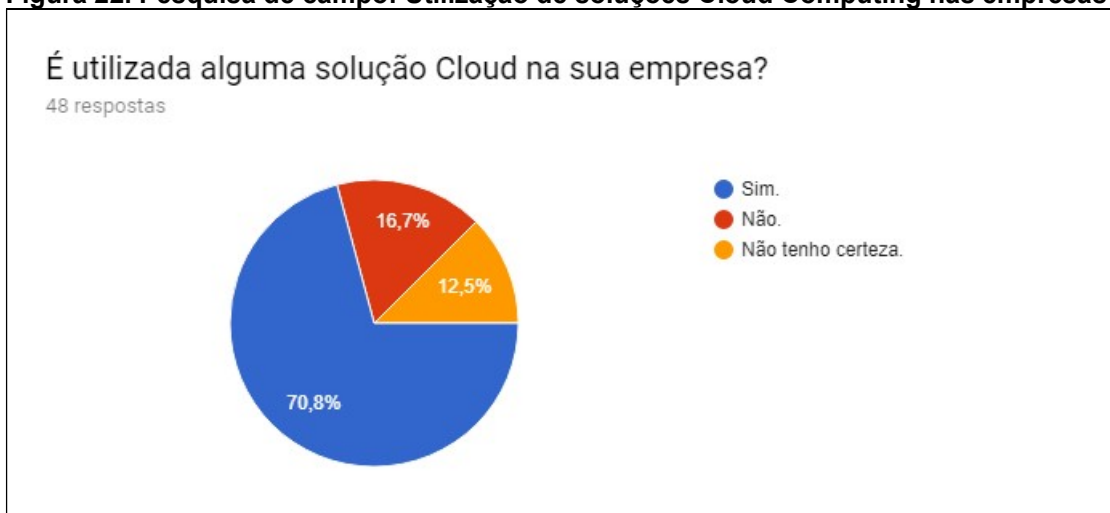
**Figura 21. Pesquisa de campo: Conhecimento sobre a cultura DevOps**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizada alguma solução Cloud na sua empresa?”, apresentados na Figura 22, demonstram que com 70,8% das respostas, a maioria dos participantes consideram que “Sim”, 16,7% consideram que “Não” e 12,5% consideram que “Não tenho certeza”.

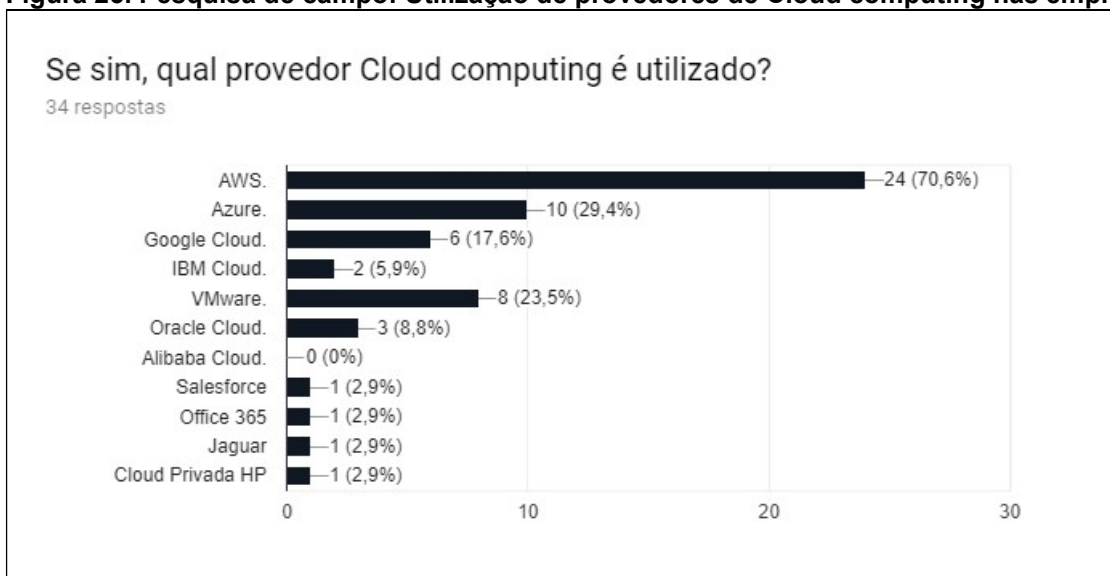
**Figura 22. Pesquisa de campo: Utilização de soluções Cloud Computing nas empresas**



Fonte: Autoria própria.

Os resultados para a pergunta “Se sim, qual provedor Cloud computing é utilizado?”, apresentados na Figura 23, demonstram que com 70,6% das respostas, a maioria dos participantes utilizam “AWS”, 29,4% utilizam “Azure”, 23,5% utilizam “VMware”, 17,6% utilizam “Google Cloud”, 8,8% utilizam “Oracle Cloud”, 5,9% utilizam “IBM Cloud”, 2,9% utilizam “Salesforce”, 2,9% utilizam “Office 365”, 2,9% utilizam “Jaguar” e 2,9% utilizam “Cloud Privada HP”.

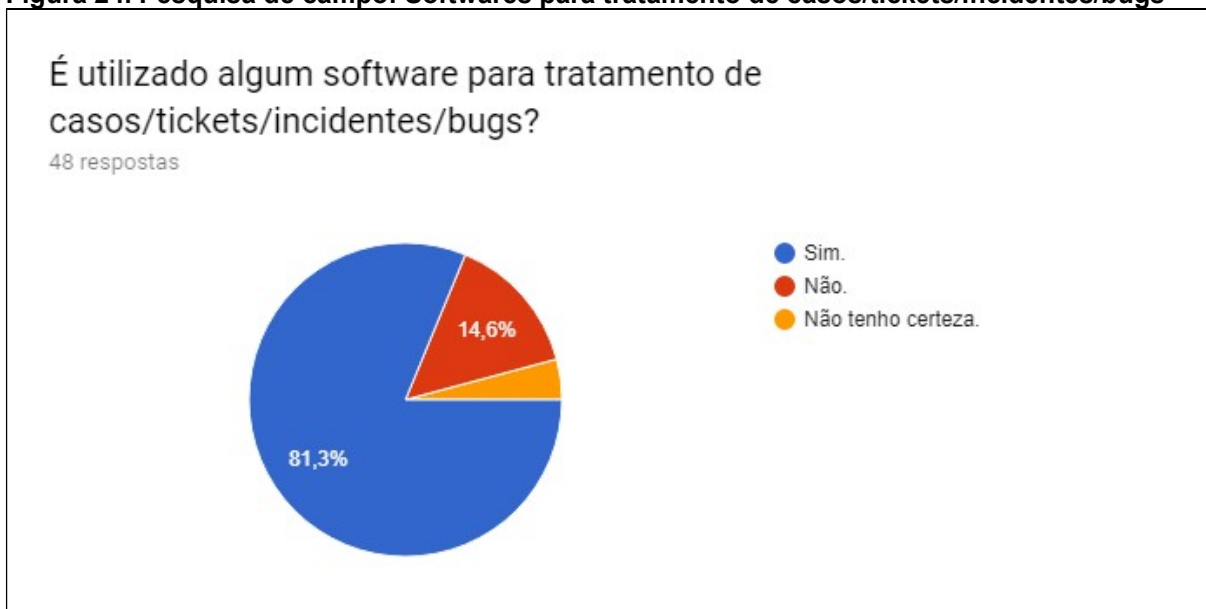
**Figura 23. Pesquisa de campo: Utilização de provedores de Cloud computing nas empresas**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizado algum software para tratamento de casos/tickets/incidentes/bugs?”, apresentados na Figura 24, demonstram que com 81,3% das respostas, a maioria dos participantes consideram que “Sim”, 14,6% consideram que “Não” e 4,2% consideram que “Não tenho certeza”.

**Figura 24. Pesquisa de campo: Softwares para tratamento de casos/tickets/incidentes/bugs**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizado algum centralizador de código?”, apresentados na Figura 25, demonstram que com 75% das respostas, a maioria dos participantes consideram que “Sim”, 12,5% consideram que “Não” e 12,5% consideram que “Não tenho certeza”.

**Figura 25. Pesquisa de campo: Utilização de centralizadores de códigos**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizado algum framework para desenvolvimento?”, apresentados na Figura 26, demonstram que com 75% das respostas, a maioria dos participantes consideram que “Sim”, 20,8% consideram que “Não” e 4,2% consideram que “Não tenho certeza”.

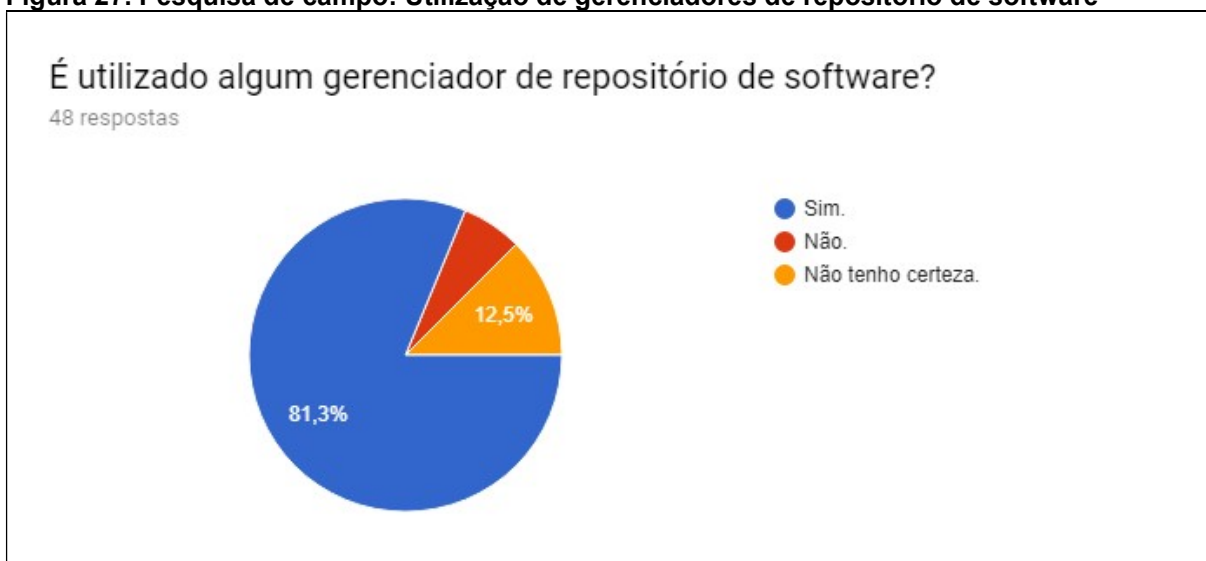
**Figura 26. Pesquisa de campo: Utilização de frameworks para desenvolvimento**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizado gerenciador de repositório de software?”, apresentados na Figura 27, demonstram que com 81,3% das respostas, a maioria dos participantes consideram que “Sim”, 12,5% consideram que “Não” e 6,3% consideram que “Não tenho certeza”.

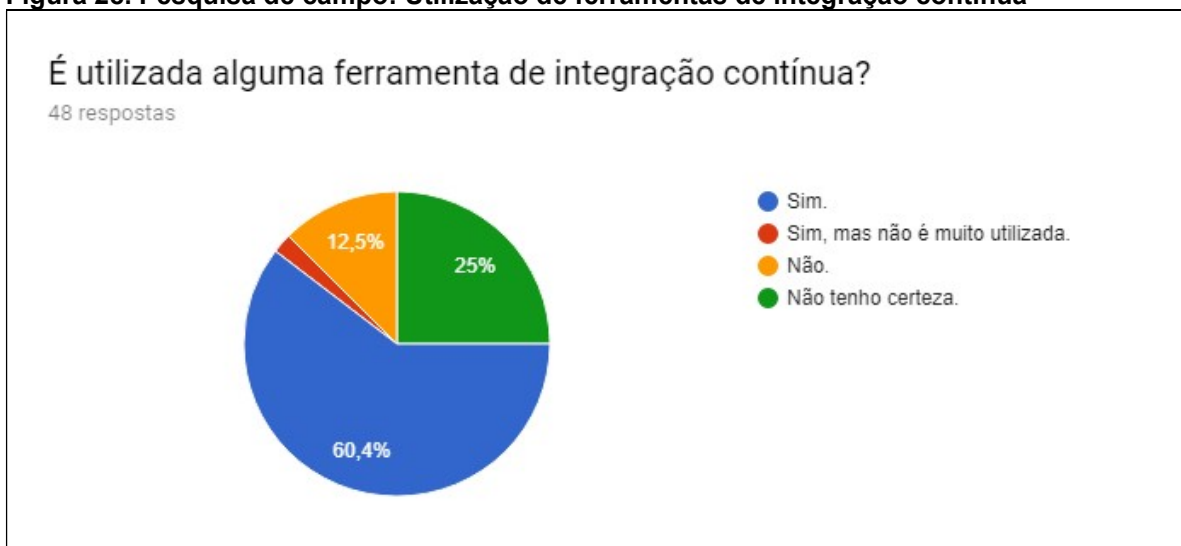
**Figura 27. Pesquisa de campo: Utilização de gerenciadores de repositório de software**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizada alguma ferramenta de integração contínua?”, apresentados na Figura 28, demonstram que com 60,4% das respostas, a maioria dos participantes consideram que “Sim”, 25% consideram que “Não tenho certeza”, 12,5% consideram que “Não” e 2,1% consideram que “Sim, mas não é muito utilizada”.

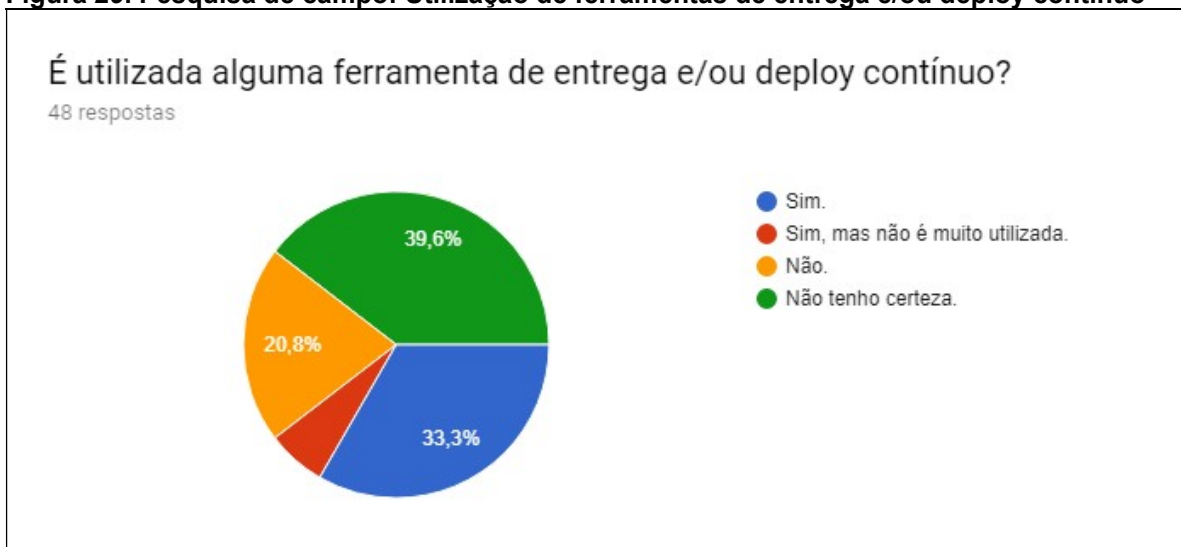
**Figura 28. Pesquisa de campo: Utilização de ferramentas de integração contínua**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizada alguma ferramenta de entrega e/ou deploy contínuo?”, apresentados na Figura 29, demonstram que com 39,6% das respostas, a maioria dos participantes consideram que “Não tenho certeza”, 33,3% consideram que “Sim”, 20,8% consideram que “Não” e 6,3% consideram que “Sim, mas não é muito utilizada”.

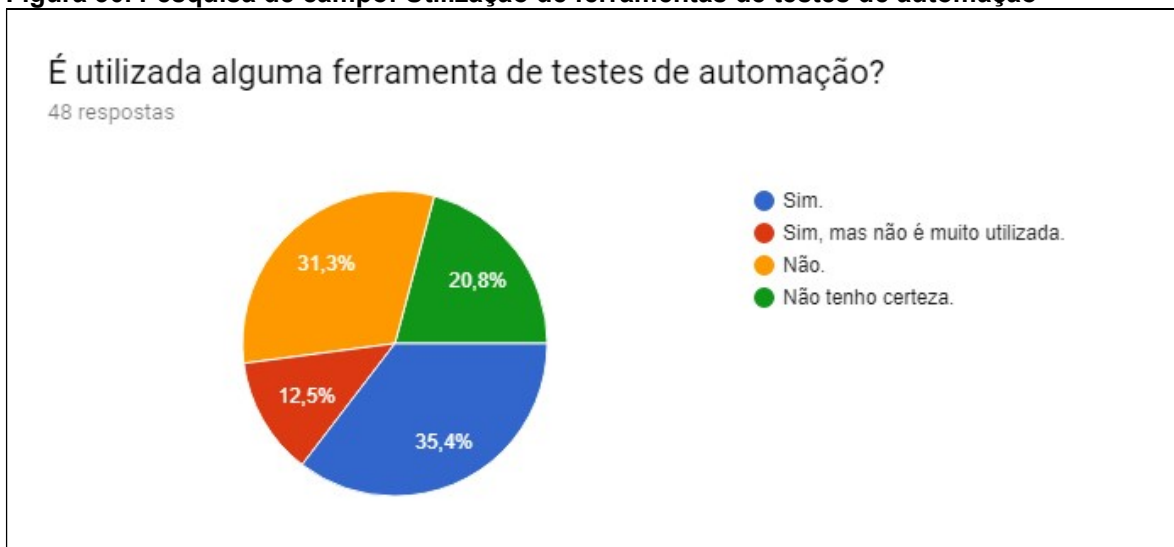
**Figura 29. Pesquisa de campo: Utilização de ferramentas de entrega e/ou deploy contínuo**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizada alguma ferramenta de testes de automação?”, apresentados na Figura 30, demonstram que com 35,4% das respostas, a maioria dos participantes consideram que “Sim”, 31,3% consideram que “Não”, 20,8% consideram que “Não tenho certeza” e 12,5% consideram que “Sim, mas não é muito utilizada”.

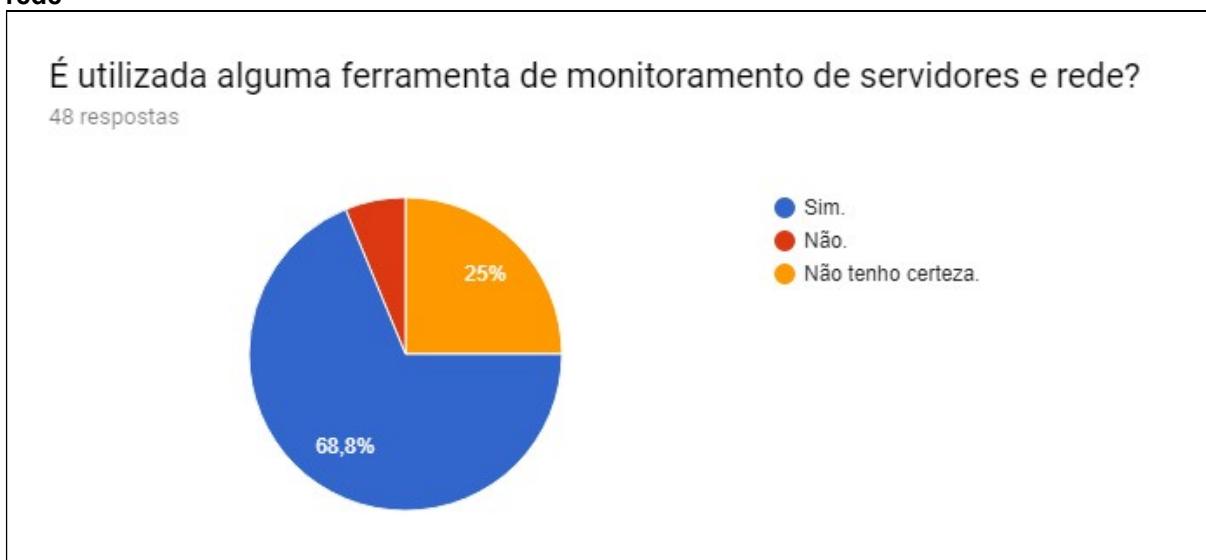
**Figura 30. Pesquisa de campo: Utilização de ferramentas de testes de automação**



Fonte: Autoria própria.

Os resultados para a pergunta “É utilizada alguma ferramenta de monitoramento de servidores e rede?”, apresentados na Figura 31, demonstram que com 68,8% das respostas, a maioria dos participantes consideram que “Sim”, 25% consideram que “Não tenho certeza” e 6,3% consideram que “Não”.

**Figura 31. Pesquisa de campo: Utilização de ferramentas de monitoramento de servidores e rede**



Fonte: Autoria própria.

## 5 CONSIDERAÇÕES FINAIS

O desenvolvimento da cultura DevOps é um desafio para todos, e com este trabalho, foi observado que as empresas de Curitiba estão investindo cada vez mais em metodologias ágeis e a cultura DevOps segue a mesma linha, pois existem inúmeros benefícios aplicados, inclusive para os clientes, que vão receber um retorno muito mais rápido do que utilizando o processo de desenvolvimento tradicional. As equipes de tecnologia da informação precisam se especializar cada vez mais, sendo preferencialmente geridas de forma que seja possível separar as atividades diárias conforme as necessidades, o que é muito importante para o desenvolvimento e a maturidade da equipe na empresa.

Após o levantamento da pesquisa de campo, é importante salientar que as empresas devem analisar antes de seguir detalhadamente um processo ágil, todos devem ter ciência de que é um processo contínuo e deve ser implantado aos poucos nas empresas e todos os integrantes devem fazer parte dessa melhoria. Para que as empresas continuem progredindo e apliquem as metodologias ágeis, profissionais especialistas, como o caso do profissional Scrum Master, são necessários para avaliar as metodologias atuais.

Os profissionais de tecnologia da informação precisam ser motivados a propor novas ferramentas DevOps, e as empresas precisam estar abertas para que esses profissionais tenham um tempo fixo para estudos e testes. Além disso, eles precisam conhecer profundamente as ferramentas utilizadas pela empresa. Durante o processo de integração de um novo colaborador de tecnologia da informação na empresa, por exemplo, é sugerido que seja feita uma apresentação sobre as tecnologias utilizadas, tendo como foco o mesmo conteúdo avaliado no questionário deste trabalho.

## REFERÊNCIAS

AGILE MANIFESTO. (2001) **Principles behind the Agile manifesto**. Disponível em: <<http://agilemanifesto.org/principles.html>>. Acesso em: 03 jan. 2018.

ATLASSIAN. (2018) **#1 Marketplace for DevOps apps: automate, integrate and extend Atlassian tools to build better software faster**. Disponível em: <<https://marketplace.atlassian.com/categories/devops>>. Acesso em: 15 jun. 2018.

BASTOS, Luca. **Cresce a adesão aos métodos ágeis**. ©2017 Digital Network!Brasileiros, publicado em: 28 out. 2013. Disponível em: <<http://cio.com.br/opiniaio/2013/10/28/cresce-a-adesao-aos-metodos-ageis/>>. Acesso em: 12 mar. de 2018.

G2CROWD. (2108) **Best continuous delivery software**. Disponível em: <<https://www.g2crowd.com/categories/continuous-delivery>>. Acesso em: 20 abr. 2018.

GAEA CONSULTING. (2017) **DevOps na prática: dia a dia do desenvolvedor**. Disponível em: <<https://gaea.com.br/devops-na-pratica-dia-a-dia-do-desenvolvedor/>>. Acesso em: 12 mar. 2018.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GILL, Asif Qumer; et al. **DevOps for information management systems**. VINE Journal of Information and Knowledge Management Systems, v. 48, 2017. p.122-139. Disponível em: <<https://www.emeraldinsight.com/doi/abs/10.1108/VJIKMS-02-2017-0007>>. Acesso em: 18 mar. 2018.

GITLAB. (2018) **Complete DevOps with GitLab**. Disponível em: <<https://docs.gitlab.com/ee/README.html#complete-devops-with-gitlab>>. Acesso em: 04 abr. 2018.

GIT-SCM. (2018) **Git**. O site “Git” é atualmente mantido em colaboração com o git-scm.com e pelo repositório no GitHub. Disponível em: <<https://www.git-scm.com/docs/git/1.7.1>>. Acesso em: 18 mar. 2018.



PRESSMAN, Roger S.; MAXIM, Bruce. **Software Engineering: a practitioner's approach**. 8. ed. McGraw-Hill, 2014.

RIGHT SCALE. **Get the RightScale state of the cloud report™**. Disponível em: <<https://www.rightscale.com/lp/state-of-the-cloud>>. Acesso em: 16 jun. 2018.

SALTSTACK. (2018) **SaltStack documentation**. Disponível em: <<https://docs.saltstack.com/en/latest/topics/states/>>. Acesso em: 21 abr. 2018.

SELENIUMHQ. (2018) **Introduction**. Disponível em: <[https://www.seleniumhq.org/docs/01\\_introducing\\_selenium.jsp#test-automation-for-web-applications](https://www.seleniumhq.org/docs/01_introducing_selenium.jsp#test-automation-for-web-applications)>. Acesso em: 20 abr. 2018.

SIMONS, Eric; et al. (2018) **RealWorld example apps**. Disponível em: <<https://github.com/gothinkster/realworld>>. Acesso em: 18 jun. 2018.

SIMPLILEARN SOLUTIONS. **DevOps architect: masters program**. © 2009-2018 Simplilearn Solutions. Disponível em: <<https://www.simplilearn.com/devops-engineer-masters-program-certification-training>>. Acesso em: 15 mar. 2018.

SMITH, Steve. **Practical continuous deployment: a guide to automated software delivery**. ATlassian Blog, publicado em: 17 fev. 2017. Disponível em: <<https://www.atlassian.com/blog/continuous-delivery/practical-continuous-deployment>>. Acesso em: 10 jan. 2018.

SUTHERLAND, Jeff; SCHWABER, Ken. **The definitive guide to Scrum: the rules of the game**. ©2018 ScrumGuides.org, publicado em: jul. 2016. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf>>. Acesso em: 13 mar. 2018.

ZANINOTTO, François; POTENCIER, Fabien. **The definitive guide to symfony**. 1. ed. Apress, 2007.