

***DevOps* - Análise do processo de implantação da cultura *DevOps* em uma empresa brasileira**

Thiago Gonzaga Gomes



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB
Outubro - 2023

Thiago Gonzaga Gomes

DevOps - Análise do processo de implantação da cultura *DevOps* em uma empresa brasileira

Monografia apresentada ao curso Engenharia de Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Engenheiro de Computação.

Orientador: Danielle Rousy Dias Ricarte

João Pessoa, PB
Outubro - 2023

Catálogo na publicação
Seção de Catalogação e Classificação

G633d Gomes, Thiago Gonzaga.

Devops - análise do processo de implantação da cultura devops em uma empresa brasileira / Thiago Gonzaga Gomes. - João Pessoa, 2023.

83 f. : il.

Orientação: Danielle Ricarte.

TCC (Graduação) - UFPB/CI.

1. Devops. 2. Implementação. 3. Fluxo de desenvolvimento de software. 4. Gestão de processo. I. Ricarte, Danielle. II. Título.

UFPB/CI

CDU 004.4



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia de Computação intitulado **Análise do processo de implantação da cultura *DevOps* em uma empresa brasileira** de autoria de **Thiago Gonzaga Gomes**, aprovada pela banca examinadora constituída pelos seguintes professores:

Profa. Dra. Danielle Rousy Dias Ricarte
UFPB

Profa. Dra. Thais Gaudêncio do Rego
UFPB

Prof. Dr. Carlos Eduardo Coelho Freire Batista
UFPB

João Pessoa, 27 de outubro de 2023

RESUMO

Neste estudo, a motivação inicial foi impulsionada pela necessidade de aprimorar os processos de desenvolvimento de software em uma empresa brasileira. A organização enfrentava desafios decorrentes de um processo em cascata com várias etapas. Para superar essas limitações, a transição para o DevOps foi adotada como um objetivo fundamental, visando alcançar maior eficiência, integração e automação nos processos de desenvolvimento e entrega de software. Para atingir esse objetivo, foram estabelecidos princípios orientadores que facilitaram a adoção das práticas DevOps. Além disso, uma revisão de literatura foi realizada para comparar metodologias tradicionais e ágeis, identificando os benefícios e obstáculos associados ao DevOps. Ao longo do estudo de caso, ficou evidente a importância da integração contínua e da entrega contínua, bem como o papel crucial do apoio da alta gestão para garantir o sucesso da adoção. Embora tenham sido registradas melhorias na qualidade do software, também foi destacada a resistência à mudança como um desafio significativo. A adoção das práticas DevOps em um ambiente organizacional alinhou-se eficazmente com as metas de lançamentos mais frequentes, automação e desenvolvimento de software confiável, satisfazendo as necessidades e práticas específicas de cada setor. Isso foi motivado pela busca contínua de aprimoramento e eficiência no contexto da empresa.

Palavras-chave: *DevOps*, Implementação, Fluxo de Desenvolvimento de *software*, Gestão de processo.

ABSTRACT

In this study, the initial motivation was driven by the need to enhance software development processes in a Brazilian company. The organization faced challenges stemming from a multi-stage, waterfall process. To overcome these limitations, the transition to DevOps was adopted as a core objective, aiming to achieve greater efficiency, integration, and automation in software development and delivery processes. To achieve this goal, guiding principles were established to facilitate the adoption of DevOps practices. Additionally, a literature review was conducted to compare traditional and agile methodologies, identifying the benefits and obstacles associated with DevOps. Throughout the case study, the importance of continuous integration and continuous delivery became evident, as well as the crucial role of senior management support in ensuring the success of adoption. While improvements in software quality were documented, resistance to change was also highlighted as a significant challenge. The adoption of DevOps practices in the organizational setting effectively aligned with the goals of more frequent releases, automation, and reliable software development, meeting the specific needs and practices of each sector. This was driven by the ongoing pursuit of improvement and efficiency within the company's context.

Key-words: *DevOps*, Implementation, *Software Development Flow*, Process management.

LISTA DE FIGURAS

Figura 1. Etapas da metodologia de investigação “Estudo de Caso”.....	19
Figura 2. Fluxo <i>DevOps</i>	27
Figura 3.1 Melhor abordagem (a nível organizacional) para a adoção do <i>DevOps</i> ?.....	38
Figura 3.2 Fases com maior risco na implementação do <i>DevOps</i>	39
Figura 3.3 Práticas do <i>DevOps</i> mais impactantes (positivamente) para a implementação do <i>DevOps</i>	41
Figura 3.4 Processos da organização semelhantes a um processo inerente à cultura <i>DevOps</i>	43
Figura 3.5 Alternativas à adoção do <i>DevOps</i>	44
Figura 3.6 Utilização de guia de boas práticas de implementação do <i>DevOps</i>	45
Figura 3.7 Necessidade de participação de entidades externas no processo de implementação.....	46
Figura 3.8 Áreas com maior mudança organizacional/cultural.....	48
Figura 3.9 Tipologia do processo implementado.....	49
Figura 3.10 Implementação do <i>DevOps</i>	51
Figura 3.11 Categorias das ferramentas implementadas.....	53
Figura 3.12 Principais benefícios da adoção do <i>DevOps</i>	56
Figura 3.13 As equipas passaram a trabalhar de forma mais colaborativa, após a implementação?.....	57
Figura 3.14 Frequência das correções de emergência em produção.....	58
Figura 3.15 Satisfação do cliente.....	58
Figura 3.16 Satisfação da equipa de desenvolvimento.....	59
Figura 3.17 Satisfação da equipa de operações.....	59
Figura 3.18 Satisfação da organização.....	60
Figura 3.19 Frequência das entregas.....	61
Figura 3.20 Automatização.....	61
Figura 3.21 Importância da mudança para a cultura <i>DevOps</i> na perspetiva de stakeholder.....	62
Figura 3.22 Principais obstáculos à adoção do <i>DevOps</i>	65

Figura 3.23 Resistência à mudança.....	67
Figura 3.24 Melhoria da comunicação entre equipes com a implementação do <i>DevOps</i>.....	68
Figura 3.25 Dimensão da organização como barreira.....	68
Figura 3.26 Cumprimento dos objetivos.....	70
Figura 3.27 Satisfação como stakeholder.....	71
Figura 3.28 Critérios de sucesso de implementação do <i>DevOps</i>.....	72
Figura 3.29 Aplicação do <i>DevOps</i>.....	73
Figura 3.30 Sucesso da adoção do <i>DevOps</i> na organização.....	74

LISTA DE TABELAS

<u>Tabela 1. Pilares do <i>DevOps</i>.....</u>	<u>24</u>
<u>Tabela 2. Comparação modelo tradicional vs <i>DevOps</i>.....</u>	<u>28</u>
<u>Tabela 3. Benefícios da adoção do <i>DevOps</i>.....</u>	<u>29</u>
<u>Tabela 4. Barreiras à adoção do <i>DevOps</i>.....</u>	<u>31</u>
<u>Tabela 5. Identificação dos entrevistados.....</u>	<u>35</u>

LISTA DE ABREVIATURAS

CI	–	Continuous Integration
CD	–	Continuous Deployment
Dev	–	Developer/Development
DevOps	–	Development + Operations
LGPD	–	Lei Geral de Proteção de Dados
Ops	–	Operations
SaaS	–	Software as a Service
SDLC	–	Systems Development Life Cycle
SRE	–	Site Reliability Engineering
TI	–	Tecnologia da Informação

SUMÁRIO

1. INTRODUÇÃO	14
1.1 Problema	15
1.1.1 Objetivo geral	15
1.1.2 Objetivos específicos	15
1.2 Estrutura da monografia	16
2. METODOLOGIA	17
2.1 Estudo de caso	17
2.2.2 Seleção dos casos	18
2.2.3 Elaboração do protocolo	18
2.2.4 Questionário	18
2.2.4 Realização do estudo de caso	18
3. FUNDAMENTAÇÃO TEÓRICA	20
3.1 Princípios do DevOps	21
3.2 Pilares do DevOps	21
3.3 Práticas DevOps	23
3.3.1 Integração Contínua (Continuous Integration/CI)	23
3.3.2 Entrega Contínua (Continuous Delivery)	23
3.3.3 Implementação Contínua (Continuous Deployment/CD)	24
3.3.4 Monitoramento Contínuo	24
3.4 Fluxo DevOps	25
3.5 Abordagem tradicional vs. DevOps	26
3.6 Possíveis benefícios do DevOps	27
3.7 Desafios e barreiras à adoção	29
3.8 DevOps e Automação são nossa próxima bala de prata?	30
4. EXECUÇÃO DA METODOLOGIA	32
4.1 Seleção dos casos e contextualização	32
4.2 Elaboração do Protocolo	32
4.2.1 Entrevista	34
5. ANÁLISE E DISCUSSÃO DOS RESULTADOS	36
5.1 Definição do DevOps	36
5.1.1 O que é DevOps?	36
5.1.2 Melhor abordagem (a nível organizacional) para a adoção do DevOps	36
5.1.3 Fases com maior risco na implementação do DevOps	38
5.1.4 Práticas do DevOps mais impactantes (positivamente) para a implementação do DevOps	39
5.2 Pré-DevOps	40
5.2.1 Motivações para a implementação do DevOps	40

5.2.2 Ferramentas utilizadas para gerir o SDLC anteriores à implementação do DevOps	40
5.2.3 Processos semelhantes a DevOps anteriores à sua implementação	41
5.3 Implementação do DevOps	42
5.3.1 Alternativas à adoção do DevOps	42
5.3.2 Objetivos da implementação do DevOps	43
5.3.3 Melhores práticas na implementação do DevOps	43
5.3.4 Participação de entidades externas	44
5.3.5 Custo estimado da implementação do DevOps	44
5.3.6 Área(s) com maior mudança organizacional/cultural	45
5.3.7 Tipologia do processo implementado	46
5.4 Práticas DevOps	47
5.4.1 Práticas implementadas	47
5.4.2 Principais fatores influenciadores do sucesso da implementação do DevOps	47
5.5 Técnicas e Ferramentas do DevOps	48
5.5.1 Implementação do DevOps	48
5.5.2 Ferramentas implementadas	49
5.5.3 Utilização do Provedor de Nuvem (Cloud)	51
5.5.4 Sistema de controle de versões	51
5.6 Benefícios DevOps	52
5.6.1 Principais benefícios da adoção do DevOps	52
5.6.2 Colaboração entre equipes	53
5.6.3 Correções de emergência em produção	54
5.6.4 Satisfação do cliente e equipes	55
5.6.5 Satisfação da organização	57
5.6.6 Frequência de entregas	57
5.6.7 Automatização	58
5.7 Gestão da mudança	59
5.7.1 Mudança para a cultura DevOps como stakeholder	59
5.7.2 Formação para a equipe responsável pela implementação do DevOps	59
5.7.3 Formação das equipes envolvidas não só no processo de implementação	60
5.7.4 Alterações na infraestrutura da organização	60
5.7.5 Alterações estruturais ou reorganização de departamentos da organização	60
5.7.6 Gestão da mudança	61
5.8 Barreiras DevOps	61
5.8.1 Principais obstáculos à adoção do DevOps	61
5.8.2 Resistência à mudança	63
5.8.3 Melhoria da comunicação entre as equipes com a implementação do DevOps	64
5.8.4 Dimensão da organização como barreira	64
5.9 Pós-Implementação do DevOps	65

5.9.1 Cumprimento dos objetivos	66
5.9.2 Satisfação como stakeholder	66
5.9.3 Critérios para avaliação do sucesso da implementação do DevOps	67
5.9.4 Avaliação formal do sucesso da implementação do DevOps	69
5.9.5 Aplicação do DevOps	69
5.9.6 Sucesso da adoção do DevOps na organização	70
5.10 Considerações finais	70
5.10.1 Adoção do DevOps	71
5.10.2 Benefícios do DevOps	72
5.10.3 Barreiras à adoção do DevOps	72
6. CONCLUSÃO	74
REFERÊNCIAS	75
APÊNDICE	78

1. INTRODUÇÃO

Nos últimos anos, o lançamento de *software* tornou-se mais acessível devido à adoção do modelo *software* como Serviço (*SaaS*). As empresas que conseguem lançar *software* com antecedência e frequência têm uma vantagem competitiva. Os profissionais de TI precisam se manter atualizados tecnicamente para garantir a qualidade técnica e obter melhores resultados para suas empresas, além de aumentar suas chances de empregabilidade, visto que quanto mais áreas de conhecimento o profissional tiver, mais preparado estará.

O mercado de TI está adotando o *DevOps* e práticas contínuas de engenharia de *software* para competir no mercado. A prática *DevOps* integra o **desenvolvimento** (Dev) e as **operações** (Ops) de *software*, com automação de desenvolvimento, implantação e monitoramento. A escassez de desenvolvedores profissionais torna essencial melhorar a eficiência, eliminando tarefas manuais repetitivas. No entanto, a transição para essas práticas modernas não é simples para grandes empresas com várias funções e departamentos especializados. A adoção da entrega contínua depende da estrutura organizacional.

A abordagem do *DevOps* permite que as empresas alcancem seus objetivos, criando colaboração multifuncional. O objetivo é reduzir o tempo entre o desenvolvimento e a operação do *software*, sem comprometer a qualidade.

O *DevOps* é uma prática que visa promover a colaboração entre equipes, aumentando o fluxo de trabalho, a frequência de entregas e a estabilidade do ambiente de produção. Não se trata de uma ferramenta, mas sim de um conjunto de práticas que modificam a cultura organizacional para a implantação de processos ágeis. O objetivo é garantir a qualidade dos produtos, cumprir prazos e integrar todas as equipes para tornar o desenvolvimento de *software* mais eficiente.

No contexto de *software*, o desenvolvimento é responsável pela codificação, testes, implementação de funcionalidades e integração de sistemas. As operações são encarregadas da implantação de *software*, monitoramento de sistemas, segurança, gerenciamento de incidentes e outros aspectos relacionados à infraestrutura e operação do *software*.

Os benefícios incluem a entrega rápida e contínua de valor, melhor comunicação e resolução acelerada de problemas. O objetivo é ter ciclos de aprendizado curtos para eventuais falhas serem identificadas rapidamente e criar valor de forma ágil.

O estudo utiliza a metodologia de estudo de caso descritivo para analisar e caracterizar o fenômeno do *DevOps* dentro do contexto de uma empresa real. O estudo de caso é uma abordagem qualitativa amplamente utilizada em sistemas de informação, permitindo a compreensão das interações entre inovações tecnológicas e os contextos organizacionais.

1.1 Problema

O presente estudo se propõe a responder o seguinte questionamento: “Como a cultura *DevOps* pode beneficiar a qualidade do processo de desenvolvimento de *software* em um ambiente organizacional?”.

1.1.1 Objetivo geral

O objetivo geral deste estudo é, além de explicar o conceito de *DevOps*, fazer uma análise de sentimento dos períodos antes/durante/depois da implementação do *DevOps*, e analisar os impactos da implantação desse processo em uma organização que não possuía uma cultura *DevOps* previamente implementada.

1.1.2 Objetivos específicos

Com o intuito de alcançar o objetivo geral deste estudo, destacam-se os seguintes objetivos específicos:

- Identificar as práticas de *DevOps* no desenvolvimento de *software*, especificamente em um contexto organizacional.
- Analisar como as práticas de *DevOps* influenciam a qualidade no desenvolvimento de *software*.
- Avaliar a efetividade da implementação das práticas de *DevOps* em um contexto organizacional.

1.2 Estrutura da monografia

Este estudo está dividido em 5 capítulos, os quais contêm as seguintes abordagens:

No primeiro capítulo, é realizada a introdução do conteúdo, abordando o problema de pesquisa e seus objetivos gerais e específicos.

No segundo capítulo, é apresentada a metodologia de pesquisa utilizada para a realização deste estudo.

No terceiro capítulo, é apresentado o embasamento teórico obtido, incluindo a definição de *DevOps* e seu contexto.

No quarto capítulo, descreve-se a execução da metodologia e suas motivações.

No quinto capítulo, são apresentadas a análise e a discussão dos resultados obtidos.

No sexto e último capítulo, é apresentada a conclusão do estudo.

2. METODOLOGIA

2.1 Estudo de caso

De acordo com Yin (1989), um estudo de caso é uma investigação empírica que analisa fenômenos contemporâneos relacionados a um evento, entidade, indivíduo ou unidade de análise. Por meio da coleta de evidências do contexto da realidade, é possível obter esclarecimentos sobre o estudo em questão.

A Figura 1 mostra um diagrama que ilustra as etapas envolvidas na condução de uma investigação utilizando a metodologia de "estudo de caso". Onde a formulação do problema foi a investigação da implementação do processo, seguido pelas etapas de seleção de caso e elaboração de protocolo, que acontecem em paralelo, e após esses passos, os dados são colhidos e analisados a fim de gerar um relatório conclusivo.

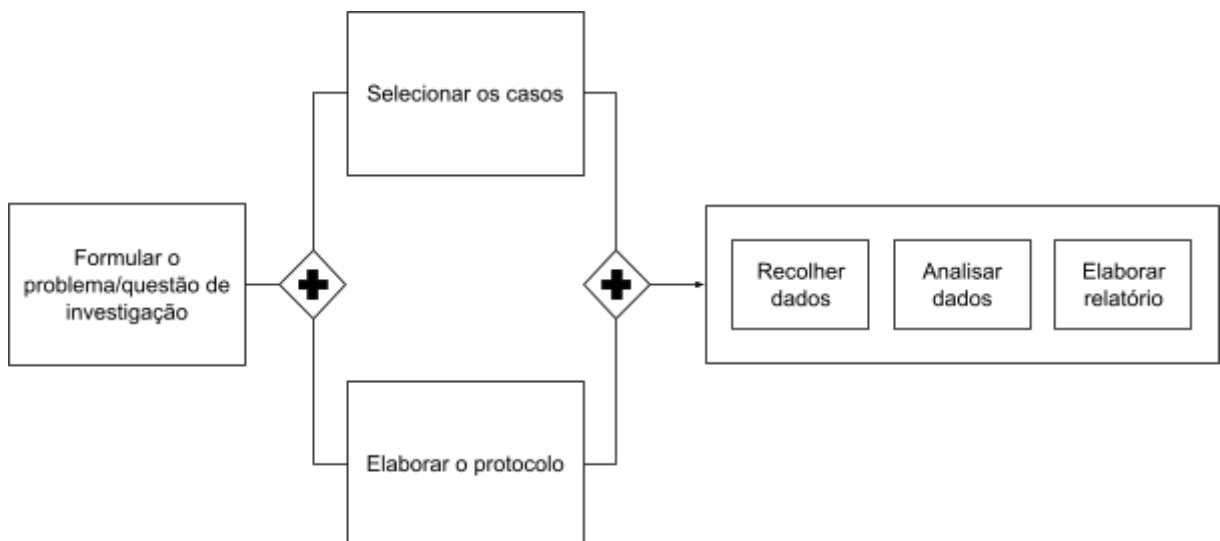


Figura 1. Etapas da metodologia de investigação “Estudo de Caso”. Fonte: Adaptado de Yin (2009).

O objetivo da pesquisa é estudar o fenômeno do *DevOps*, explorando aspectos relacionados à sua implementação, ferramentas, técnicas, benefícios e desafios para sua adoção e sucesso, tudo isso no contexto de uma empresa brasileira.

2.2.2 Seleção dos casos

Na etapa de definição da unidade de estudo, é escolhido o caso a ser investigado. Pode-se optar por estudar um único caso ou vários casos. Neste estudo, foi selecionado um único caso.

O caso selecionado é a implementação da cultura *DevOps* numa organização brasileira, que será melhor descrita na sessão 4.1, contendo a descrição da metodologia utilizada no estudo.

2.2.3 Elaboração do protocolo

O protocolo de pesquisa contém os elementos necessários para a coleta de dados e define a conduta a ser adotada para sua aplicação. Ele inclui uma visão geral do estudo de caso, os procedimentos de campo, as questões de pesquisa e como os resultados serão relatados.

2.2.4 Questionário

Foi desenvolvido um questionário (mais detalhes no Capítulo 4) que foi utilizado para conduzir uma entrevista a respeito de todo o ciclo de implementação e de suas opiniões a respeito do mesmo. O questionário incluiu questões de resposta fechada, de escolha múltipla ou usando uma escala de satisfação de 0-10.

2.2.4 Realização do estudo de caso

A etapa de execução do estudo de caso, conforme ilustrado acima na Figura 1, envolve três atividades distintas, conforme descrito a seguir.

A coleta de dados ocorreu na organização selecionada para o estudo de caso, abrangendo o período de janeiro a maio de 2023. Durante esse período, foram realizadas observações diretas, permitindo identificar os processos e ferramentas utilizados, além de

auxiliar na descrição geral do projeto. Por meio do acesso à documentação, foi possível identificar os processos e sua relação.

Além das observações diretas, foram conduzidas 14 entrevistas com diversos *stakeholders* (principais interessados) do projeto. Isso incluiu membros da equipe de *SRE/Cloud* (*Site Reliability Engineering* e *Cloud Engineering*, são cargos da equipe de Plataforma, responsáveis pela manutenção e disponibilidade da infraestrutura que mantém as aplicações) e da equipe de gestão de projetos. No capítulo 4 será melhor destrinchado a execução da metodologia descrita.

3. FUNDAMENTAÇÃO TEÓRICA

Essa seção será uma referência aos conhecimentos comum e literário no que se diz respeito ao conceito de *DevOps*, seus pilares, práticas, fluxo, comparação entre um cenário com e sem *DevOps*, assim como possíveis benefícios, desafios e barreiras da sua implementação.

O termo *DevOps* surge da contração de dois termos: Desenvolvimento (*Development*) e Operações (*Operations*). Ambos se referem a escopos de atuação no ciclo de desenvolvimento e manutenção de um sistema/produto.

- ***Development*** (Desenvolvimento): escopo responsável pelo planejamento, projeto, implementação e testes. É o desenvolvedor de fato.
- ***Operations*** (Operações): escopo responsável pela implantação, manutenção e monitoramento do sistema/produto no ambiente de produção, assim como a solução de incidentes e eventuais problemas. Geralmente o escopo de operações também é responsável ou co-responsável pela infraestrutura do sistema/produto.

A cultura *DevOps* enfoca na colaboração entre as equipes de desenvolvimento e operações para melhorar a entrega de valor aos usuários finais de forma mais rápida e eficiente. Surgiu no início dos anos 2000, quando empresas perceberam os benefícios de incorporar princípios ágeis e enxutos em seus processos de desenvolvimento de *software*. O movimento *DevOps* defende a adoção desses princípios para melhorar a velocidade, qualidade e agilidade do desenvolvimento de *software*.

O termo *DevOps* surgiu em 2009 durante uma palestra na conferência *Velocity* da *O'Reilly*, intitulada "*10+ Deploys per Day: Dev and Ops Cooperation at Flickr*". Os palestrantes identificaram problemas comuns entre as equipes de desenvolvimento e operações, como falta de eficácia e dificuldades na comunicação. A solução proposta foi integrar o desenvolvimento e as operações em uma unidade coesa.

3.1 Princípios do *DevOps*

Como o *DevOps* não tem uma definição *patenteada*, e não existe um guia universal de implementação, foram definidos princípios básicos a serem seguidos para que a curva de implementação do *DevOps* seja o mais linear possível (Kim et al., 2016), sendo eles:

- Testar sempre que possível, visando garantir a integridade de todo o ciclo de desenvolvimento e da própria aplicação.
- Promover melhoria contínua dos processos implementados, com a utilização de novas técnicas e ferramentas que busquem aprimorar a qualidade do produto, tornando as entregas ainda mais rápidas.
- Automatizar tudo o que for possível, com o objetivo de reduzir a repetição desnecessária de atividades cotidianas e evitar erros humanos.
- Trabalhar em sinergia em uma única equipe para buscar os mesmos objetivos, eliminando assim as lacunas existentes.
- Separar o processo de desenvolvimento por áreas e atividades, de forma não disruptiva, para que se saiba onde começa e termina cada uma das fases do ciclo.

3.2 Pilares do *DevOps*

Assim como com os princípios, alguns pilares foram definidos para facilitar a sua implementação e entendimento, esses pilares representam os principais princípios e práticas que formam a base da abordagem *DevOps* para o desenvolvimento de *software*.
BUCHANAN, Ian - Atlassian (2023).

Os pilares do *DevOps* são representados pela sigla *CALMS*, que significa Cultura (*Culture*), Automação (*Automation*), Metodologia *Lean* (*LeanIT*), Medição (*Measurement*), Colaboração (*Sharing*) e Integração (*Integration*).

Na tabela 1, os pilares são destrinchados e descritos de acordo com seu papel no ecossistema do processo *DevOps*.

Pilar	Descrição
Cultura	O <i>DevOps</i> enfatiza uma cultura de colaboração, comunicação e melhoria contínua entre as equipes que lidam com diferentes aspectos do ciclo de vida

	do desenvolvimento de <i>software</i> . Essa cultura é essencial para permitir que as equipes trabalhem juntas de maneira eficaz e impulsionem a inovação e a eficiência no processo de desenvolvimento.
Automação	O <i>DevOps</i> utiliza a automação para simplificar o processo de desenvolvimento de <i>software</i> e para permitir que as equipes criem, testem e implantem aplicativos com mais rapidez e confiabilidade. As ferramentas de automação podem ser usadas para automatizar tarefas repetitivas, como criação e implantação de código, bem como para monitorar o desempenho de aplicativos em produção.
Metologia Lean	O <i>Lean</i> é uma filosofia de gestão originária do sistema <i>Toyota</i> de produção, que visa eliminar desperdícios e otimizar processos. Ao implementar práticas <i>Lean</i> no ambiente de desenvolvimento de software, é possível reduzir o tempo de entrega, aumentar a qualidade do produto final e melhorar a satisfação do cliente.
Medição	O <i>DevOps</i> depende de dados e métricas para orientar a tomada de decisões e melhorar a eficiência e a eficácia do processo de desenvolvimento. As equipes usam uma variedade de ferramentas e técnicas para coletar e analisar dados, como métricas de desempenho, dados de uso e <i>feedback</i> dos usuários, para identificar áreas de melhoria e tomar decisões informadas.
Colaboração	O <i>DevOps</i> enfatiza o compartilhamento de conhecimento, ferramentas e práticas recomendadas entre as equipes para promover a colaboração e o aprendizado contínuo. As equipes utilizam uma variedade de técnicas, como revisão de código, programação em pares e documentação, para compartilhar conhecimento e experiência e melhorar a qualidade e a capacidade de manutenção da base de código.

Tabela 1. Pilares do *DevOps*.

3.3 Práticas *DevOps*

De forma similar aos princípios e pilares, o mercado precisou adotar práticas que buscam otimizar os ambientes de desenvolvimento de *software*, decisão impulsionada pela crescente competitividade nos cenários corporativos

Entre essas práticas, destacam-se a integração contínua (*Continuous Integration/CI*), a entrega contínua (*Continuous Delivery*), a implementação contínua (*Continuous Deployment/CD*) e o monitoramento contínuo. HALL, Tom - Atlassian (2023).

3.3.1 Integração Contínua (*Continuous Integration/CI*)

A Integração Contínua (*CI*) é uma prática comum no desenvolvimento de *software* em que os desenvolvedores mesclam regularmente suas alterações de código em um repositório central.

Esse processo automatizado inclui a construção e teste do código para garantir sua validade e conformidade com os padrões de qualidade do projeto. A *CI* visa identificar e corrigir problemas no início do processo de desenvolvimento, economizando tempo e esforço no lançamento de atualizações de *software*. A prática visa integrar o desenvolvimento em andamento, fornecendo *feedback* constante para toda a equipe de desenvolvedores.

A *CI* está conectada a um repositório de versionamento de código, monitorando constantemente as alterações mescladas pelos desenvolvedores em um ramo específico do repositório. Quando são detectadas novas alterações, o código é compilado e, em alguns casos, passa por testes unitários para confirmar sua aceitação nas etapas subsequentes.

Após a conclusão do processo, a *CI* gera um "artefato", que é um arquivo compactado contendo todo o código gerado e compilado. Esse artefato é o produto final pronto para ser enviado para as etapas seguintes do desenvolvimento. Atlassian (2023).

3.3.2 Entrega Contínua (*Continuous Delivery*)

A Entrega Contínua é uma prática que garante que o código compilado pelo processo de Integração Contínua esteja pronto para ser disponibilizado nos ambientes de

desenvolvimento e produção. Isso permite que os desenvolvedores entreguem novos recursos e alterações com eficácia, facilidade e rapidez aos usuários finais, mantendo o produto sempre em um estado pronto para ser lançado.

O código compilado é submetido a testes de qualidade para reduzir o risco de falhas durante o lançamento (*deploy*). Uma vez aprovadas, as alterações podem ser implantadas nos ambientes desejados. No ambiente de produção, há um responsável que decide se as alterações serão lançadas, realizando uma última revisão e análise das alterações propostas. Esse é um dos poucos passos manuais em todo o processo de entrega contínua. Atlassian (2023).

3.3.3 Implementação Contínua (*Continuous Deployment/CD*)

A Implementação Contínua é responsável por eliminar o passo manual no processo. Após um ciclo bem-sucedido de Integração Contínua seguido de Entrega Contínua, o artefato gerado pelo *CI* é importado pelo *CD*, que aguarda novos artefatos. Quando o *CD* identifica um novo artefato, significa que há alterações prontas para serem lançadas nos ambientes correspondentes, permitindo o uso independente das alterações de diferentes equipes. Isso aumenta a produtividade e reduz os ciclos de lançamento.

Essas três práticas - *CI*, Entrega Contínua e Implementação Contínua - são essenciais para o fluxo *DevOps*. Em resumo, o *CI/CD* permite a produção mais rápida e frequente de novas versões do código, criando uma conexão entre o desenvolvimento e a produção. Os clientes e usuários finais se beneficiam dessas práticas, pois têm acesso contínuo e rápido a novas atualizações, com correções e recursos adicionais. Atlassian (2023).

3.3.4 Monitoramento Contínuo

No contexto *DevOps*, o monitoramento contínuo refere-se à prática automatizada de acompanhar o desempenho, disponibilidade e segurança de um sistema de *software* em produção. Isso permite que as equipes identifiquem e resolvam rapidamente problemas, garantindo que o sistema funcione da melhor maneira possível e agregue valor aos usuários finais.

O monitoramento contínuo é uma parte importante do desenvolvimento de *software* no *DevOps*, pois permite que as equipes avaliem e melhorem constantemente o desempenho do sistema. Isso garante que o *software* seja confiável e esteja sempre disponível, além de permitir uma resposta ágil a problemas.

Além disso, o monitoramento contínuo possibilita que as equipes entreguem rapidamente novos recursos e atualizações para o sistema, fornecendo dados e informações em tempo real para tomar decisões informadas sobre o projeto. Atlassian (2023).

3.4 Fluxo *DevOps*

De forma sucinta, pode-se resumir os princípios, pilares e práticas *DevOps* com a Figura 2, onde é possível observar o real fluxo *DevOps*, com todas as suas etapas, que serão destrinchadas abaixo.

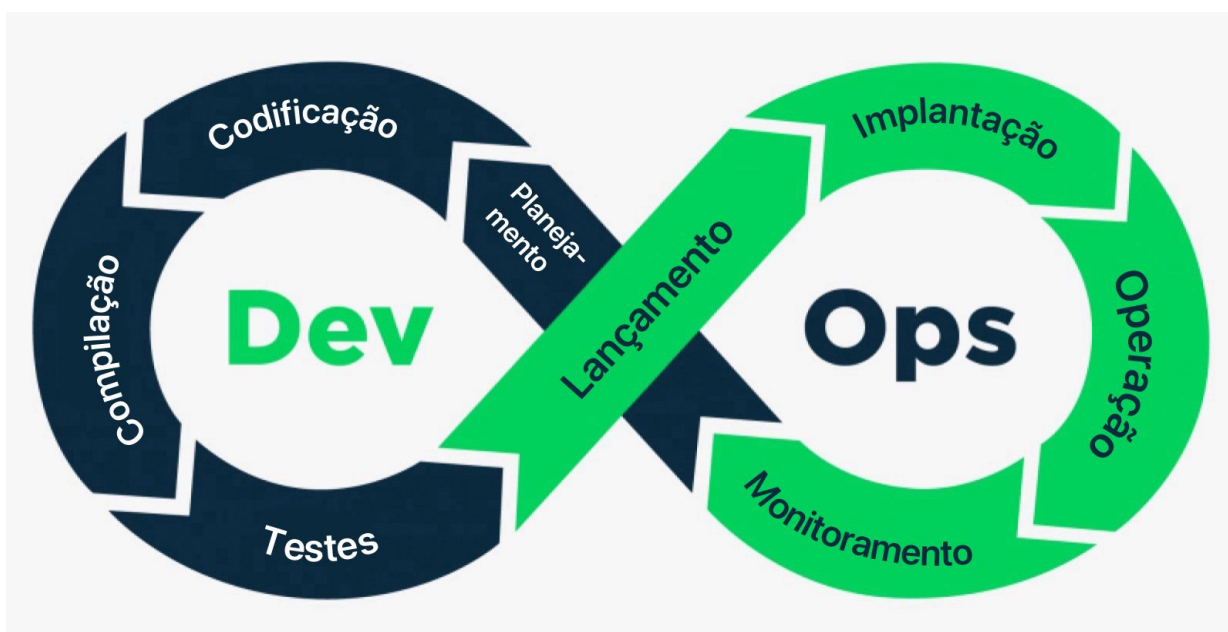


Figura 2. Fluxo *DevOps*. Fonte: [The Principles of DevOps](#). (Traduzido)

Através das práticas do *DevOps*, emerge um fluxo implícito no processo de desenvolvimento de *software*, que se divide em 8 etapas:

- **Planejamento:** Nesta etapa, a equipe define metas, objetivos, recursos e ferramentas necessárias para o projeto.
- **Codificação:** Nesta fase, a equipe desenvolve o código-fonte da aplicação, frequentemente usando sistemas de controle de versão para colaboração.

- **Compilação/Build:** Aqui, uma ferramenta de *build* é empregada para compilar o código e criar um artefato, pronto para testes e implantação.
- **Testes:** Nesta etapa, a aplicação é minuciosamente testada, assegurando que ela funcione conforme o esperado e cumpra os requisitos, utilizando tanto testes manuais, quanto automatizados.
- **Lançamento:** Durante essa fase, é gerada uma nova versão ou atualização da aplicação.
- **Implantação:** A equipe efetua a implantação da aplicação em um ambiente específico, colaborando frequentemente com equipes de operações ou infraestrutura.
- **Operação:** Nesta etapa, o desempenho da aplicação é monitorado no ambiente em que está em funcionamento, sendo tomadas medidas para assegurar seu correto funcionamento.
- **Monitoramento:** O desempenho da aplicação é continuamente monitorado, coletando dados sobre sua utilização e desempenho. Esses dados fornecem *insights* para identificação de problemas, melhorias e orientam decisões futuras de desenvolvimento.

As fases de compilação/*build*, testes, lançamento e implantação são automatizadas por meio de práticas de *CI/CD*.

3.5 Abordagem tradicional vs. *DevOps*

A visibilidade de um cenário comparativo entre um modelo tradicional e um modelo *DevOps* fica mais evidente na vivência e, na prática, porém padrões podem ser observados em quaisquer cenários onde houve a implementação do *DevOps*. Na [Tabela 2](#) apresenta-se uma lista das características da entrega de *software* do modelo tradicional vs. *DevOps*:

Característica	Tradicional	<i>DevOps</i>
Comunicação	Segregada	Colaborativa
Definição de requisitos	Definidos apenas no início do projeto	Podem ser alterados ao longo do projeto
<i>Deploy</i> em produção	Tipicamente manual	<i>CI/CD</i>
Automatização de processos	Podem existir processo automatizados	Ciclo totalmente automatizado

Colaboração entre equipes	Produto final com os requisitos desenvolvidos no início do projeto	Produto iterativo com valor crescente
Equipes de desenvolvimento	Preocupadas apenas em desenvolver os requisitos e entregá-los	Colaboram ao longo de todo o ciclo para uma maior entrega de valor
Equipes de operações	Preocupadas apenas em manter as aplicações estáveis	Gestão colaborativa com as equipes de desenvolvimento

Tabela 2. Comparação modelo tradicional vs *DevOps*.

3.6 Possíveis benefícios do *DevOps*

A adoção do *DevOps* traz consigo uma série de possíveis benefícios abrangendo as esferas técnica, cultural e de negócios. Os benefícios técnicos englobam a implementação de práticas de *CI/CD* e uma resolução mais ágil de problemas.

No âmbito cultural, promove uma comunicação e colaboração mais eficazes entre as equipes, visando deixar os colaboradores mais engajados. Já os benefícios de negócios estão diretamente ligados à entrega mais ágil de valor aos clientes, uma maior estabilidade do negócio e espaço para inovação.

Na tabela 3, é possível listar algum dos benefícios previstos descritos na literatura.

Benefício	Referência
Melhoria da comunicação entre equipes	(Amaradri & Nutalapati, 2016; Lwakatare et al., 2019; Senapathi, Buchan, & Osman, 2018)
Compartilhamento de conhecimento entre equipes	(Senapathi et al., 2018)
Maior motivação e alcance dos objetivos dos colaboradores	(Lwakatare, 2017)
Aumento do desempenho e da produtividade organizacional	(Forsgren et al., 2017; Luz, 2018; Lwakatare et al., 2019)
Integração contínua	(Amaradri & Nutalapati, 2016; Kim et al., 2016)
<i>Deploy</i> contínuo	(Amaradri & Nutalapati, 2016; Kim et al., 2016; Lwakatare et al., 2019)

Melhoria do tempo de entrega	(Lwakatare et al., 2019; Ozkaya, 2019)
Testes contínuos	(Amaradri & Nutalapati, 2016; Laihonen, 2018)
Monitoramento contínuo	(Amaradri & Nutalapati, 2016; Shahin et al., 2017)
Entrega de valor contínuo	(Lwakatare, 2017; Shahin et al., 2017)
Feedback contínuo	(Amaradri & Nutalapati, 2016; Shahin et al., 2017)
Melhoria da qualidade do <i>software</i>	(Luz, Pinto, & Bonifácio, 2018; Lwakatare et al., 2019; Senapathi et al., 2018)
Melhoria contínua (<i>lean</i>)	(Kim et al., 2016; Lwakatare et al., 2019)

Tabela 3. Benefícios da adoção do *DevOps*.

Esses possíveis benefícios, relacionados a ganhos palpáveis no negócio, são concretizados através da aplicação cuidadosa dos princípios e práticas do *DevOps*, tais como versionamento de código, integração contínua, testes contínuos, implantação contínua e monitoramento contínuo. A automação desempenha um papel crucial ao liberar recursos humanos e possibilitar a adoção consistente dessas práticas em todos os ambientes.

A entrega contínua e rápida de valor aos clientes, juntamente com o *feedback* valioso que pode ser obtido por meio de métricas e *dashboards*, constituem benefícios significativos. O *DevOps* também eleva o desempenho e a produtividade do departamento de TI, reduz custos ao longo do ciclo de vida do desenvolvimento de *software*, aprimora a eficiência operacional e fomenta uma maior sinergia entre as equipes de desenvolvimento e operações.

Em síntese, o *DevOps* proporciona vantagens substanciais em diversas áreas, impulsionando a eficiência, a colaboração, a entrega de valor e a inovação nas organizações

3.7 Desafios e barreiras à adoção

A adoção do *DevOps* enfrenta desafios que envolvem resistência à mudança, deficiências na comunicação entre equipes e a complexidade das estruturas organizacionais. É crucial investir na capacitação e motivação das pessoas para superar a resistência à mudança e fomentar a colaboração entre as equipes. Organizações de grande porte podem enfrentar obstáculos adicionais ao implementar a cultura e práticas do *DevOps*. A abordagem de implementação pode ser de cima para baixo, liderada pela alta administração, ou de baixo para cima, impulsionada pelos programadores e equipes técnicas.

Na tabela 4, algumas das eventuais barreiras e desafios encontrados para a implementação da cultura DevOps são listadas, juntamente com sua referência no estado atual da literatura.

Barreira	Referência
Prática não generalizada	(Bass et al., 2015; Lwakatare, 2017)
Resistência à mudança	(Amaradri & Nutalapati, 2016; Bass et al., 2015)
Falta de comunicação	(Amaradri & Nutalapati, 2016; Ozkaya, 2019; Riungu-Kalliosaari et al., 2016)
Evolução tecnológica	(Lwakatare, 2017; Riungu-Kalliosaari et al., 2016)
Mudança dos processos	(Lwakatare, 2017; Lwakatare et al., 2019; RiunguKalliosaari et al., 2016)
Mudança das ferramentas	(Lwakatare, 2017; Lwakatare et al., 2019; RiunguKalliosaari et al., 2016)
Dimensão da organização	(Amaradri & Nutalapati, 2016)
Falta de competências técnicas	(Lwakatare et al., 2019)

Tabela 4. Barreiras à adoção do *DevOps*.

É importante reconhecer os benefícios da adoção do *DevOps*, como a entrega mais ágil de valor aos clientes, ao mesmo tempo em que se deve estar preparado para lidar com desafios durante o processo de implementação. A evolução constante dos processos do *DevOps* requer que as organizações estejam dispostas a fazer ajustes e adaptações conforme necessário.

Essas barreiras culturais, técnicas e de negócios podem ser superadas através do aprimoramento da comunicação entre as equipes, da promoção da colaboração e do compartilhamento de responsabilidades. As barreiras técnicas podem ser transpostas por meio de treinamento apropriado e pela adoção de novas ferramentas. As barreiras de negócios, embora mais desafiadoras, podem ser abordadas pela criação de equipes multidisciplinares que trabalham em projetos conjuntos, com foco na entrega de valor alinhada aos requisitos do negócio.

3.8 DevOps e Automação são nossa próxima bala de prata?

Em seu livro "*The Mythical Man-Month*", Fred Brooks destacou que não existe uma solução mágica para alcançar uma substancial melhoria na produtividade de *software*. Ele enfatizou a importância de eliminar obstáculos artificiais que dificultam as tarefas acidentais, como restrições de *hardware*, linguagens de programação de baixo nível e/ou não úteis para o contexto atual da aplicação em questão, e falta de tempo de máquina.

No contexto do desenvolvimento de *software*, o propósito dos processos é justamente remover essas barreiras, com o intuito de entregar *software* de melhor qualidade, de forma mais rápida e econômica aos usuários. A automação desempenha um papel crucial nesse contexto, ao automatizar tarefas repetitivas e bem definidas. O movimento *DevOps* segue o princípio de automatizar qualquer tarefa que seja realizada mais de duas vezes. Isso facilita a comunicação objetiva através de artefatos, economizando tempo e evitando interpretações equivocadas.

O *DevOps* é definido como um conjunto de práticas que visa diminuir o intervalo entre a validação de uma alteração no sistema e sua implantação em produção, mantendo um alto padrão de qualidade. Embora não seja em si um processo de desenvolvimento de *software*, ele complementa esses processos ao eliminar transferências e atividades acidentais.

Ao longo das últimas décadas, testemunhamos o surgimento e declínio de várias metodologias de desenvolvimento de *software*, porém nenhuma delas proporcionou uma melhoria significativa na produtividade. O sucesso dos processos ágeis e enxutos deve-se, em parte, à remoção de barreiras, como a sobrecarga de documentação, substituindo-a por demonstrações de *software* funcional. No entanto, desafios persistentes ainda existem, como a adaptação ao contexto específico e uma comunicação eficaz.

A automação e o *DevOps* podem contribuir para a redução de erros acidentais e otimizar tarefas essenciais, como o design, resultando em economia de tempo valioso. Entretanto, não existe uma solução única e definitiva para aprimorar a produtividade. Cada organização deve compreender seu contexto e gerenciar eficazmente os canais de comunicação essenciais. Os processos ágeis e enxutos de desenvolvimento de *software* enfatizam a comunicação frequente e precisa entre todas as partes interessadas e membros da equipe.

Apesar dos benefícios da automação na redução do tempo de ciclo e do desperdício, ela não representa a solução final. Embora haja relatos de êxito na diminuição de custos e retrabalho, também há casos de insucesso devido à falta de compreensão do contexto, comunicação inadequada e escassez de colaboração. Portanto, não devemos esperar que a automação e o *DevOps* sejam a resposta definitiva para a melhoria da produtividade.

4. EXECUÇÃO DA METODOLOGIA

4.1 Seleção dos casos e contextualização

O caso selecionado é a implementação da cultura DevOps numa empresa brasileira, de grande porte, possuindo mais de 500 funcionários no time de engenharia, e mais dezenas de funcionários em seus outros setores não correlacionados com seus sistemas de TI. Sua área de atuação é diversificada, variando entre *e-commerce*, logística e instituição financeira. A infraestrutura da empresa é robusta, utilizando a *Azure* como plataforma *Cloud* e a arquitetura de microserviços para suas aplicações e sistemas internos.

A empresa escolhida não possuía qualquer método ágil correlacionado com DevOps, seu primeiro contato foi com o processo descrito nesse estudo. Tendo em vista que a empresa não possuía um processo de desenvolvimento de *software* com um método correlato ao *DevOps*, a análise desse estudo nos possibilitou estudar os períodos antes/depois/após implementação.

A empresa em questão, possui em seu corpo de engenharia, um time composto apenas por engenheiros *DevOps*, trabalhando como Engenheiros de Confiabilidade (*SREs*), Engenheiros de Plataforma (*Platform Engineers*) e Engenheiros de Nuvem (*Cloud Engineering*), e esse time é responsável pela desenvolvimento de algumas aplicações, monitoramento e otimização de caminhos. Nesse estudo, será considerado o ambiente e ferramentas utilizadas por essa equipe. O autor deste estudo obteve acesso a documentações, ferramentas, *logs* e conversas com diretores e coordenadores da área.

A análise dos dados adotada é qualitativa com fins descritivos, onde foi realizada um questionário sobre o assunto em pauta, juntamente aos membros das equipes correspondentes ao processo de implementação do *DevOps*.

Busca-se manter a integridade e sigilo dos dados, não divulgando informações que possam ferir a *LGPD* (Lei Geral de Proteção de Dados).

4.2 Elaboração do Protocolo

Neste estudo, já foram apresentadas a questão de pesquisa e a seleção dos casos, e a apresentação dos resultados será feita no próximo capítulo. Agora, serão apresentados o

procedimento de trabalho de campo e as questões de estudo, juntamente com o roteiro criado para a coleta de dados.

Os procedimentos de campo utilizados dividem-se em dois tipos:

- Observação direta no local pelo ponto de vista do autor;
- Realização de entrevistas com os funcionários das equipes em questão.

A observação direta é efetuada ao nível dos processos, dos comportamentos e das ferramentas utilizadas, visando avaliar o sucesso da adoção da cultura *DevOps*, com base no antes e depois.

As entrevistas foram planejadas e executadas tendo como foco grupos específicos de pessoas. Esses grupos referem-se aos desenvolvedores, Engenheiros de Confiabilidade (*SREs*), Engenheiros de Plataforma (*Platform Engineers*) e Engenheiros de Nuvem (*Cloud Engineering*). Em todos os grupos, pessoas que participaram diretamente da implementação (com conhecimento efetivo sobre a cultura implementada, os seus moldes, pré e pós-implementação) foram priorizadas para responderem o questionário. Cada entrevista tem como base o mesmo roteiro e questões.

O grupo de entrevistados é composto por 14 pessoas da mesma empresa, provenientes de diferentes áreas, a fim de obter opiniões complementares. As equipes de implementação da cultura têm uma visão abrangente sobre o conceito, técnicas, ferramentas e práticas do *DevOps*, pois foram responsáveis pela maioria das iniciativas relacionadas.

O *scrum master* desempenha um papel fundamental na equipe, tendo sido responsável por estabelecer todos os processos e ferramentas a serem utilizados, os quais são agora coordenados por ele. Os membros da equipe envolvidos no suporte do *SDLC* (*Software Development Life Cycle*, ou Ciclo de vida de desenvolvimento de *software*) têm uma compreensão mais aprofundada dos procedimentos anteriores e uma visão mais centralizada, tanto na fase pré-*DevOps* quanto na fase pós-*DevOps*.

Os gestores de projeto têm visão global de todos os processos e ajudam na tomada de decisões e aconselhamento. São elementos externos à equipe, servindo como suporte para o gerenciamento da atribuição e priorização de atividades dos ciclos de entrega.

O grupo de entrevistados encontra-se caracterizado na Tabela 5.

ID	Área	Função	Experiência (Anos)
Entrevistado 1	Equipe <i>Dev</i>	<i>Front-end</i>	2
Entrevistado 2	Equipe <i>Dev</i>	<i>Full-stack</i>	4
Entrevistado 3	Equipe <i>Dev</i>	<i>Full-stack</i>	2
Entrevistado 4	Equipe <i>Dev</i>	<i>Full-stack</i>	5
Entrevistado 5	Equipe <i>SRE</i>	Observabilidade	6
Entrevistado 6	Equipe <i>SRE</i>	<i>CI/CD</i>	1
Entrevistado 7	Equipe <i>SRE</i>	<i>CI/CD / Scrum Master</i>	9
Entrevistado 8	Equipe <i>SRE</i>	Automação	10
Entrevistado 9	Equipe <i>SRE</i>	Banco de dados	7
Entrevistado 10	Equipe <i>Cloud</i>	Redes	8
Entrevistado 11	Equipe <i>Cloud</i>	Redes	2
Entrevistado 12	Equipe <i>Cloud</i>	Segurança	6
Entrevistado 13	Gestão	Gestor de projeto	5
Entrevistado 14	Gestão	Gestor de projeto	3

Tabela 5. Identificação dos entrevistados.

4.2.1 Entrevista

O questionário desenvolvido (consultar Apêndice) foi utilizado para os 14 entrevistados darem sua opinião a respeito de todo o ciclo de implementação, e de suas opiniões a respeito desse mesmo ciclo.

Ele foi estruturado em 9 seções:

- 1: Definição do *DevOps*;
- 2: Pré-*DevOps*;
- 3: Implementação do *DevOps*;
- 4: Práticas *DevOps*;

- 5: Técnicas e Ferramentas do *DevOps*;
- 6: Benefícios do *DevOps*;
- 7: Gestão da mudança;
- 8: Barreiras da implementação;
- 9: Pós-Implementação.

Cada seção se propõe a contextualizar uma das fases do processo de implementação, desde a definição do conceito até os resultados da pós-implementação.

5. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo são apresentados os resultados relativos às entrevistas, cujo objetivo é compreender a adoção da cultura *DevOps* por parte da entidade estudada. Os resultados são apresentados seguindo a mesma estrutura do questionário, com o título da seção representado por 5.x e a pergunta feita dentro da seção representado por 5.x.y (ver Apêndice).

Nestes excertos, os entrevistados são sempre identificados com a letra que lhes foi atribuída na Tabela 5.

5.1 Definição do *DevOps*

A primeira seção é relativa ao conhecimento dos entrevistados sobre o tema, abordagens, ciclo, práticas e princípios.

5.1.1 O que é *DevOps*?

A interpretação do *DevOps* pode variar, e não há uma resposta única e definitiva. Ele engloba a integração das equipes, aprimoramento da comunicação, aceleração e aprimoramento das entregas, otimização dos processos e redução de custos. Como destacado por um dos membros da equipe de *SRE*:

"O DevOps transcende a simples automatização e monitoramento de processos", de acordo com um membro da equipe de testes automatizados. É uma abordagem abrangente que envolve a colaboração e a integração contínua entre as equipes de desenvolvimento e operações, com o objetivo de aprimorar a entrega de software, a qualidade e a eficiência dos processos.

5.1.2 Melhor abordagem (a nível organizacional) para a adoção do *DevOps*

No contexto de gestão, a abordagem top-down (de cima para baixo) é quando as decisões de toda a empresa são feitas pela liderança, no topo, enquanto a abordagem bottom-up (de baixo para cima) dá voz a todas as equipes nesse tipo de decisão.

Na Figura 3.1 é possível verificar que 64,3% dos entrevistados considera que a melhor abordagem para a adoção da cultura *DevOps* é a *Bottom-Up*, 28,6% considera que é a *Top-Down*, e 7,1% considera que existem outras abordagens capazes de alavancar a cultura.

Um dos entrevistados respondeu: “Ambos, pois um processo de *DevOps* sem apoio da diretoria/administração é apenas um time implementando algo, mas não quebrando silos” (ID 7).

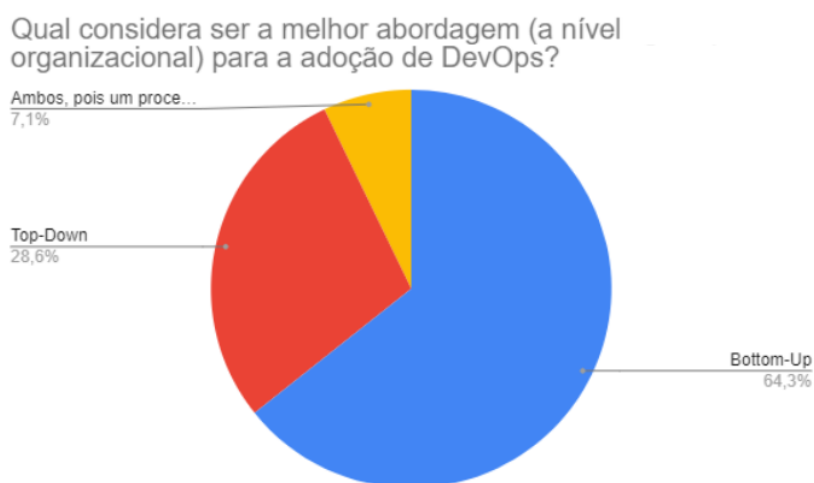


Figura 3.1 Melhor abordagem (a nível organizacional) para a adoção do *DevOps*?

Não existe uma forma correta de se implementar esta cultura, ambas as perspectivas são válidas e têm vantagens diferentes dependendo da maturidade da organização. A abordagem *bottom-up* representa a colaboração dos intervenientes no processo de mudança para alcançar a administração através dos diversos patamares hierárquicos.

No contexto da empresa do estudo, a abordagem padrão adotada foi a *top-down* para o planejamento da implementação, e a *bottom-up* para a implementação de fato.

Um dos membros da equipe de implementação defendeu o padrão *bottom-up*:

“DevOps consiste em pessoas, processos e ferramentas. E toda revolução é melhor iniciada da base para o topo” (ID 7).

No entanto, existe a perspectiva oposta, onde todo o suporte para a adoção desta cultura advém da administração através de todos os níveis da hierarquia, até chegar à equipe implementadora. O ponto de vista do *scrum master* é o seguinte:

“Com a abordagem *bottom-up* na cultura *DevOps*, é esperado encontrar mais resistência à medida que se sobe na hierarquia da organização. Por outro lado, a adoção de

novas metodologias e tecnologias é facilitada e acelerada quando há um apoio e incentivo da alta hierarquia desde o início do processo.” (ID 8)

5.1.3 Fases com maior risco na implementação do *DevOps*

Estas fases contemplam riscos de implementação distintos, consoante a organização em questão. Os entrevistados apontaram as diferentes fases nas suas respostas, como mostra o gráfico apresentado na Figura 3.2, tendo 21,4% dos entrevistados, considerado que as fases com maior risco de implementação são as fases de planeamento e *deploy*, seguida da fase de *deploy* e *operate* com 14,3%. Todas as outras fases foram consideradas terem risco igual, com 7,1%.

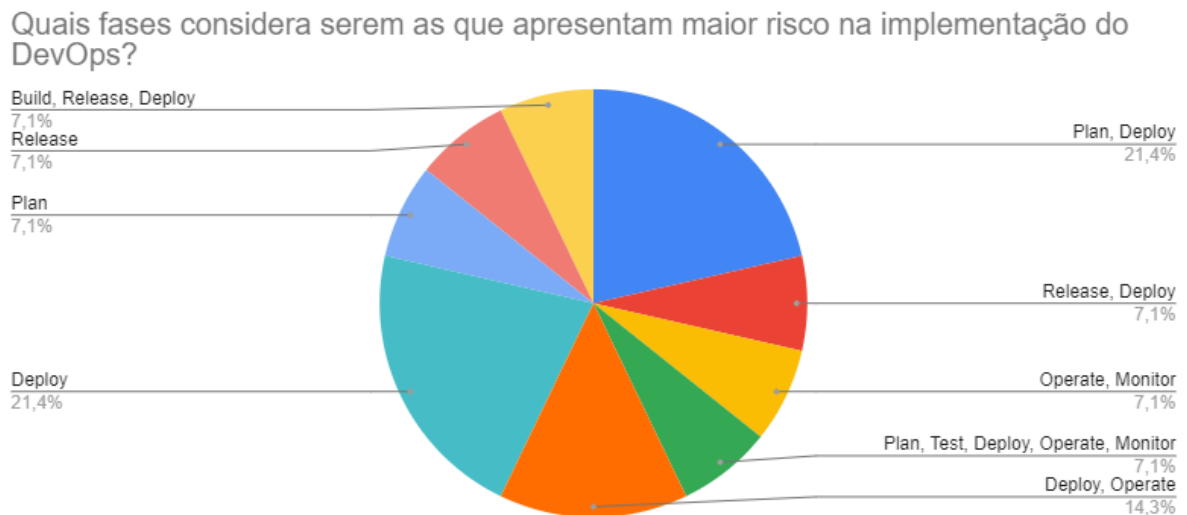


Figura 3.2 Fases com maior risco na implementação do *DevOps*

O planeamento é uma fase de extrema importância, conferindo assim o maior risco de implementação, visto que um mau planeamento pode levar todo o processo a ser refeito.

5.1.4 Práticas do *DevOps* mais impactantes (positivamente) para a implementação do *DevOps*

A Figura 3.3 apresenta as respostas dos entrevistados quanto às práticas mais impactantes (de forma positiva) para a implementação do *DevOps*. No topo foi escolhida a fase de planejamento, com 22,2% dos entrevistados considerando que esta é a prática mais impactante. Empatadas em 16,7%, estão as práticas de integração contínua, implantação contínua e desenvolvimento colaborativo. Os testes automatizados foram considerados por 11,1% dos entrevistados. As práticas de entrega contínua, *feedback* contínuo do cliente e monitorização contínua, empataram em 5,6%.

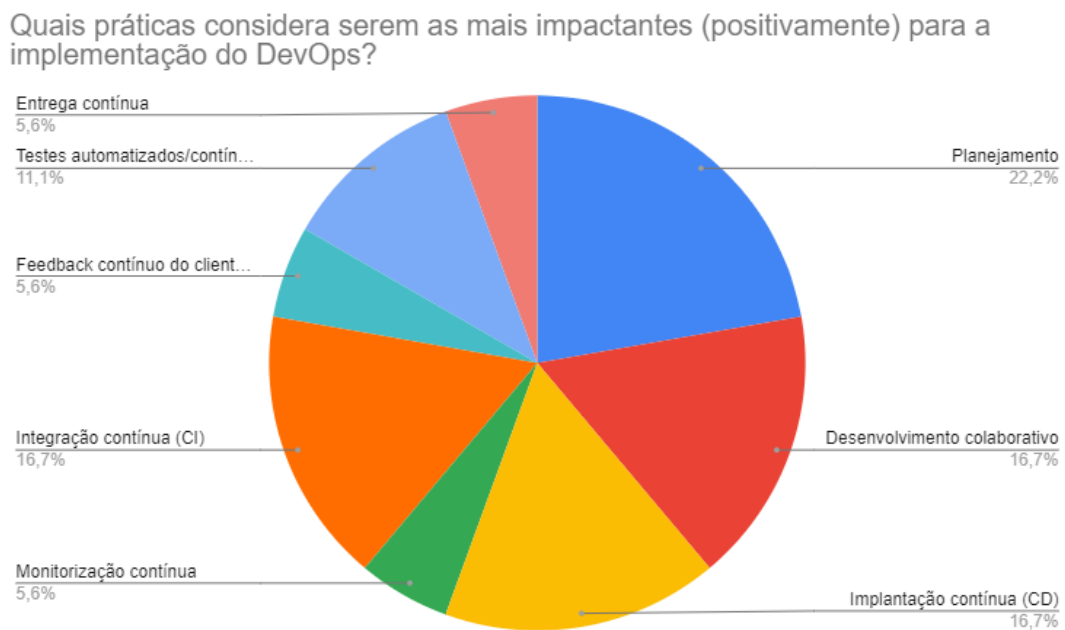


Figura 3.3 Práticas do *DevOps* mais impactantes (positivamente) para a implementação do *DevOps*.

Com isso, se reforça a importância de um planejamento robusto, juntamente com um desenvolvimento colaborativo e ferramentas (CI/CD) que facilitem e agilizem o processo.

5.2 Pré-DevOps

Na segunda seção do questionário, foi pedido que os entrevistados falassem sobre a experiência anterior à adoção da cultura *DevOps*, tendo em conta ferramentas, práticas e princípios anteriores que suportassem o ciclo de desenvolvimento de *software*.

5.2.1 Motivações para a implementação do *DevOps*

Os entrevistados classificam as motivações para esta implementação como solução para os problemas de comunicação e barreiras existentes entre as equipas inerentes ao *SDLC* da organização.

No cenário anterior a implementação, era possível identificar silos entre as equipas e um processo não otimizado e eficaz, como refere um dos entrevistados, a grande motivação foi:

“Dificuldades nas entregas e falta de comunicação entre as equipas de desenvolvimento e operações, devido à diferença de linguagem e conhecimento técnico.” (ID 3).

A cultura *DevOps* visa mitigar os problemas de comunicação referidos pelos entrevistados com vista a unificar as equipas e por forma a trabalharem mais colaborativamente.

5.2.2 Ferramentas utilizadas para gerir o *SDLC* anteriores à implementação do *DevOps*

Existem processos e ferramentas inerentes à cultura *DevOps*, como, por exemplo, um repositório central de código, um orquestrador de compilação, e *deploy* e ferramentas de monitoria.

Antes da implementação do *DevOps*, na empresa em questão, eram utilizadas algumas ferramentas que são consideradas válidas numa implementação *DevOps*, como o orquestrador Jenkins para processos de automatização de algumas compilações e instalação de produtos, porém nenhum processo mais maduro que se equipare ao fluxo completo do *DevOps*.

5.2.3 Processos semelhantes a *DevOps* anteriores à sua implementação

Antes da adoção da iniciativa do *DevOps*, foi identificada a existência de processos alinhados com essa cultura, como o planejamento, desenvolvimento colaborativo, integração contínua (CI) e monitoramento contínuo, conforme se observa na Figura 3.4, mencionado por 42,3% e 15,4% dos entrevistados, respectivamente.



Figura 3.4 Processos da organização semelhantes a um processo inerente à cultura *DevOps*.

*Fatia cortada: “Feedback contínuo do cliente e otimização de chamados”

A implantação contínua, juntamente com testes automatizados e *feedback* contínuo, embora sejam práticas do *DevOps*, podem levantar dúvidas, pois a organização já possuía alguns desses processos de forma manual, e não automatizada. Apesar de não serem processos do *DevOps*, são práticas similares que facilitaram a transição para a adoção das práticas mencionadas, reduzindo a curva de aprendizado.

5.3 Implementação do *DevOps*

Esta seção é referente à implementação da cultura *DevOps*, apresentando-se diferentes aspectos da mesma, como a necessidade de seguir as boas práticas ou a necessidade da participação de entidades externas na implementação.

5.3.1 Alternativas à adoção do *DevOps*

Na Figura 3.5 é apresentado o gráfico das alternativas à adoção do *DevOps*.

Na sua organização, foram consideradas outras alternativas à adoção do *DevOps*?

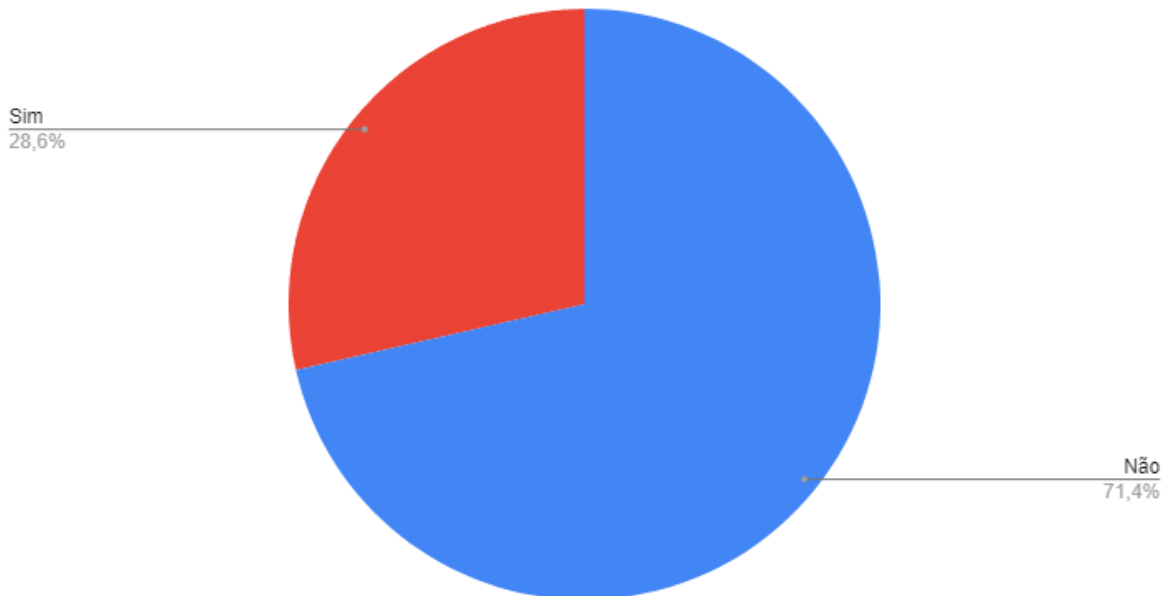


Figura 3.5 Alternativas à adoção do *DevOps*

71,4% dos entrevistados afirmam que não foi considerada outra alternativa à adoção da cultura *DevOps*, enquanto 28,6% referem que foi considerada uma alternativa, que atribui mais responsabilidades para os desenvolvedores de forma que todos tenham conhecimento e *perícia* nos pontos do processo de desenvolvimento, desde o planejamento até a manutenção de operações. Porém, essa outra alternativa não teve muita força e jamais saiu do papel.

5.3.2 Objetivos da implementação do *DevOps*

Os entrevistados referiram a velocidade da entrega e a redução de falhas como maiores objetivos de implementação. Um dos entrevistados, o *product owner* afirma que os objetivos de implementação foram:

“O primeiro objetivo é técnico: envolvendo automatizações de processos repetitivos diariamente e melhora da eficiência de recursos alocados, o segundo é cultural: desenvolvimento colaborativo, aumentando a colaboração entre as equipes.” (ID 11).

Os objetivos técnicos e culturais, como listados pelo entrevistado 11, podem ser observados na Tabela 3, reforçando a ideia dos benefícios em produtividade e colaboração.

5.3.3 Melhores práticas na implementação do *DevOps*

Foi perguntado se algum guia de boas práticas foi utilizado para o processo de implementação do *DevOps*, e observa-se na Figura 3.6 que 71,4% dos entrevistados afirmam que foi seguido um guia de boas práticas, enquanto 28,6% afirmam que não foi seguido nenhum guia.

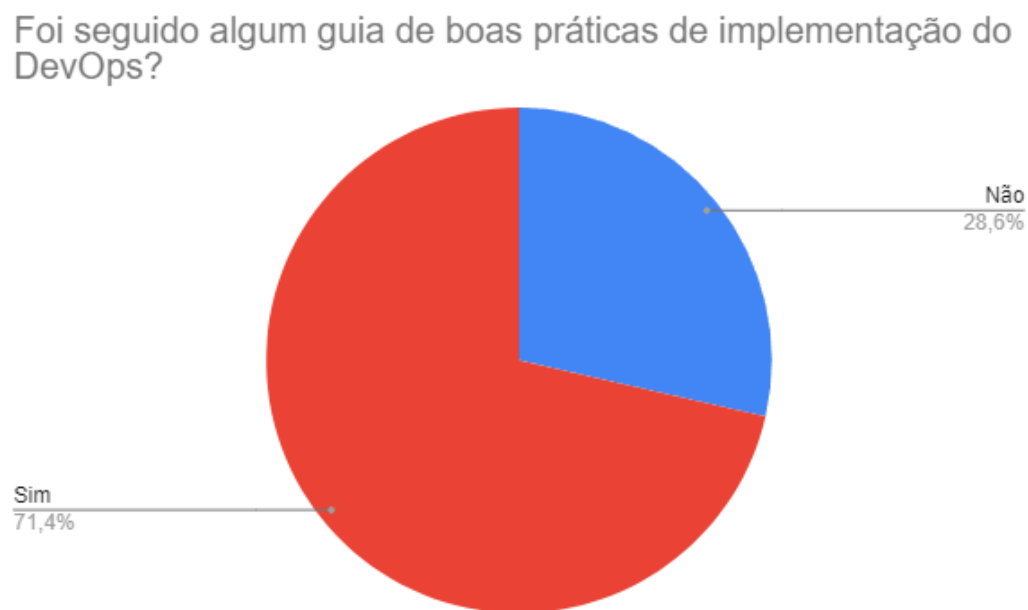


Figura 3.6 Utilização de guia de boas práticas de implementação do *DevOps*

Um dos membros que ajudou na implementação:

“Baseado na experiência de implementação de *CI/CD* em projetos semelhantes, fóruns e eventos da comunidade *DevOps*, pudemos ter grandes referências de boas práticas e conseguimos criar nossas próprias boas práticas” (ID 7).

No fim, não foi utilizado nenhum guia ou manual comercial, e sim vários guias e fóruns online da própria comunidade.

5.3.4 Participação de entidades externas



Figura 3.7 Necessidade de participação de entidades externas no processo de implementação

Pouco existiu a necessidade da participação de entidades externas nesta implementação, uma vez que a equipe teve total acesso de gerenciamento das ferramentas e processos implementados, mais a frente serão abordadas as dificuldades encontradas.

A própria equipe foi responsável pela maior parte das responsabilidades, apenas o suporte de uma das ferramentas de monitoramento (*New Relic*) foi contatado para ajudar na integração dos serviços com o monitoramento.

5.3.5 Custo estimado da implementação do *DevOps*

Dado o contexto, não foi possível ser estimado um valor para o custo global do projeto de implementação, pois este contempla várias fases e variáveis, e boa parte dos custos foram visíveis apenas para a diretoria, onde a equipe de implementação não teve acesso completo.

O maior custo é o de esforço-humano, onde se precisou recriar boa parte dos processos, focando em automação e determinação de padrões como retroalimentação para essas mesmas automações.

5.3.6 Área(s) com maior mudança organizacional/cultural

Na Figura 3.7 é apresentado o gráfico que determina as áreas, que os entrevistados consideram, onde existiu maior mudança organizacional e/ou cultural.

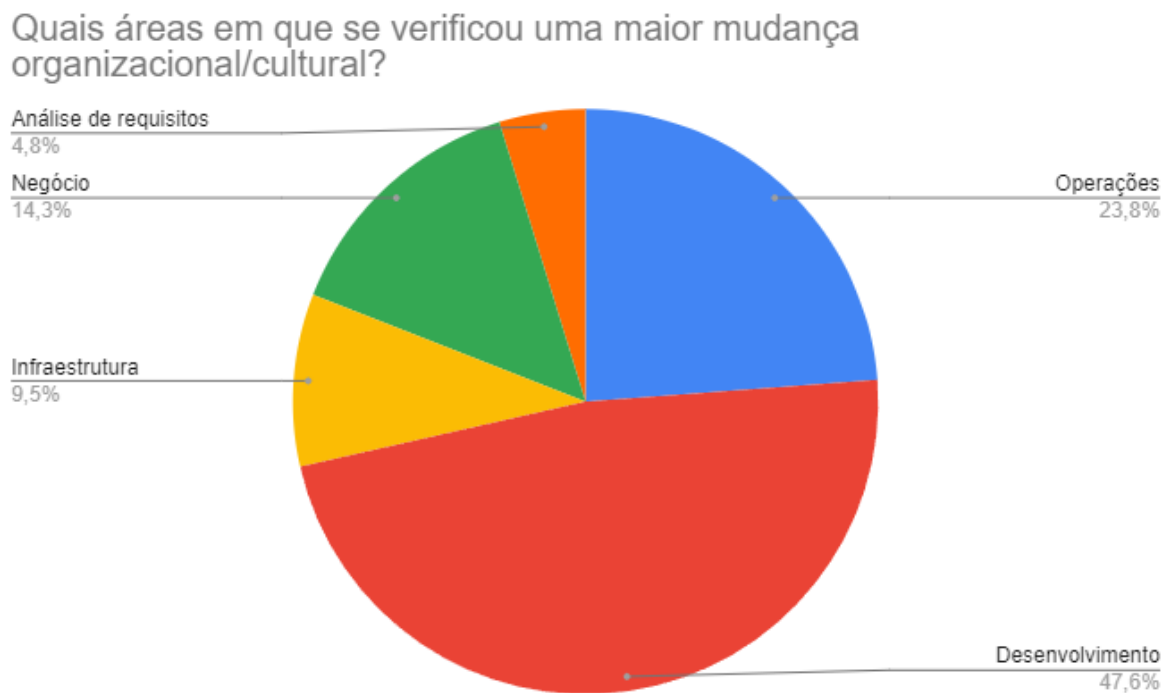


Figura 3.8 Áreas com maior mudança organizacional/cultural

A área onde se verificou maior mudança foi no desenvolvimento, como pode ser observado em algumas respostas, como:

“O padrão pré-implementação era o de silos entre equipes, com a implementação do *DevOps*, a colaboração entre os times ficou cada vez maior.” (ID 6)

Com o apoio *Top-Bottom*, o processo de adoção do *DevOps* foi iniciado na equipe de desenvolvimento e, em seguida, estendido para o suporte de ambientes, na equipe de operações. Conforme o processo amadurecia em cada fase, houve progresso tanto na parte de planejamento quanto na parte de gerenciamento de *releases*.

5.3.7 Tipologia do processo implementado

A Figura 3.9 apresenta qual a tipologia de processo implementado, sendo ela puramente *DevOps* ou híbrida (junção do método *DevOps* com algum outro, ou até mesmo junção de métodos diferentes não inerentes ao *DevOps*).

Considerando a literatura, o processo implementado na sua organização é:

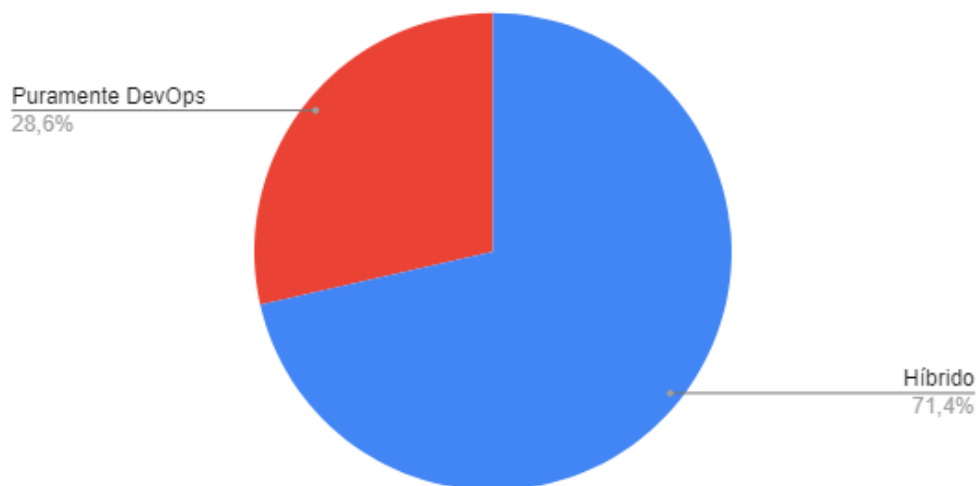


Figura 3.9 Tipologia do processo implementado

Apesar de considerado por vários elementos que o processo implementado foi puramente *DevOps*, o *scrum master* defende que:

“Na prática, é pouco comum que uma adoção do *DevOps* seja bem-sucedida se for puramente focada no *DevOps*, deixando de lado completamente o ambiente anterior a

implementação. Uma abordagem híbrida, em que a adoção ocorre gradualmente conforme a transformação dos projetos para o novo processo *DevOps*, resulta em uma maior conscientização de todas as equipes envolvidas e suaviza a transição, permitindo a demonstração de resultados ao longo do tempo.” (ID 8)

5.4 Práticas *DevOps*

Nesta quarta seção são caracterizadas as práticas implementadas, e os fatores influenciadores de sucesso na organização.

5.4.1 Práticas implementadas

Através da observação e contato direto com os processos e ferramentas, foram identificadas as seguintes práticas implementadas pela organização:

- Planejamento;
- Desenvolvimento colaborativo;
- Integração contínua (*CI*);
- Entrega contínua (*CD*);
- Testes contínuos;
- Monitoramento contínuo.

Estas práticas permitem à organização alcançar um maior nível de maturidade de automatização e de gestão do *SDLC*.

5.4.2 Principais fatores influenciadores do sucesso da implementação do *DevOps*

Foi questionado quais os principais fatores influenciadores para uma boa implementação do *DevOps*, os destacados foram:

- Espírito de equipe e entreajuda;
- Qualidade da infraestrutura;
- Apoio da gestão de topo;

- Comunicação e entendimento da equipe;
- Abertura para mudanças;
- Experiência;
- Treinamentos;
- Definição dos objetivos e planejamento;
- Cultura ágil;
- Tecnologias que habilitem as práticas;
- Análise de impacto;
- Automatização;
- Tomada de decisão.

Os participantes referiram que estes fatores se verificam na organização como um todo.

5.5 Técnicas e Ferramentas do DevOps

A quinta seção é referente ao conhecimento das técnicas e ferramentas escolhidas pela organização.

5.5.1 Implementação do *DevOps*

A Figura 3.10 apresenta o tipo de implementação do *DevOps* realizado pela organização.

Tendo em conta os processos e as ferramentas da organização, como foi implementado o DevOps?

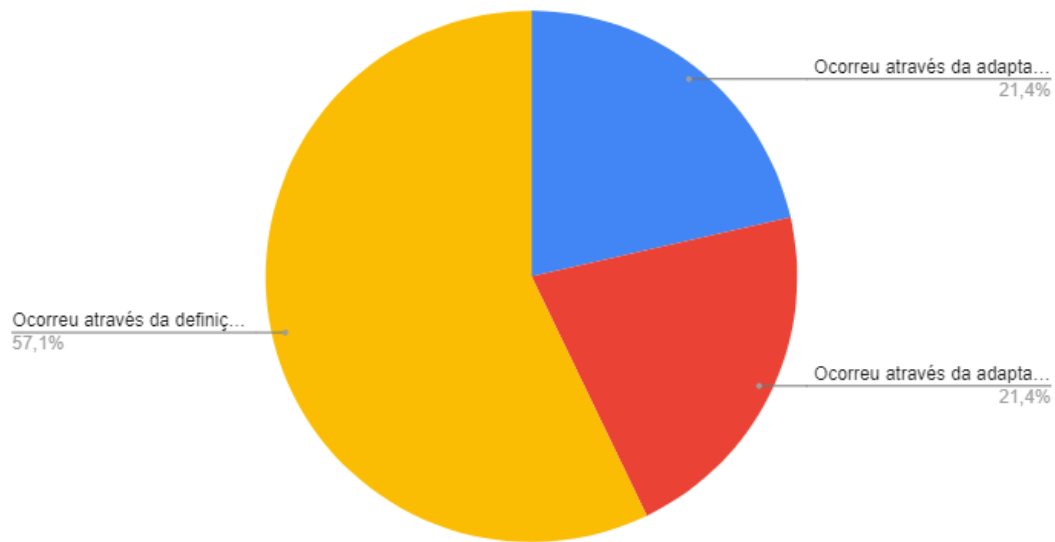


Figura 3.10 Implementação do *DevOps*.

57,1% dos entrevistados afirmaram que a implementação ocorreu por meio da definição de novos processos e da introdução de novas ferramentas, enquanto 21,4% mencionaram que houve a necessidade de adaptar os processos e ferramentas já existentes com novos elementos.

Na prática, a implementação ocorreu por meio da definição de novos processos e introdução de novas ferramentas, visto que o cenário anterior a implementação, muitos pontos de falha e melhoria eram identificados, e ao adotar novas ferramentas e processos, a mitigação desses pontos pôde ser melhor gerenciada.

Os processos e ferramentas empregados na adoção do *DevOps* variam de uma organização para outra. Não existe um conjunto padrão de ferramentas a ser seguido, uma vez que essa escolha depende das necessidades e da infraestrutura específica de cada empresa. Nesse contexto, a organização já fazia uso de várias ferramentas compatíveis com a cultura *DevOps*, como o Jenkins, o que facilitou a incorporação dos novos processos às ferramentas existentes.

Por outro lado, foi necessário adotar novas ferramentas para dar suporte aos processos já em vigor. Além disso, foram desenvolvidos novos processos e introduzidas ferramentas adicionais para sustentar as práticas específicas do *DevOps* implementadas pela organização.

Essa abordagem personalizada permitiu que a empresa utilizasse as ferramentas mais adequadas às suas necessidades e integrasse os processos já existentes de maneira eficaz.

5.5.2 Ferramentas implementadas

Na Figura 3.11 estão representadas as categorias das ferramentas implementadas.



Figura 3.11 Categorias das ferramentas implementadas.

Foram implementadas ferramentas de todas as categorias referidas na Figura 3.11. Por serem uma adoção recente, as ferramentas de orquestração de *release* e de auditoria de código são as que apresentam menor expressão.

Após a categorização das ferramentas, foi feita uma varredura no ambiente para identificar todas as ferramentas utilizadas na e para a implementação, elas sendo:

- **Gerenciamento de projeto:** *Jira, Confluence*
- **Versionamento e colaboração:** *Github*
- **Orquestração de contêineres:** *Kubernetes, Rancher*
- **CI/CD:** *Azure Pipelines, Azure DevOps*
- **Testes automatizados:** *Selenium*

- **Monitoramento:** *New Relic, Dynatrace, Sentry*
- **Segurança e qualidade de código:** *Snyk, Codacy, SonarQube, ZeroTrust*
- **Experiência de desenvolvimento:** *Backstage*

Estas ferramentas permitiram a adoção da cultura *DevOps*.

O *Jira* permite a gestão de todo o planejamento da organização, desde a organização dos itens de trabalho até à gestão das releases. Dentro da mesma temática, o planejamento, o *Confluence* permite gerir toda a documentação associada a um projeto.

O *Github* é um sistema de controle de versões adotado pela organização. A infraestrutura das aplicações é suportada pelo *Kubernetes*, em uma instância *AKS* (*Azure Kubernetes Service*), que serve como motor da containerização, e tem o *Rancher* integrado servindo como orquestrador.

O *Azure Pipelines* e *Azure DevOps* permitem o controle de *build* e *deploy* das aplicações, assim como a configuração de conexões entre serviços.

O *Selenium* foi utilizado como principal provedor de testes automatizados.

As ferramentas de monitoramento escolhidas foram o *New Relic* e o *Dynatrace*, sendo a primeira a ferramenta-núcleo de monitoramento de métricas, *logs* e alertas para todas as aplicações. Já a segunda é mais voltada para o monitoramento da infraestrutura e sua saúde.

O *Sentry* e o *Snyk* permitem a visualização e acompanhamento em tempo real de vulnerabilidades e falhas de segurança em aplicações ou componentes das aplicações, assim como também é auditado através das ferramentas *Codacy, ZeroTrust, SonarQube*.

5.5.3 Utilização do Provedor de Nuvem (Cloud)

A plataforma *Cloud* escolhida para a implementação foi a *Microsoft Azure*, que provê todas as ferramentas e estruturas necessárias para hospedagem, manutenção e visualização dos serviços e sistemas desenvolvidos pela organização.

5.5.4 Sistema de controle de versões

A adoção de um sistema de controle de versões é um elemento chave na adoção da cultura *DevOps*, independentemente até mesmo de praticar ou não *DevOps*, no contexto dessa empresa, o sistema de versionamento utilizado foi o *Azure Repos*.

Um sistema de controle de versões é essencial para qualquer organização, pois oferece benefícios significativos. Ele garante que os desenvolvedores trabalhem de forma colaborativa e evita conflitos no código-fonte, uma vez que cada alteração é rastreada e gerenciada. Além disso, um sistema de controle de versões centralizado torna o código facilmente acessível a todos os membros da equipe. Isso é especialmente importante em ambientes de desenvolvimento distribuídos ou remotos.

Um dos principais benefícios do controle de versões é a possibilidade de implementar práticas de Integração Contínua (*CI*). Com um sistema de controle de versões adequado, é possível automatizar a integração de alterações de código de diferentes desenvolvedores em um fluxo contínuo. Isso permite detectar e corrigir problemas rapidamente, além de fornecer um histórico detalhado de todas as alterações feitas no código.

5.6 Benefícios *DevOps*

Nesta seção são apresentados os principais benefícios, como a colaboração entre as equipes, as correções de emergência em produção, a satisfação do cliente e das equipes de desenvolvimento e operações, e da organização como um todo. Aborda-se, também, a quantidade de entregas e a automatização dos processos existentes.

5.6.1 Principais benefícios da adoção do *DevOps*

Dentre os principais benefícios da adoção do *DevOps*, foram listados pelos entrevistados:

- Entregas mais rápidas;
- Eficiência;
- União equipes de Desenvolvimento e Operações;

- Fiabilidade do processo;
- Automatização dos processos;
- Melhoria do desempenho e da produtividade;
- Melhoria da colaboração;
- Aumento do *time to market*;
- Unificação dos processos;
- Partilha responsabilidade de todos os envolvidos nas entregas;
- Redução de erros em produção.

Foi também pedido, dada uma lista de fatores anteriormente já identificados, que escolhessem os que considerassem mais benéficos após a adoção de uma cultura *DevOps*.

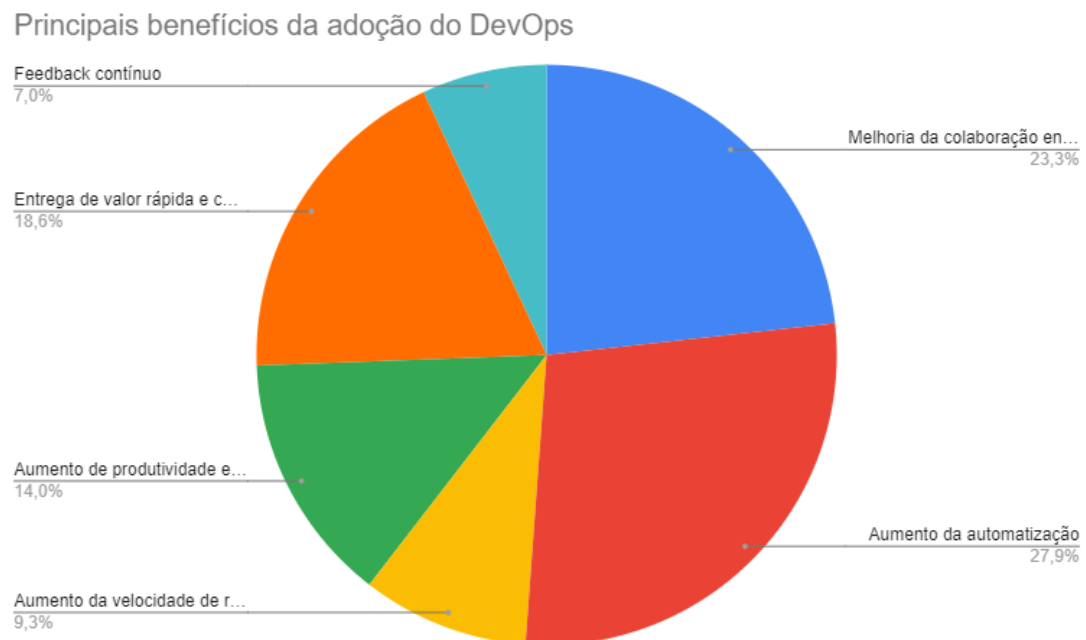


Figura 3.12 Principais benefícios da adoção do *DevOps*.

Como se pode ver na Figura 3.12, 27,9% dos entrevistados indicam que o maior benefício é o aumento da automatização, seguido da melhoria da colaboração entre as equipas com 23,3%. Logo atrás, com 18,6%, é considerada a entrega de valor rápida e contínua. O aumento de produtividade e desenvolvimento, foi indicado por 14% dos entrevistados.

O aumento da velocidade de resolução de problemas com 9,3% e o feedback contínuo com 7% foram os outros pontos elencados como principais benefícios da adoção do *DevOps*.

Com os benefícios listados, pode-se inferir que a utilização de boas práticas juntamente com boas ferramentas, em um processo bem definido, pode impactar diretamente na qualidade final do produto, assim como aumentar a eficiência da linha de produção adotada pela empresa.

5.6.2 Colaboração entre equipes

Foi questionado como as equipes passaram a trabalhar após a implementação, na Figura 3.13, observa-se que 85,7% dos entrevistados consideram que as equipes passaram a ter uma forma mais colaborativa de trabalhar.

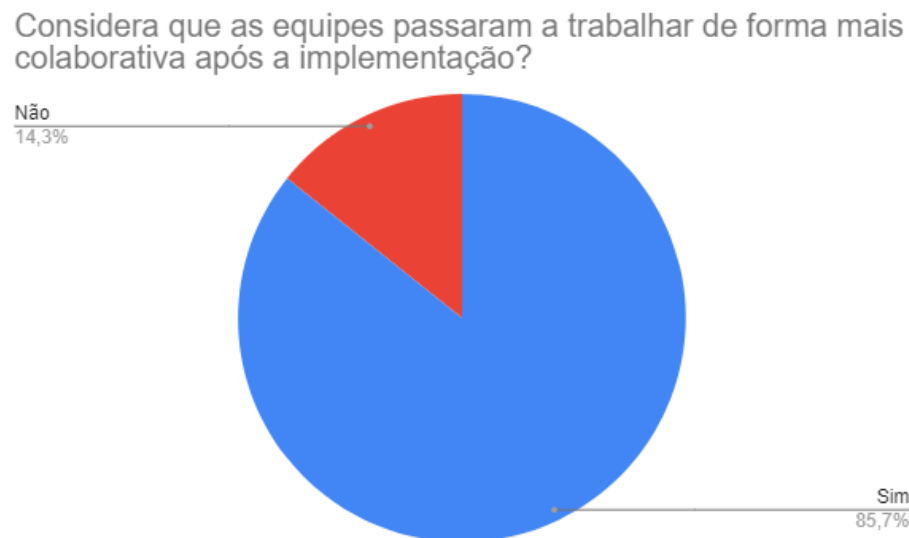


Figura 3.13 As equipes passaram a trabalhar de forma mais colaborativa, após a implementação?

Um membro da equipe de implementação refere que:

“A implementação do *DevOps* é apenas o primeiro passo para promover um ambiente de trabalho mais colaborativo. É necessário investir nas pessoas, fornecer treinamento adequado e criar uma cultura organizacional que valorize a colaboração e o compartilhamento de conhecimento. E só assim, será possível promover uma cultura de colaboração e melhorar a eficiência e a qualidade dos processos de desenvolvimento de *software*.” (ID 6)

5.6.3 Correções de emergência em produção

A Figura 3.14 apresenta a frequência das correções de emergência de sistemas em produção, em que 1 significa “menos frequente” e 10 “mais frequente”. A Figura 3.15 mostra que, 60% dos entrevistados consideram que as correções de emergência de sistemas em produção diminuíram com a adoção do *DevOps* e 40% consideram que as correções em produção passaram a ser mais frequentes.

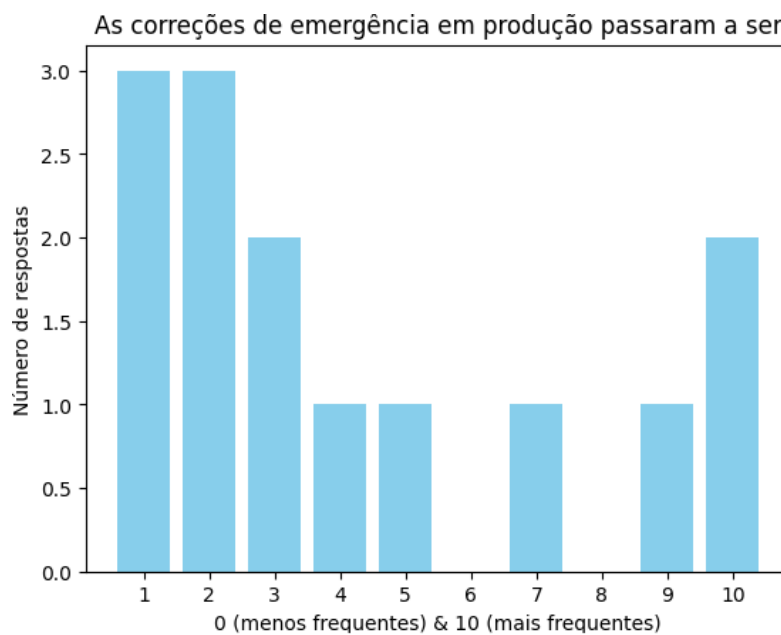


Figura 3.14 Frequência das correções de emergência em produção

5.6.4 Satisfação do cliente e equipes

Os gráficos seguintes (figuras 3.15 a 3.17) apresentam a satisfação do cliente e das equipes, relativamente à adoção da cultura *DevOps*, tendo sido utilizada uma escala em que 1 significa “diminuiu” e 10 “aumentou”.

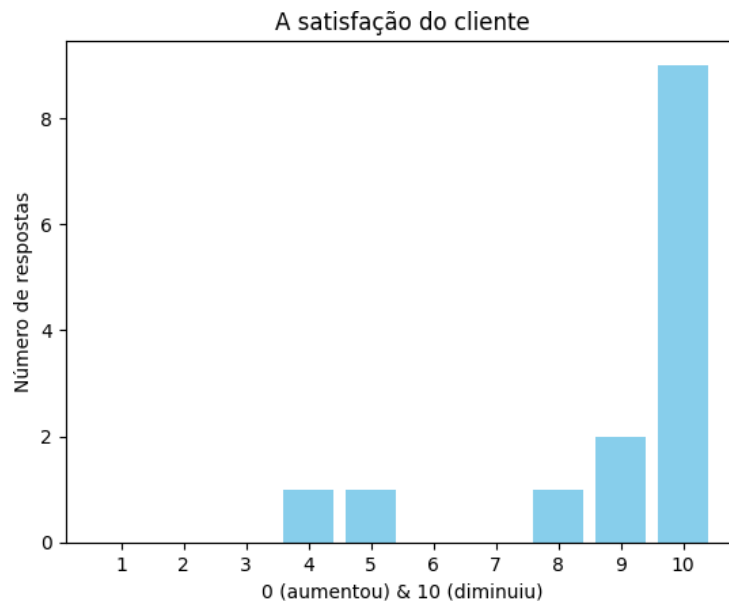


Figura 3.15 Satisfação do cliente

Conforme os entrevistados, observa-se na Figura 3.15, que 85% acreditam que a satisfação do cliente aumentou.

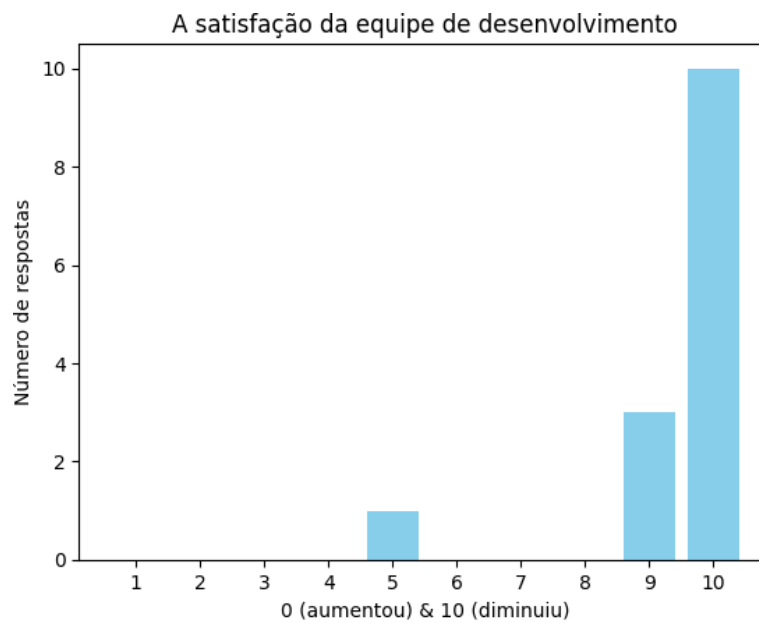


Figura 3.16 Satisfação da equipe de desenvolvimento

Conforme os entrevistados, observa-se na Figura 3.16, que 92% acreditam que a satisfação da equipe de desenvolvimento aumentou.

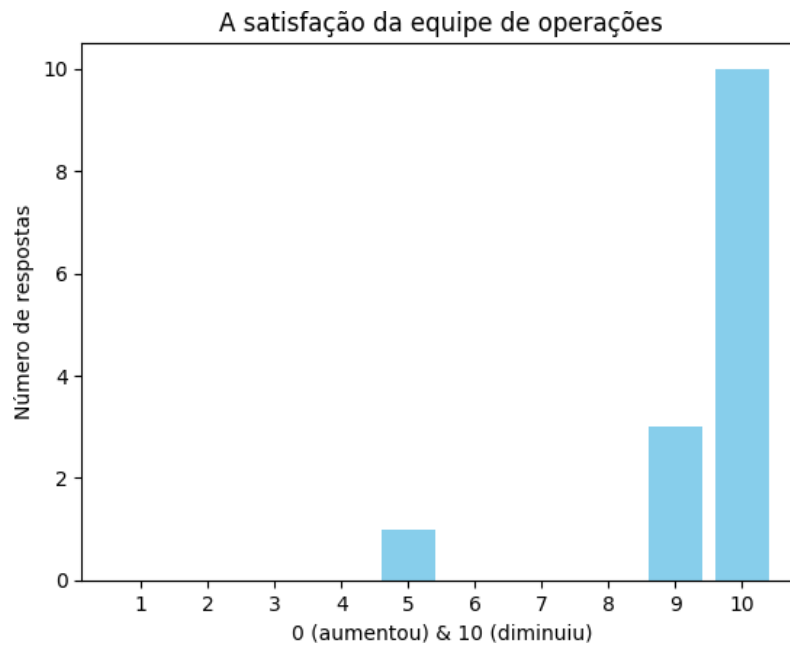


Figura 3.17 Satisfação da equipe de operações

Conforme os entrevistados, observa-se na Figura 3.17, que 92% acreditam que a satisfação da equipe de operações aumentou.

5.6.5 Satisfação da organização

Para aferir a satisfação da organização, também foi utilizada uma escala em que 1 significa “diminui” e 10 “aumentou”.



Figura 3.18 Satisfação da organização

Conforme os entrevistados, observa-se na Figura 3.18, observa-se que há um pensamento majoritariamente positivo a respeito da satisfação da organização.

5.6.6 Frequência de entregas

A frequência das entregas, avaliada também numa escala em que 1 significa “diminuiu” e 10 “aumentou”, também aumentou, mas de forma não muito expressiva, com a maioria dos entrevistados a considerar que aumentaram.

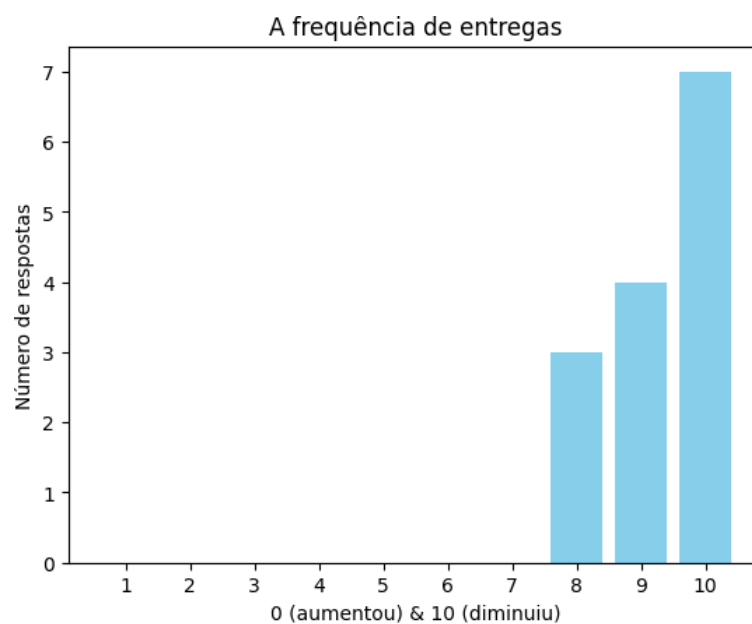


Figura 3.19 Frequência das entregas

Os entrevistados, majoritariamente, acreditam que a frequência das entregas aumentou.

5.6.7 Automação

De forma unívoca, os participantes referem atentar que a automação aumentou (ver Figura 3.20). Este é um resultado esperado, pois com a implementação das práticas do *DevOps* é necessária sempre esperada a automação de processos.

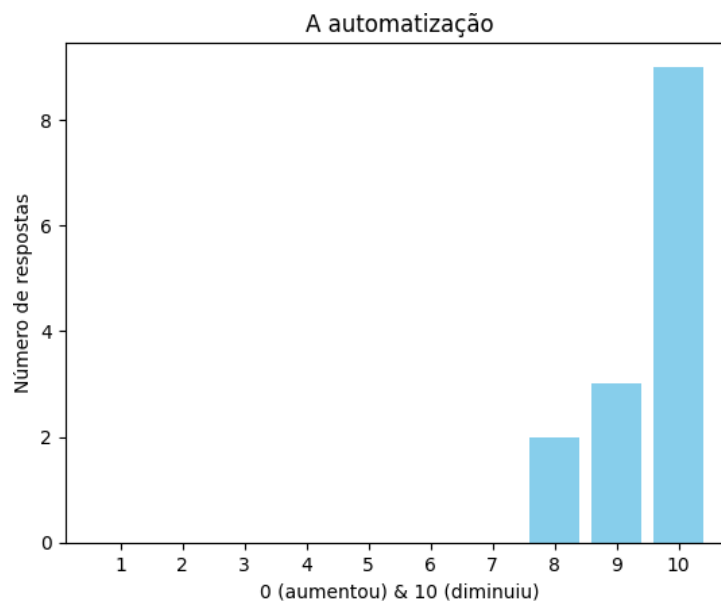


Figura 3.20 Automatização

Os entrevistados, majoritariamente, acreditam que a automatização das suas tarefas repetitivas aumentou, diminuindo o tempo de atuação nessas repetições manuais e utilizando esse tempo salvo em prioridades mais interessantes para a empresa.

5.7 Gestão da mudança

Na sétima seção, são apresentados os resultados relativos à gestão da mudança da organização, como esta ocorreu tendo em conta os vários processos inerentes a ela, incluindo a formação das equipas, a alteração das infraestruturas e a reorganização dos departamentos.

5.7.1 Mudança para a cultura *DevOps* como stakeholder

A mudança para a cultura *DevOps* verificou-se relevante, como mencionam os participantes. 85,7% afirmam que se revelou crucial e 14,3% desadequada (ver Figura 3.21).

Como stakeholder, você considera que a mudança foi:

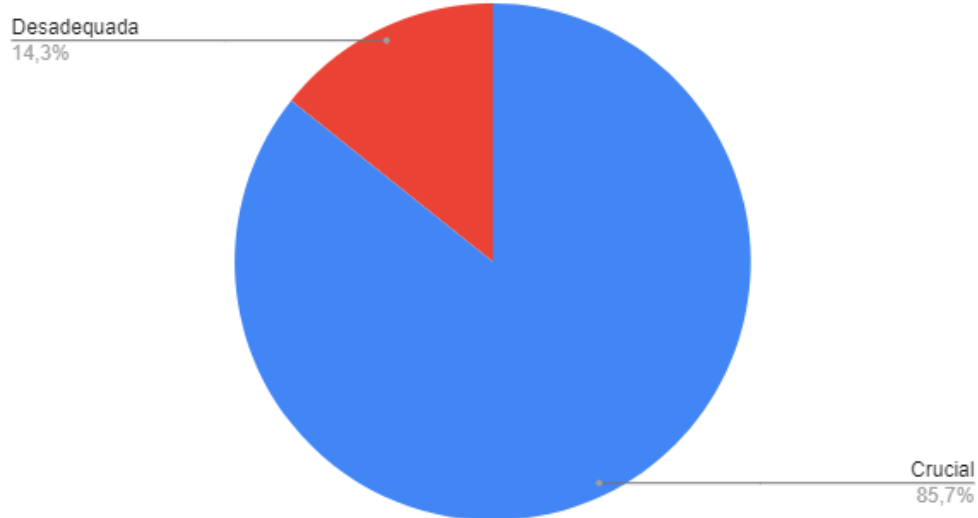


Figura 3.21 Importância da mudança para a cultura *DevOps* na perspectiva de stakeholder

“Se tratando de comunicação e entrosamento, o time não estaria unido e buscando aperfeiçoar os produtos da forma correta. E em relação a custo operacional, não teria tido economias superiores a 80% em relação ao que estava sendo gasto” (ID 10)

Comprova-se com o comentário que a implementação era desejada pela equipe de engenharia.

5.7.2 Formação para a equipe responsável pela implementação do *DevOps*

A formação (de conhecimento) para a implementação do *DevOps* é essencial para que seja possível entregar valor para a implementação a realizar. Neste sentido, a organização optou por um modelo *self-learning*. Este modelo, apesar de não ser considerado formação oficial, que garante certificação, é o melhor numa perspectiva de ir aprendendo à medida que se vai necessitando e avançando no projeto.

Todos eram responsáveis por algo que tinham mais afinidade, no fim, todos participaram de alguma etapa da implementação.

5.7.3 Formação das equipes envolvidas

Não existiu nenhuma formação (de conhecimento) formal das equipes envolvidas em novos processos e ferramentas implementadas, contudo, foi crucial reunir com as equipes com vista à explicação desta nova metodologia de trabalho. Como refere um dos entrevistados:

“A equipe implementadora foi responsável por explicar para todas as outras equipes o que estava acontecendo, explicar sobre os novos processos e ferramentas e documentar tudo.”
(ID 11)

5.7.4 Alterações na infraestrutura da empresa

Para a implementação do *DevOps*, a infraestrutura da empresa não sofreu alterações físicas, apenas virtuais, como foi a reorganização dos serviços e microserviços dentro dos *clusters Kubernetes*.

5.7.5 Alterações estruturais ou reorganização de departamentos da empresa

Não existiu reorganização dos departamentos da empresa, porém, foram criadas as equipes do *DevOps*: *SRE* e *Cloud*, que permitiu endereçar os novos desafios para o desenvolvimento do novo *SDLC*.

A equipe de *SRE* e *Cloud* tem a responsabilidade de garantir a confiabilidade, escalabilidade e eficiência dos sistemas e infraestrutura da organização, por meio do monitoramento, automação, otimização e colaboração com outras equipes. Seu objetivo é oferecer serviços confiáveis e de alto desempenho para os usuários e clientes da organização.

5.7.6 Gestão da mudança

A mudança de processos e ferramentas deve ser gradual e adaptativa, para que os intervenientes possam integrar o processo e compartilhar opiniões para melhorá-lo. Deve

começar-se sempre por um modelo *MVP* (*Produto Mínimo Viável*), para provar que o conceito funciona e que é possível retirar benefícios dos novos processos e ferramentas, como refere um dos entrevistados:

“A mudança foi gerida migrando as várias aplicações/equipes em pequena escala, de um por um. Quando se iniciou a implementação, começou-se com uma versão mais "básica" do *DevOps*, que foi aumentando de complexidade à medida que o processo estava mais maduro e ia sendo adotado por mais equipes.” (ID 7)

5.8 Barreiras *DevOps*

Nesta seção, são apresentados os resultados relativos as barreiras da adoção da cultura *DevOps*, tendo sido estudadas questões como, a identificação dos principais obstáculos, a resistência à mudança para o cenário planeado, novo contexto no que toca à interação entre equipes, e a influência do tamanho da organização.

5.8.1 Principais obstáculos à adoção do *DevOps*

Para avaliar os principais obstáculos à adoção do *DevOps* foram colocadas duas questões, a primeira fechada, para avaliar os obstáculos identificados na revisão de literatura, e a segunda, aberta, em que se pedia aos entrevistados que caracterizassem os principais obstáculos à adoção do *DevOps*, de acordo com a sua experiência.

Relativamente à primeira questão, os entrevistados salientam os obstáculos que têm maior impacto para o sucesso do projeto são os referidos na Figura 3.22.

Principais obstáculos à adoção do DevOps

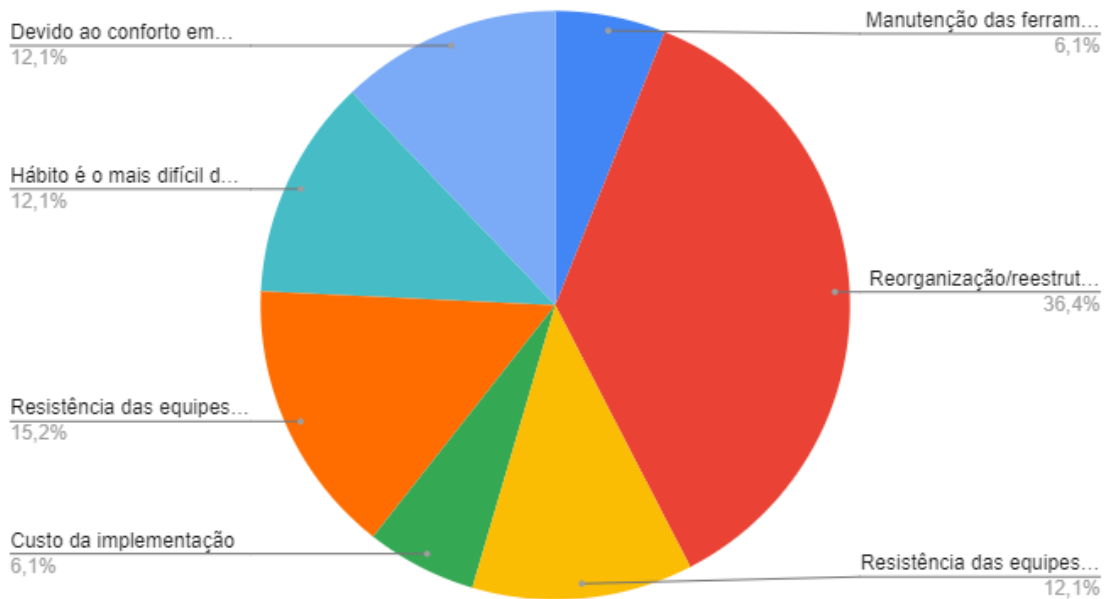


Figura 3.22 Principais obstáculos à adoção do *DevOps*

1. Reorganização/reestruturação dos fluxos internos de desenvolvimento e planejamento (36,4%)
2. Resistência das equipes de desenvolvimento (15,2%)
3. Resistência das equipes de operações (12,1%)
4. Devido ao conforto em que se encontra cada indivíduo (12,1%)
5. Hábito é o mais difícil de se mudar (12,1%)
6. Manutenção das ferramentas (6,1%)
7. Custo da implementação (6,1%)

A reorganização/reestruturação dos fluxos internos de desenvolvimento e planejamento foi considerada pela maioria dos entrevistados como a maior barreira, como refere um dos entrevistados da equipe de implementação:

“A resistência à mudança é normalmente é onde se prende a maioria das empresas, o medo do desconhecido, continua a assombrar as organizações no que toca à adoção de novas metodologias, processos e ferramentas.” (ID 4)

Na segunda pergunta, foi perguntado o seguinte: Quais considera serem os principais obstáculos à adoção do *DevOps*?

As respostas obtidas foram:

- Manutenção das ferramentas;
- Mudança de cultura das pessoas, pois não querem sair da zona de conforto;
- Pessoas sem conhecimento;
- Mentalidade e hábitos;
- Custo humano e financeiro.
- Comumente o ser humano busca poupar energia, e a mudança cultural e de mentalidade, assim como hábitos, são os mais complicados de transformar.

5.8.2 Resistência à mudança

A mudança acarreta sempre a possibilidade de existir resistência. Neste sentido, 57,1% dos entrevistados referem que não existiu resistência à mudança, enquanto 28,6% considera que sim. 14,3% considera que houve ao menos algum tipo de resistência (ver Figura 3.25). Esta divisão de opiniões está relacionada com o envolvimento dos intervenientes na adoção da cultura. Existiram pessoas mais ligadas apenas à implementação de novos processos e que não tiveram necessidade de interagir com as restantes equipes.

Você considera que existiu resistência à mudança por parte dos envolvidos no projeto?

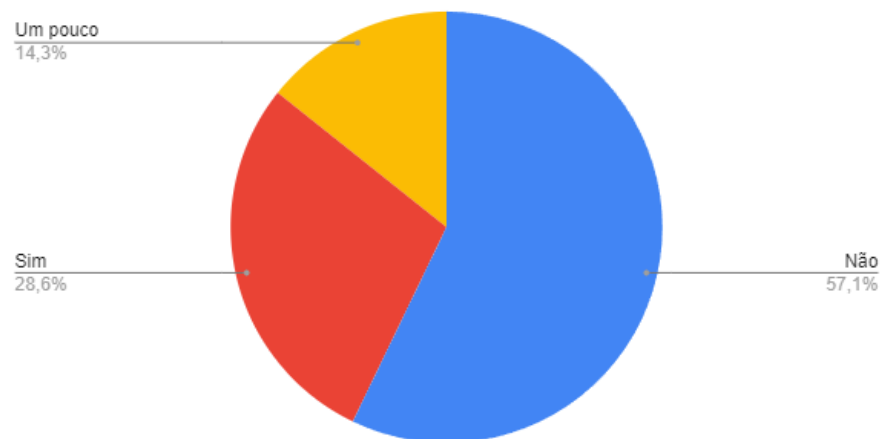


Figura 3.23 Resistência à mudança

5.8.3 Melhoria da comunicação entre as equipes com a implementação do *DevOps*

Existiu uma clara melhoria de comunicação entre as equipes, como refere a maioria dos entrevistados, 92,9% (ver Figura 3.24). A melhoria da comunicação conduz à consequente quebra de barreiras impostas pelos processos antigos.

A interação entre as equipes envolvidas no processo de implementação do *DevOps*, levou à melhoria da comunicação e consequente quebra de silos existentes entre elas?

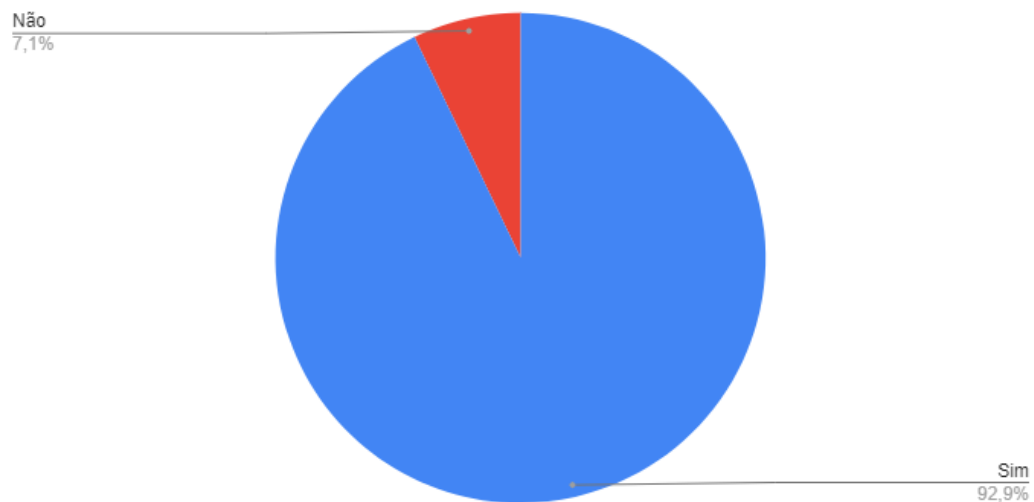


Figura 3.24 Melhoria da comunicação entre equipes com a implementação do *DevOps*

Com a maioria dos entrevistados afirmando que a comunicação das equipes melhorou após a implementação do *DevOps*, a defasagem causada por silos de comunicação é bem mitigada.

5.8.4 Dimensão da organização como barreira

Na Figura 3.25 é apresentado o gráfico que representa a dimensão da organização como barreira da adoção da cultura, sendo 1 pouco relevante e 10 muito relevante.



Figura 3.25 Dimensão da organização como barreira

A implementação do *DevOps* teve por base algumas iniciativas que ocorreram em algumas equipes de desenvolvimento, relacionadas com *CI*, como refere um membro da equipe de implementação:

“Já existia um processo de *CI* implementado na organização usado por quase todas as equipes envolvidas, e que facilitou a evolução para processo *DevOps*. Houve também muito apoio da gestão de topo para a implementação do processo.” (ID 6)

Contudo, esse processo foi melhorado e normalizado por toda a organização, para recolher os benefícios inerentes a esta prática. O apoio da gestão de topo também foi crucial para a implementação, como refere este membro. Desta forma considera que, neste caso, a dimensão não foi o mais relevante.

5.9 Pós-Implementação do *DevOps*

Na última seção apresenta-se o sucesso da adoção da cultura, avaliando se os objetivos foram alcançados, a satisfação dos *stakeholders*, bem como a existência ou não de processos formais de avaliação do sucesso.

5.9.1 Cumprimento dos objetivos

Na Figura 3.26 é possível verificar que a maioria dos entrevistados considerou que o *DevOps* cumpriu os objetivos estipulados.



Figura 3.26 Cumprimento dos objetivos

Não há um consenso sobre o sucesso dos objetivos estipulados, porém é notório que todos os objetivos propostos e alcançados com a adoção da cultura *DevOps*, têm-se revelado satisfatórios, contudo, este é um processo em crescimento e em constante desenvolvimento, muito devido ao tamanho da organização.

5.9.2 Satisfação como *stakeholder*

A Figura 3.27 indica que 92,9% dos entrevistados revelam-se satisfeitos com os objetivos atingidos.

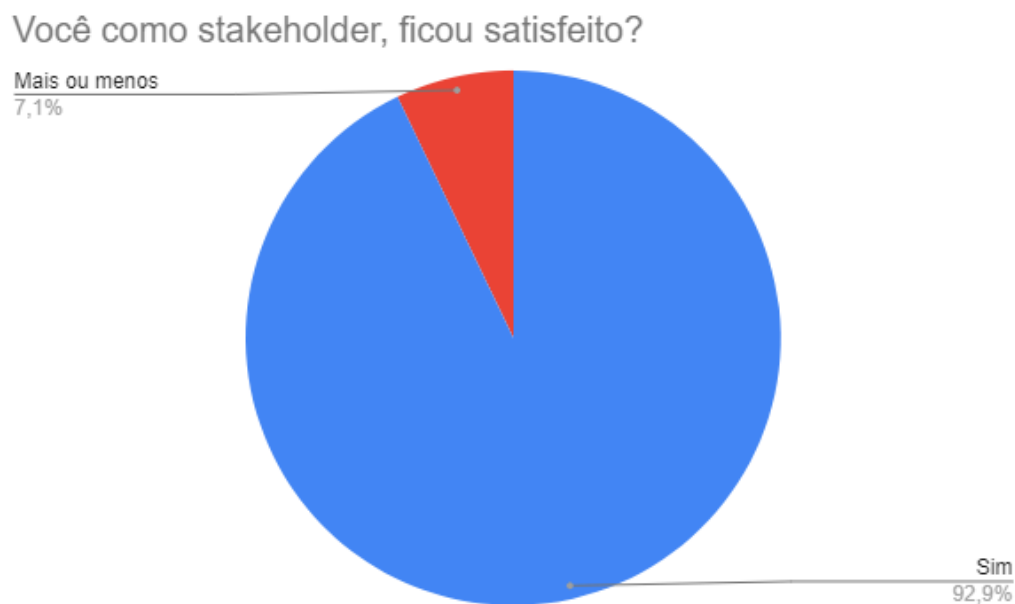


Figura 3.27 Satisfação como stakeholder

Contudo, os restantes 7,1% referem que ainda existe algum caminho a percorrer, sendo sempre um processo evolutivo, referindo:

“Satisfaz, tem de ser visto como um processo em crescimento, e não um processo já acabado. Ainda tem muita margem para evoluir.” (ID 6)

5.9.3 Critérios para avaliação do sucesso da implementação do *DevOps*

Segundo os entrevistados, os melhores critérios para avaliar o sucesso de uma implementação *DevOps* são os seguintes apresentadas na Figura 3.28.

Com base em quais critérios se pode/deve avaliar o sucesso da implementação do DevOps?

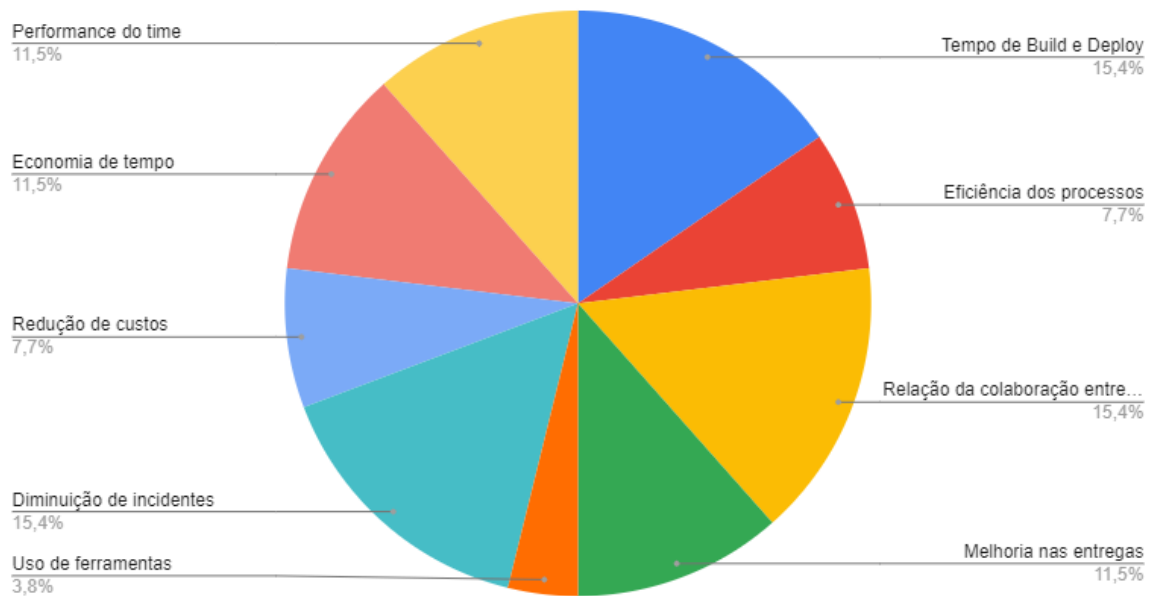


Figura 3.28 Critérios de sucesso de implementação do *DevOps*

Dentre os itens listados como critérios avaliadores do sucesso para implementação do *DevOps*, encontram-se:

- Tempo de *Build* e *Deploy* (15,4%);
- Relação da colaboração entre as equipes (15,4%);
- Diminuição de incidentes (15,4%);
- Economia de tempo (11,5%);
- Performance do time (11,5%);
- Melhoria nas entregas (11,5%);
- Eficiência dos processos (7,7%);
- Redução de custos (7,7%);
- Uso de ferramentas (3,8%).

5.9.4 Avaliação formal do sucesso da implementação do *DevOps*

Na organização não existiu nenhum momento de avaliação formal para avaliar o sucesso da implementação do processo. No entanto, ao longo da implementação foram feitas reuniões para validar se o progresso era benéfico para a organização.

5.9.5 Aplicação do *DevOps*

A Figura 3.31 apresenta as ideologias de implementação do *DevOps* na organização.

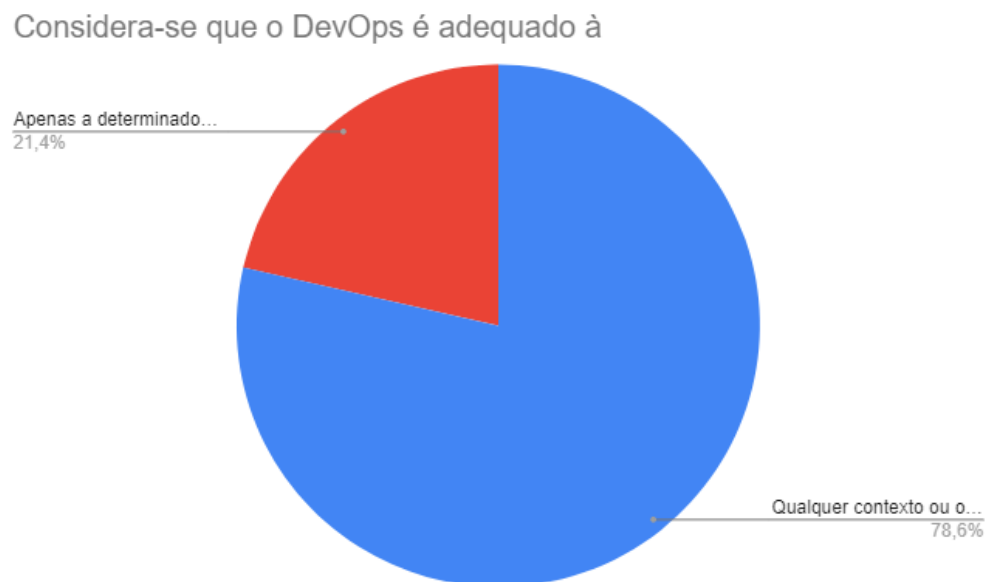


Figura 3.29 Aplicação do *DevOps*

78,6% dos entrevistados afirmam que a cultura *DevOps* é aplicável a qualquer contexto ou a qualquer organização. Os benefícios que podem ser retirados desta cultura são inúmeros e podem ser aplicadas alguns princípios e práticas. Um dos membros da equipe de *Cloud* afirma o seguinte:

“Os fundamentos de *DevOps* possibilitam resultados mais rápidos para alcançar os objetivos.”
(ID 10)

5.9.6 Sucesso da adoção do *DevOps* na organização

A maioria dos entrevistados considera que houve sucesso na adoção da cultura, como mostra a Figura 3.30.

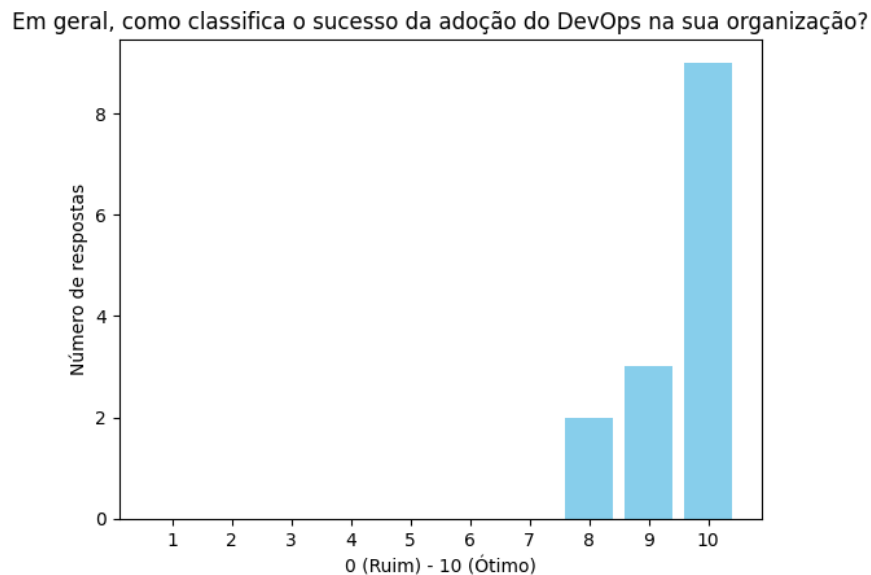


Figura 3.30 Sucesso da adoção do *DevOps* na organização

“Este sucesso começa cada vez mais a contagiar toda a organização e estrutura organizacional, no qual todos querem abraçar esta ‘aventura’. Foi constatado uma maior previsibilidade em todo o ciclo de vida de *software*”. (ID 10)

Esta mudança tem-se mostrado cada vez mais vantajosa para toda a organização.

5.10 Considerações finais

Por meio deste estudo de caso, foi possível examinar a situação da organização antes e depois da adoção da cultura *DevOps*. Além disso, compreendemos as motivações e objetivos da mudança, tanto na fase anterior, como na fase posterior à implementação do *DevOps*. Analisamos o processo adotado nas diferentes etapas de adoção, as práticas e princípios implementados, bem como os processos e ferramentas que foram reutilizados nesse processo. Também identificamos as áreas que passaram por mudanças significativas, tanto em termos organizacionais, quanto de infraestrutura.

Através da análise das entrevistas, pudemos avaliar se a organização compartilha dos mesmos princípios, benefícios e desafios encontrados na revisão da literatura sobre a adoção do *DevOps*.

5.10.1 Adoção do *DevOps*

Como mencionado anteriormente, não há um guia ou caminho definitivo para a adoção do *DevOps*. A implementação deve envolver a análise dos processos que precisam ser alterados e das ferramentas a serem adotadas. Nesse caso, a organização já tinha uma mentalidade aberta em relação à adoção de novas tecnologias, o que facilitou uma adoção mais rápida, já que algumas ferramentas relacionadas ao *DevOps* já estavam sendo utilizadas.

A adoção do *DevOps* pode ocorrer a partir de duas perspectivas: *top-down* ou *bottom-up*. Não há uma abordagem correta, ou melhor, ambas são válidas. É um processo que deve envolver todos os envolvidos no desenvolvimento de *software*, e é importante haver alinhamento entre eles sobre o que será implementado.

Os principais objetivos foram alcançados, mas como se trata de um processo de melhoria contínua, ainda há um caminho a percorrer. No entanto, a evolução pode ser limitada pela dimensão da organização. Os entrevistados consideraram a adoção relevante e positiva, destacando os benefícios obtidos.

Embora não tenha sido realizada uma avaliação formal, os entrevistados mencionaram que a adoção do *DevOps* pode ser avaliada com base em critérios como mencionado na seção 5.9.3. Com base nas respostas das questões das seções 5.6 (benefícios do *DevOps*) e 5.9 (pós-implementação), como aumento da frequência de entregas, redução de correções em produção e alta satisfação da organização e das partes interessadas, conclui-se que a adoção foi bem-sucedida.

5.10.2 Benefícios do *DevOps*

Com base na comparação dos benefícios identificados na Tabela 1.2 com os mencionados pelos entrevistados, pode-se concluir que as opiniões destes estão alinhadas com a revisão da literatura. Além dos benefícios mencionados na Tabela 1.2, os entrevistados identificaram outros benefícios, como o aumento da fiabilidade do processo e a unificação deles.

Os entrevistados foram solicitados a escolher os benefícios que consideravam mais relevantes dentre os identificados na revisão da literatura. A melhoria da qualidade do *software*, dos tempos de entrega, do desempenho e produtividade organizacional, da colaboração e da velocidade de resolução de problemas foram os mais selecionados. De modo geral, os entrevistados confirmaram o aumento da comunicação entre as equipes, a melhoria dos tempos de entrega e resolução de problemas, bem como a satisfação do cliente, das equipes de desenvolvimento, operações e relacionadas ao *SDLC*. Também houve um aumento na frequência das entregas, o que reforça o benefício mencionado anteriormente. Globalmente, a satisfação da organização também aumentou consideravelmente, mesmo que já houvesse um contentamento identificado com o estado atual dos processos internos, antes da implementação.

A implementação das práticas de Integração Contínua (*CI*) e Entrega Contínua (*CD*), adotadas pela organização, reforça alguns desses benefícios. Essas práticas visam reduzir erros humanos e, conseqüentemente, contribuem para a resolução de problemas.

Esses benefícios são obtidos por meio das práticas implementadas pela organização. O pensamento e o princípio de melhoria contínua são fundamentais na cultura organizacional após a adoção, garantindo o sucesso contínuo da organização e uma maior participação de todas as equipes relacionadas ao desenvolvimento de *software* no processo de tomada de decisões.

5.10.3 Barreiras à adoção do *DevOps*

Existem várias barreiras que podem dificultar ou atrasar a adoção do *DevOps*, como mencionado anteriormente. A resistência à mudança foi identificada na revisão da literatura

como uma dessas barreiras. Essa barreira também foi destacada pelos entrevistados, sendo que metade deles considerou que houve resistência à mudança por parte das equipes. Com a introdução de novos processos, é essencial capacitar as equipes com o conhecimento necessário para lidar com as ferramentas e compreender os procedimentos.

A falta de comunicação é outra barreira significativa. Antes da adoção do *DevOps*, as equipes trabalham geralmente de forma isolada. No entanto, conforme relatado pelos entrevistados, a adoção do *DevOps* promoveu uma maior comunicação e colaboração entre as equipes, superando essa barreira. Isso também contribui para superar a barreira do negócio, que é mais desafiadora, devido à dimensão da organização. Uma abordagem para superar essa barreira é a criação de equipes multidisciplinares que possam entregar valor de forma mais rápida e contínua. Embora essa barreira ainda persista, a organização está implementando iniciativas nesse sentido.

As barreiras técnicas podem ser superadas por meio da capacitação da equipe de implementação e da formação das equipes envolvidas. A equipe de implementação aprendeu de forma autodidata e criou ambientes de teste para experimentar o que foi aprendido, o que permitiu identificar o caminho a seguir e as ferramentas a serem adotadas. No entanto, a falta de formação certificada pode prolongar o tempo de implementação e resultar em subutilização das ferramentas. Quanto à formação das equipes envolvidas, ela foi explicativa em relação ao uso das ferramentas e processos definidos, sendo considerada suficiente para o propósito.

A organização conseguiu superar a maioria das barreiras antes mesmo de adotar a cultura *DevOps*, graças a uma visão estratégica e decisões acertadas que permitiram a adoção adequada das tecnologias necessárias para habilitar as práticas do *DevOps*.

6. CONCLUSÃO

Este projeto teve como objetivo analisar a adoção do *DevOps* em uma organização, abordando princípios, práticas, benefícios e barreiras envolvidos. Um estudo de caso foi conduzido em uma grande empresa que estava passando por uma migração para a cultura *DevOps*. O estudo permitiu avaliar a situação "antes" e "depois" da adoção do *DevOps* e explorar suas diversas facetas.

Antes da adoção, a organização seguia um processo de desenvolvimento em cascata, com um ciclo de vida bem definido e diferentes responsáveis em cada etapa, desde o desenvolvimento até a produção, passando por testes e ambientes de pré-produção. A mudança para o *DevOps* envolveu a transição para práticas ágeis de desenvolvimento, incorporando os processos de forma mais integrada e automatizada. A adoção de práticas de (CI/CD) permitiu a execução mais eficiente dos processos, com menor intervenção humana e redução de falhas. Alguns princípios foram estabelecidos e bem implementados para facilitar a adoção dessas práticas.

Os resultados do estudo de caso destacaram a importância da (CI/CD) como práticas relevantes, bem como benefícios como a melhoria da qualidade do *software* e barreiras como a resistência à mudança. O apoio da alta gestão foi identificado como um fator-chave para o sucesso na adoção do *DevOps*, assim como em outras iniciativas organizacionais. No entanto, houve limitações, como a dificuldade em agendar e conduzir entrevistas devido à disponibilidade de tempo dos participantes.

Em estudos futuros, pretende-se repetir esse estudo de caso em outras organizações para comparar os resultados e obter uma compreensão mais detalhada. Conclui-se que não há uma definição única para o *DevOps*, pois cada organização adota a cultura de acordo com suas necessidades e práticas favoráveis ao seu setor de atividade. O *DevOps* visa alcançar lançamentos mais frequentes, automação robusta e *software* confiável, acelerando o alcance dos objetivos da organização.

REFERÊNCIAS

- A. Ravichandran et al. (2016). Practical *DevOps*. *DevOps* for Digital Leaders. <https://doi.org/10.1007/978-1-4842-1842-6>
- Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A software Architect's Perspective*. <https://doi.org/10.1017/CBO9781107415324.004>
- Broadcom. (2019). CA Harvest *software* Change Manager.
- Cois, C. A., Yankel, J., & Connell, A. (2015). Modern *DevOps*: Optimizing *software* development through effective system interactions. IEEE International Professional Communication Conference, 2015-Janua. <https://doi.org/10.1109/IPCC.2014.7020388>
- Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal*, 8(4), 273–289. <https://doi.org/10.1046/j.1365-2575.1998.00040.x>
- Debois, P. (2008). Agile infrastructure and operations: how infra-gile are you? In *Agile 2008 Conference* (pp. 202–207). IEEE.
- Docker. (2019). What is a Container? | Docker. Retrieved June 21, 2019, from <https://www.docker.com/resources/what-container>
- Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of *DevOps* usage in practice. *Journal of software: Evolution and Process*, 29(6), e1885. <https://doi.org/10.1002/smr.1885>
- Fallis, A. G. (2013). Effective *DevOps*. *Effective DevOps* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Forsgren, N., Humble, J., Kim, G., Brown, A., & Kersten, N. (2017). 2017 State of *DevOps* Report. Retrieved from <http://www.berrykersten.nl/wpcontent/uploads/2017-state-of-DevOps-report.pdf>
- Hamunen, J. (2016). Challenges in Adopting a *DevOps* Approach to *software* Development and Operations. Aalto University. Retrieved from <http://urn.fi/URN:NBN:fi:aalto-201609083476>

Huckabee, W. A. (2015). Requirements Engineering in an Agile *software* Development Environment. *Defense ARJ*, 22(4), 394–415. Retrieved from https://www.researchgate.net/profile/Allen_Huckabee/publication/281850655_Defense_Acquisition_Research_Journal/links/55fb466008ae07629e07bbd8.pdf

Humble, J., & Farley, D. (2010). *Continuous Delivery*. Pearson Education.

Humble, J., & Molesky, J. (2011). Why enterprises must adopt *DevOps* to enable continuous deliver. *Cutter IT Journal*, 24(8).

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook : How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. The *DevOps Handbook*. <https://doi.org/2016951904>

Kornilova, I. (2017). *DevOps* is a culture, not a role! Retrieved June 24, 2019, from <https://medium.com/@neonrocket/DevOps-is-a-culture-not-a-role-be1bed149b0>

Krafcik, J. F. (1988). Triumph of the lean production system. *MIT Sloan Management Review*, 30(1), 41.

Laihonen, P. (2018). *Adoption of DevOps Practices in the Finnish software Industry: an Empirical Study*. Aalto University.

Luz, W. P. (2018). Uma Caracterização da Adoção do *DevOps* Utilizando Grounded Theory. Universidade de Brasília. Retrieved from http://repositorio.unb.br/bitstream/10482/33950/1/2018_WelderPinheiroLuz.pdf

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). An exploratory study of *DevOps* extending the dimensions of *DevOps* with practices. In *The Eleventh International Conference on software Engineering Advances (ICSEA 2016)* (pp. 91–99). Rome, Italy.

Nuottila, J., Aaltonen, K., and Kujala, J. (2016). Challenges of adopting agile methods in a public organization. *International Journal of Information Systems and Project Management*, 4(3), 65–85. <https://doi.org/10.12821/ijispm040304>

Riley, C. (2014). How to Keep CALMS and Release More! Retrieved from <https://blog.rapid7.com/2014/10/24/how-to-keep-calms-and-release-more/>

Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). *DevOps* Adoption Benefits and Challenges in Practice: A Case Study. In *Lecture*

Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10027 LNCS, pp. 590–597). https://doi.org/10.1007/978-3-319-49094-6_44

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in *software* engineering. *Empirical software Engineering*, 14(2), 131–164. <https://doi.org/10.1007/s10664-008-9102-8>

Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5(Ci), 3909–3943. <https://doi.org/10.1109/ACCESS.2017.2685629>

STH. (2019). Agile vs. Waterfall: Which is the Best Methodology for Your Project? Retrieved June 2, 2019, from <https://www.softwaretestinghelp.com/agile-vs-waterfall/>

Trigo, A., Varajão, J., Algoritmi, C., Molina-Castillo, F. J., Gonzalvez-Gallego, N., SotoAcosta, P., & Barroso, J. (2010). IT professionals: An iberian snapshot. *International Journal of Human Capital and Information Technology Professionals*, 1(1). <https://doi.org/10.4018/jhcitp.2010091105>

Varajão, J. (2018). The many facets of information systems (+projects) success. *International Journal of Information Systems and Project Management*, 6(4), 5–13.

Virmani, M. (2015). Understanding *DevOps* & bridging the gap from continuous integration to continuous delivery. 5th International Conference on Innovative Computing Technology, INTECH 2015, (Intech), 78–82. <https://doi.org/10.1109/INTECH.2015.7173368>

Visser, J., Rigal, S., Wijnholds, G., & Lubsen, Z. (2016). Building *software* Teams: Ten Best Practices for Effective *software* Development. (O'REILLY, Ed.). O'Reilly Media.

Walls, M. (2013). Building a *DevOps* Culture. O'Reilly.

Wills, J. (2010). What *DevOps* Means to Me. Retrieved June 24, 2019, from <https://blog.chef.io/2010/07/16/what-DevOps-means-to-me/>

Yin, R. K. (1989). Case study research. California: Sage publications.

Yin, R. K. (2009). Case Study Research: Design and Methods (4th ed.). Sage Publications.

APÊNDICE

Na seção de apêndice, serão listadas as perguntas feitas no questionário realizado. As perguntas estão separadas por sessões, conforme o apresentado no Capítulo 5, em: **Introdução, Ciclo *DevOps*, Pré-*DevOps*, Implementação do *DevOps*, Práticas *DevOps*, Técnicas e Ferramentas *DevOps*, Benefícios do *DevOps*, Barreiras *DevOps* e Pós-implementação do *DevOps*.**

Introdução

1. Em que consiste o *DevOps*? (Descreva sucintamente o que entende por *DevOps*).
2. Qual considera ser a melhor abordagem (a nível organizacional) para a adoção de *DevOps*? (Top-Down: Iniciativa partindo da administração; Bottom-Up: Iniciativa partindo da equipe implementadora).

Ciclo *DevOps*

3. Das fases listadas na imagem acima, quais considera serem as que apresentam maior risco na implementação do *DevOps*? (Nesse contexto, como maior risco, subentende-se que as outras fases possam influenciar diretamente umas nas outras).
4. Considerando as práticas *DevOps* abaixo, relacionadas com o ciclo de desenvolvimento de *software*, quais considera serem as mais impactantes (positivamente) para a implementação do *DevOps*? (Planejamento; Desenvolvimento colaborativo; Integração contínua (CI); Testes automatizados/contínuos; Entrega contínua; Implantação contínua (CD); Monitorização contínua; Feedback contínua do cliente e otimização).

Pré-*DevOps*

5. Antes da implementação do *DevOps*, quais eram as ferramentas utilizadas na sua empresa para gerir o ciclo de vida de desenvolvimento de *software* (SDLC)?
6. Já existia, antes da implementação, algum processo da organização semelhante a um processo inerente à Cultura *DevOps*? (Planejamento; Desenvolvimento colaborativo;

Integração contínua (CI); Testes automatizados/contínuos; Entrega contínua; Implantação contínua (CD); Monitorização contínua; Feedback contínua do cliente e otimização).

Implementação do *DevOps*

7. Na sua organização, foram consideradas outras alternativas à adoção do *DevOps*? (Sim; Não; Se sim, quais?).

8. Foi seguido algum guia de boas práticas de implementação do *DevOps*? (Sim; Não; Se sim, quais?).

9. Houve a necessidade de participação de entidades externas à organização para dar suporte ao processo de implementação? (Sim; Não; Se sim, liste).

10. Qual foi o custo estimado da implementação *DevOps*? (Licenças, formação, alteração de processos, etc).

11. Quais áreas em que se verificou uma maior mudança organizacional/cultural? (Desenvolvimento, operações, infraestrutura, negócio, etc).

12. Considerando a literatura, o processo implementado na sua organização é: (Puramente *DevOps*; Híbrido; *DevOps* apenas no nome).

Práticas *DevOps*

13. A prática de planeamento foi implementada? (Sim; Não).

14. A prática de desenvolvimento colaborativo foi implementada? (Sim; Não).

15. A prática de integração contínua (CI) foi implementada? (Sim; Não).

16. A prática de testes contínuos foi implementada? (Sim; Não).

17. A prática de entrega contínua foi implementada? (Sim; Não).

18. A prática de implantação contínua foi implementada? (Sim; Não).

19. A prática de monitoramento contínuo foi implementada? (Sim; Não).

20. A prática de feedback contínuo do cliente e otimização foi implementada? (Sim; Não).

21. Quais são, na sua opinião, os principais fatores influenciadores do sucesso da implementação do *DevOps*? (Indique ao menos 3 fatores).

22. Na sua organização, verificaram-se todos esses fatores?

Técnicas e Ferramentas *DevOps*

23. Tendo em conta os processos e as ferramentas da organização, como foi implementado o *DevOps*? (Ocorreu através da adaptação dos novos processos às ferramentas existentes; Ocorreu através da adaptação dos processos existentes às novas ferramentas; Ocorreu através da definição de novos processos e novas ferramentas).

24. Quais das seguintes ferramentas foram implementadas na sua organização? (Planejamento e documentação; Sistema de controle de versão; Orquestrador de pipelines; Containerização; Orquestrador de contêineres; Testes Integrados; Orquestrador de releases; Monitoramento).

25. Liste as ferramentas implementadas.

26. Foi considerada alguma plataforma Cloud para suporte da infraestrutura? (AWS; Azure; GCP; Oracle; Outro).

Benefícios do *DevOps*

27. Quais considera serem os principais benefícios da adoção do *DevOps*?

28. Selecione os principais benefícios da adoção do *DevOps* (Melhoria da colaboração entre as equipes; Aumento da velocidade de resolução de problemas; Aumento da automatização; Entrega de valor rápida e contínua; Feedback contínuo; Aumento de produtividade e desempenho).

29. Considera que as equipes passaram a trabalhar de forma mais colaborativa após a implementação? (Sim; Não).

30. As correções de emergência em produção passaram a ser (0-10 mais/menos frequentes).

31. A satisfação do cliente (0-10 diminuiu/aumentou).

32. A satisfação da equipe de desenvolvimento (0-10 diminuiu/aumentou).
33. A satisfação da equipe de operações (0-10 diminuiu/aumentou).
34. A satisfação da sua organização (0-10 diminuiu/aumentou).
35. A frequência de entregas (0-10 diminuiu/aumentou).
36. A automatização (0-10 diminuiu/aumentou).
37. Como stakeholder, você considera que a mudança foi (Crucial; Desadequada; Irrelevante).
38. Houve uma formação para a equipe responsável pela implementação do *DevOps*? (Sim; Não; Se sim, qual?).
39. Houve formação para as equipes envolvidas na mudança (desenvolvimento, operações) focada nos novos processos e ferramentas? (Sim; Não; Se sim, qual?).
40. Houve necessidade de alteração na infraestrutura da organização? (Sim; Não; Se sim, qual?).
41. Houve necessidade de alterações estruturais ou reorganização de departamentos da organização? (Sim; Não; Se sim, qual?).
42. Como foi gerida a transição para o fluxo *DevOps*? O processo revelou-se adequado?

Barreiras *DevOps*

43. Quais considera serem os principais obstáculos à adoção do *DevOps*?
44. Selecione os principais obstáculos à adoção do *DevOps* (Resistência das equipes de desenvolvimento; Resistência das equipes de operações; Manutenção das ferramentas; Custo da implementação; Reorganização/reestruturação dos fluxos internos de desenvolvimento e planejamento).
45. Você considera que existiu resistência à mudança por parte dos envolvidos no projeto? (Sim; Não; Se sim, qual?).
46. A interação entre as equipes envolvidas no processo de implementação do *DevOps*, levou à melhoria da comunicação e consequente quebra de silos existentes entre elas? (Sim; Não).

47. A dimensão da organização foi uma barreira para a adoção do *DevOps*? (0-10 pouco/muito relevante).

Pós-implementação do *DevOps*

48. Os objetivos estipulados com a adoção do *DevOps* foram atingidos? (0-10 foram/não foram atingidos).

49. Você como *stakeholder*, ficou satisfeito? (Sim; Não).

50. Com base em quais critérios se pode/deve avaliar o sucesso da implementação do *DevOps*? (Redução de custos, eficiência, etc).

51. Na sua organização houve uma avaliação formal da implementação do *DevOps*? Se sim, quais critérios foram utilizados?

52. Considera-se que o *DevOps* é adequado à (qualquer contexto ou organização; apenas a determinados contextos; apenas a determinadas organizações).

53. Em geral, como classifica o sucesso da adoção do *DevOps* na sua organização? (0-10 fracasso/sucesso total).