



Instituto Politécnico de Coimbra
Instituto Superior de Contabilidade
e Administração de Coimbra

Leandro Sousa

DevOps – estudo de caso

DevOps – estudo de caso

Leandro Sousa

ISCAC | 2019

Coimbra, outubro de 2019



Instituto Politécnico de Coimbra
Instituto Superior de Contabilidade
e Administração de Coimbra

Leandro Sousa

DevOps – estudo de caso

Trabalho de projeto submetida(o) ao Instituto Superior de Contabilidade e Administração de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de **Mestre em Sistemas de Informação de Gestão**, realizada(o) sob a orientação do Professor Doutor António Trigo e do Professor Doutor João Eduardo Quintela Alves Sousa Varajão.

Coimbra, Outubro de 2019

TERMO DE RESPONSABILIDADE

Declaro ser o autor deste projeto, que constitui um trabalho original e inédito, que nunca foi submetido a outra Instituição de ensino superior para obtenção de um grau acadêmico ou outra habilitação. Atesto ainda que todas as citações estão devidamente identificadas e que tenho consciência de que o plágio constitui uma grave falta de ética, que poderá resultar na anulação do presente projeto.

PENSAMENTO

“If I had asked people what they wanted, they would have said faster horses.” – Henry Ford

DevOps não são apenas processos mais rápidos, é um conjunto de inovação capaz de abranger de todo o ciclo de vida de desenvolvimento de software, através da adaptação da organização às suas necessidades atuais.

“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.” — Charles Darwin

AGRADECIMENTOS

Este projeto no âmbito do mestrado em Sistemas de Informação de Gestão foi uma longa viagem, preenchido de alegrias, tristezas, percalços e cheio de desafios contado com apoios e incentivos essenciais, sem os quais não seria possível a sua realização.

Percorrer este caminho só foi possível com o auxílio de diversas pessoas, às quais estarei eternamente grato.

Ao meu orientador, professor Doutor António Trigo, por ter sempre acreditado em mim, agradeço a orientação marcada pelo rigor científico, pelo interesse permanente e pelo empenho insuperável contribuindo para que este projeto pudesse ser concluído.

Ao meu coorientador, professor Doutor João Eduardo Varajão, por ter sempre acompanhado este projeto e se ter revelado cientificamente exigente, traspondo sempre os seus pensamentos e contributos para o projeto.

À organização em que foi realizado o estudo de caso pela autorização e disponibilidade, sem a qual o projeto não teria sido possível. A todos os entrevistados, pertencentes à organização, um agradecimento especial, pelo acompanhamento e dedicação que depositaram neste projeto.

Aos meus pais Acrísio Sousa e Luísa Sousa, por terem feito parte do modelo de coragem que implementaram em mim e pelo apoio ao longo de todo o meu período académico.

A toda a minha família e amigos queria agradecer sob a forma de pedido de desculpas, pelas ausências, devido aos momentos menos fáceis deste projeto. Por terem sido sempre compreensíveis, pelo companheirismo e força que depositaram em mim.

Um agradecimento especial aos que foram revendo a minha tese através de uma leitura crítica de forma a ser possível aperfeiçoar este projeto.

Sem vocês nada disto seria e possível, e assim dedico-vos este trabalho!

RESUMO

O *DevOps* é frequentemente referido pelos profissionais da área das tecnologias da informação como um movimento, cultural ou profissional, que apresenta uma nova abordagem de entrega de aplicações informáticas (*software*), através da colaboração entre as equipas de desenvolvimento e das equipas de operações. Tem subjacente um conjunto diversificado de princípios, relacionados com cultura, automatização, *lean*, monitorização e partilha; e práticas, tais como, *continuous integration* e *continuous deployment*.

É um fenómeno em expansão, mas cuja adoção ao nível das organizações ainda está numa fase embrionária, necessitando de mais investigação no sentido de clarificar os benefícios, custos e barreiras à adoção do mesmo.

Foi com este objetivo subjacente que se conduziu o trabalho de investigação apresentado neste relatório, um estudo de caso de uma organização de grande dimensão da área das telecomunicações, que está atualmente a passar por um processo de migração para a cultura *DevOps*. Através do caso de estudo foi possível verificar o “antes” e o “depois” da adoção de *DevOps* na organização, e estudar a sua adoção nas diferentes vertentes.

Como principais resultados identificam-se a revisão da literatura efetuada e a caracterização vários aspetos como, por exemplo, as práticas (que os entrevistados identificaram como mais relevantes a *continuous integration* e *continuous deployment*), os benefícios, (ex: melhoria da qualidade de software), barreiras, (ex: resistência à mudança). De destacar, ainda, como principal fator influenciador do sucesso na adoção/implementação de *DevOps*, à semelhança de outras iniciativas nas organizações, o apoio da gestão de topo.

Este trabalho representa um primeiro contributo no estudo do fenómeno *DevOps*, esperando os autores repetir a realização deste estudo noutras organizações.

Palavras-chave: DevOps; Desenvolvimento; Operações; Equipas; Estudo de Caso.

ABSTRACT

DevOps is often referred to by information technology professionals as a cultural or professional movement that presents a new approach to the delivery of computer applications (software), through collaboration between development teams and operations teams. It is underpinned by a diverse set of principles, related to culture, automation, lean, monitoring and sharing; and practices, such as continuous integration and continuous deployment.

It is an expanding phenomenon, but its adoption at the level of organizations is still at an embryonic stage, requiring research to clarify the benefits, costs, and barriers to its adoption.

It was with this objective in mind that the research work presented in this report was conducted, a case study of a large telecommunications organization that is currently undergoing a process of migration to DevOps culture. Through the case study, it was possible to verify the “before” and “after” the adoption of DevOps in the organization and thus study their adoption in the different areas.

The main results were the literature review and the description of different aspects, for example, regarding practices, (in which the participants identified continuous integration and continuous deployment as the most relevant), the benefits, (*ex*: software improvement), the barriers, (*ex*: resistance to change). The main factor influencing the success in the adoption/implementation of DevOps should also be highlighted, as with other initiatives in organizations, the support of top management.

This work represents a first contribution to the study of the DevOps phenomenon and the authors hope to repeat this study in other organizations.

Keywords: DevOps; Development; Operations; Teams; Case Study

ÍNDICE GERAL

INTRODUÇÃO	1
1 Revisão da literatura	3
1.1 Desenvolvimento e Entrega de <i>Software</i>	3
1.2 <i>DevOps</i>	5
1.2.1 Pilares do <i>DevOps</i>	5
1.2.2 Princípios	8
1.2.3 Práticas	9
1.2.4 Ciclo <i>DevOps</i>	10
1.2.5 Abordagem tradicional vs. <i>DevOps</i>	12
1.3 Benefícios do <i>DevOps</i>	13
1.4 Desafios e barreias à adoção	14
2 METODOLOGIA	17
2.1 Estudo de caso	17
2.2 Etapas do estudo de caso	17
2.2.1 Formulação do problema	18
2.2.2 Seleção dos casos	18
2.2.3 Elaboração do protocolo	18
2.2.4 Realizar o estudo de caso	21
2.2.5 Elaboração do relatório de casos cruzados	21
3 ANÁLISE E DISCUSSÃO DOS RESULTADOS	22
3.1 Definição <i>DevOps</i>	22
3.1.1 O que é o <i>DevOps</i>	22
3.1.2 Melhor abordagem (a nível organizacional) para a adoção de <i>DevOps</i>	22

3.1.3	Fases com maior risco na implementação de <i>DevOps</i>	23
3.1.4	Práticas de <i>DevOps</i> mais impactantes (positivamente) para a implementação de <i>DevOps</i>	25
3.2	Pré- <i>DevOps</i>	26
3.2.1	Motivações para a implementação do <i>DevOps</i>	26
3.2.2	Ferramentas utilizadas para gerir o SDLC anteriores à implementação de <i>DevOps</i>	27
3.2.3	Processos semelhantes a <i>DevOps</i> anteriores à sua implementação	27
3.3	Implementação de <i>DevOps</i>	28
3.3.1	Alternativas à adoção de <i>DevOps</i>	28
3.3.2	Objetivos da implementação de <i>DevOps</i>	28
3.3.3	Melhores práticas na implementação de <i>DevOps</i>	29
3.3.4	Participação de entidades externas	30
3.3.5	Custo estimado da implementação de <i>DevOps</i>	30
3.3.6	Área(s) com maior mudança organizacional/cultural.....	31
3.3.7	Tipologia do processo implementado	32
3.4	Práticas <i>DevOps</i>	33
3.4.1	Práticas implementadas.....	33
3.4.2	Principais fatores influenciadores do sucesso da implementação de <i>DevOps</i>	33
3.5	Técnicas e Ferramentas de <i>DevOps</i>	34
3.5.1	Implementação de <i>DevOps</i>	34
3.5.2	Ferramentas implementadas	35
3.5.3	Utilização da <i>Cloud</i>	36
3.5.4	Sistema de controlo de versões.....	36
3.6	Benefícios <i>DevOps</i>	37
3.6.1	Principais benefícios da adoção de <i>DevOps</i>	37

3.6.2	Colaboração entre equipas	38
3.6.3	Correções de emergência em produção	39
3.6.4	Satisfação do cliente e equipas	40
3.6.5	Satisfação da organização	41
3.6.6	Frequência de entregas.....	41
3.6.7	Automatização	42
3.7	Gestão da mudança	42
3.7.1	Mudança para a cultura <i>DevOps</i> como <i>stakeholder</i>	42
3.7.2	Formação para a equipa responsável pela implementação de <i>DevOps</i>	43
3.7.3	Formação das equipas envolvidas.....	43
3.7.4	Alterações na infraestrutura da organização	44
3.7.5	Alterações estruturais ou reorganização de departamentos da organização	44
3.7.6	Gestão da mudança	44
3.8	Barreiras <i>DevOps</i>	45
3.8.1	Principais obstáculos à adoção de <i>DevOps</i>	45
3.8.2	Resistência à mudança	48
3.8.3	Melhoria da comunicação entre as equipas com a implementação do <i>DevOps</i>	48
3.8.4	Dimensão da organização como barreira.....	49
3.9	Pós-Implementação de <i>DevOps</i>	50
3.9.1	Cumprimento dos objetivos	50
3.9.2	Satisfação como <i>stakeholder</i>	51
3.9.3	Critérios para avaliação do sucesso da implementação de <i>DevOps</i>	52
3.9.4	Avaliação formal do sucesso da implementação de <i>DevOps</i>	53
3.9.5	Aplicação de <i>DevOps</i>	53
3.9.6	Sucesso da adoção de <i>DevOps</i> na organização.....	53

3.10	Considerações finais	54
3.10.1	Adoção do <i>DevOps</i>	54
3.10.2	Benefícios do <i>DevOps</i>	55
3.10.3	Barreiras à adoção do <i>DevOps</i>	56
	CONCLUSÃO	58
	REFERÊNCIAS BIBLIOGRÁFICAS	60
	APÊNDICE.....	63
	APÊNDICE. GUIÃO.....	64

ÍNDICE DE TABELAS E FIGURAS

Tabela 1.1 Entrega de software: Tradicional vs. DevOps	12
Tabela 1.2 Benefícios da cultura DevOps.	14
Tabela 1.3 Barreiras à adoção de DevOps.....	15
Tabela 2.1 Entrevistados.....	19
Figura 1.1 Ciclo de vida de Waterfall vs. Agile SDLC.	4
Figura 1.2 Ciclo DevOps.	10
Figura 2.1 Etapas da metodologia de investigação “Estudo de Caso”	17
Figura 3.1 Melhor abordagem (a nível organizacional) para a adoção de DevOps?.....	23
Figura 3.2 Fases com maior risco na implementação de DevOps	24
Figura 3.3 Práticas de DevOps mais impactantes (positivamente) para a implementação de DevOps	25
Figura 3.4 Processos da organização semelhantes a um processo inerente à cultura DevOps	27
Figura 3.5 Alternativas à adoção de DevOps	28
Figura 3.6 Melhores práticas na implementação de DevOps	29
Figura 3.7 Áreas com maior mudança organizacional/cultural	31
Figura 3.8 Tipologia do processo implementado.....	32
Figura 3.9 Top 5 fatores influenciadores de sucesso	34
Figura 3.10 Implementação de DevOps	34
Figura 3.11 Categorias das ferramentas implementadas	35
Figura 3.12 Sistema de controlo de versões facilitou a adoção de DevOps	36
Figura 3.13 Principais benefícios da adoção de DevOps.....	38

Figura 3.14 As equipas passaram a trabalhar de forma mais colaborativa, após a implementação?	39
Figura 3.15 Frequência das correções de emergência em produção.....	39
Figura 3.16 Satisfação do cliente.....	40
Figura 3.17 Satisfação da equipa de desenvolvimento	40
Figura 3.18 Satisfação da equipa de operações	40
Figura 3.19 Satisfação da organização	41
Figura 3.20 Frequência das entregas.....	41
Figura 3.21 Automatização.....	42
Figura 3.22 Importância da mudança para a cultura DevOps na perspetiva de stakeholder	42
Figura 3.23 Principais obstáculos à adoção de DevOps	46
Figura 3.24 Principais obstáculos à adoção de DevOps	47
Figura 3.25 Resistência à mudança.....	48
Figura 3.26 Melhoria da comunicação entre equipas com a implementação do DevOps	49
Figura 3.27 Dimensão da organização como barreira	49
Figura 3.28 Cumprimento dos objetivos.....	50
Figura 3.29 Satisfação como stakeholder	51
Figura 3.30 Critérios de sucesso de implementação de DevOps	52
Figura 3.31 Aplicação de DevOps	53
Figura 3.32 Sucesso da adoção de DevOps na organização	54

Lista de abreviaturas, acrónimos e siglas

No presente documento são utilizadas as seguintes abreviaturas, acrónimos e siglas:

ARA – Application Release Automation

CI – Continuous Integration

CDD – Continuous Delivery

CD – Continuous Deployment

CT – Continuous Testing

CAMS – Culture, Automation, Measurement, Sharing

CALMS – Culture, Automation, Lean, Measurement, Sharing

MVP – Minimum Value Product

SDLC – Software Development Lifecycle

TI – Tecnologias de Informação

SI – Sistemas de Informação

INTRODUÇÃO

As empresas focadas no desenvolvimento de aplicações informáticas (*software*) necessitam de continuamente procurar melhorar a gestão dos seus projetos, para que os seus produtos/serviços atinjam patamares superiores de qualidade e cheguem mais depressa ao mercado (*time-to-market*). O objetivo primordial é satisfazer os seus clientes, dado que este um é importante indicador do sucesso dos projetos (Varajão, 2018).

É neste contexto que surge, há não muitos anos, o fenómeno do desenvolvimento ágil, o qual procura envolver o cliente no trabalho da equipa de desenvolvimento de *software*, visando produzir *software* de forma mais rápida e sobretudo mais iterativa. Uma vez que o cliente acompanha de forma próxima o desenvolvimento dos projetos, tal resulta num aumento da sua satisfação (Nuottila, J., Aaltonen, K., and Kujala, 2016). Associado ao desenvolvimento ágil e à necessidade de colocar frequentemente as novas versões (parciais) de *software* em produção, surgiu um outro fenómeno: o *DevOps*.

O *DevOps* procura integrar todo o processo de desenvolvimento de *software*, desde a análise de requisitos até à colocação do *software* em produção (Humble & Farley, 2010). Esta integração é desejavelmente feita através de procedimentos automáticos, incluindo, por exemplo, a realização de testes, que procuram garantir que o *software* satisfaz os requisitos de qualidade. Para além da automatização da entrega de *software*, o *DevOps* procura também garantir, através de processos automáticos (como, por exemplo alarmísticas), que os erros detetados no *software* em produção chegam rapidamente ao conhecimento das equipas de desenvolvimento. Assim, estas podem corrigir os eventuais erros e colocar rapidamente em produção uma nova versão do *software*. De modo a que estes mecanismos funcionem, é necessário que as equipas de desenvolvimento e operações colaborem e sejam solidárias entre si. É este tipo de envolvimento entre os elementos das diferentes equipas que faz com que o *DevOps* seja interpretado como um novo movimento dos profissionais de Tecnologias da Informação (TI), em que todos têm como princípio base do seu trabalho a partilha, trabalhando todos para o mesmo objetivo. É, assim, facilmente compreensível que, para que o *DevOps* tenha sucesso, é necessário que o mesmo faça parte da cultura organizacional (Humble & Farley, 2010; Kornilova, 2017; Walls, 2013).

O presente projeto, tem por finalidade estudar o fenómeno *DevOps* nas suas várias vertentes, evidenciando os principais benefícios e barreiras da sua adoção, os quais são

explanados nas próximas secções. Visa, também, realizar um estudo de caso numa organização que opera na Europa na área das telecomunicações.

O relatório está organizado em três capítulos: revisão da literatura, metodologia e análise e discussão de resultados. Para além destes, contempla ainda a introdução e a conclusão.

O primeiro capítulo apresenta a revisão da literatura sobre o fenómeno em estudo, o *DevOps*, apresentado, os pilares, princípios e práticas *DevOps*, bem como, os desafios e barreiras à adoção do *DevOps*.

O segundo capítulo é referente à metodologia de investigação adotada, o estudo de caso, onde são apresentadas as diferentes etapas seguida na realização do trabalho.

No terceiro capítulo são apresentados e discutidos os principais resultados recolhidos, bem como, apresentar algumas considerações finais.

A conclusão apresenta uma síntese do projeto desenvolvido, explanando os contributos, as limitações e os trabalhos futuros.

Em anexo encontra-se o guião desenvolvido para as entrevistas realizadas no âmbito do estudo de caso.

1 REVISÃO DA LITERATURA

Este capítulo apresenta a revisão de literatura dos conceitos associados ao *DevOps*.

1.1 Desenvolvimento e entrega de *software*

Tipicamente o desenvolvimento de *software* assenta sobre vários pilares cruciais ao sucesso do modelo de negócio. Desde o desenvolvimento até produção, existem várias etapas que devem ser respeitadas quando se desenvolve *software*. Uma das grandes diferenças entre as duas abordagens principais do desenvolvimento de *software*, cascata (*waterfall*) e ágil (*agile*), é a forma como os requisitos são definidos (Huckabee, 2015). Em *cascata*, os requisitos são identificados inicialmente no projeto e são esses requisitos que conduzem ao produto final, não sendo facilmente acomodadas alterações. Em *ágil*, os requisitos identificados podem ser alterados ao longo do projeto, consoante o *feedback* do cliente. Este trabalha colaborativamente com a equipa de desenvolvimento, de forma a criar um produto com maior qualidade e que vá de encontro às suas reais necessidades (as quais podem evoluir no decorrer do projeto).

À identificação dos requisitos segue-se o desenvolvimento dos produtos, incluindo o *software*. Posteriormente, o *software* resultante é tipicamente testado por equipas capacitadas para o efeito, que definem e executam casos de testes. O resultado dos testes é usado para dar *feedback* à equipa de desenvolvimento acerca de possíveis erros e correções que sejam necessárias. Os testes são normalmente executados num ambiente de qualidade (Visser, Rigal, Wijnholds, & Lubsen, 2016). Dependendo do produto e da criticidade que este tenha para o modelo de negócio, podem existir mais ambientes de teste, como, por exemplo, pré-produção ou aceitação (Visser et al., 2016). Desta forma, garante-se a entrega de *software* com mais qualidade. Depois dos vários testes serem executados e validados com sucesso, o *software* pode ser colocado em produção e fica pronto a ser utilizado pelo cliente. Caso surjam erros em produção, podem ser efetuadas correções, que frequentemente não passam por todas as fases do ciclo de desenvolvimento, de forma a resolver os problemas com maior rapidez.

O ciclo de vida de desenvolvimento de *software*, em inglês *Software Development Life Cycle* (SDLC), é distinto conforme a abordagem utilizada seja *cascata* ou *ágil*. Ambos têm o mesmo objetivo, ou seja, a entrega de valor para o cliente, mas organiza de forma diferente as atividades de desenvolvimento, como pode ser observado na Figura 1.1.

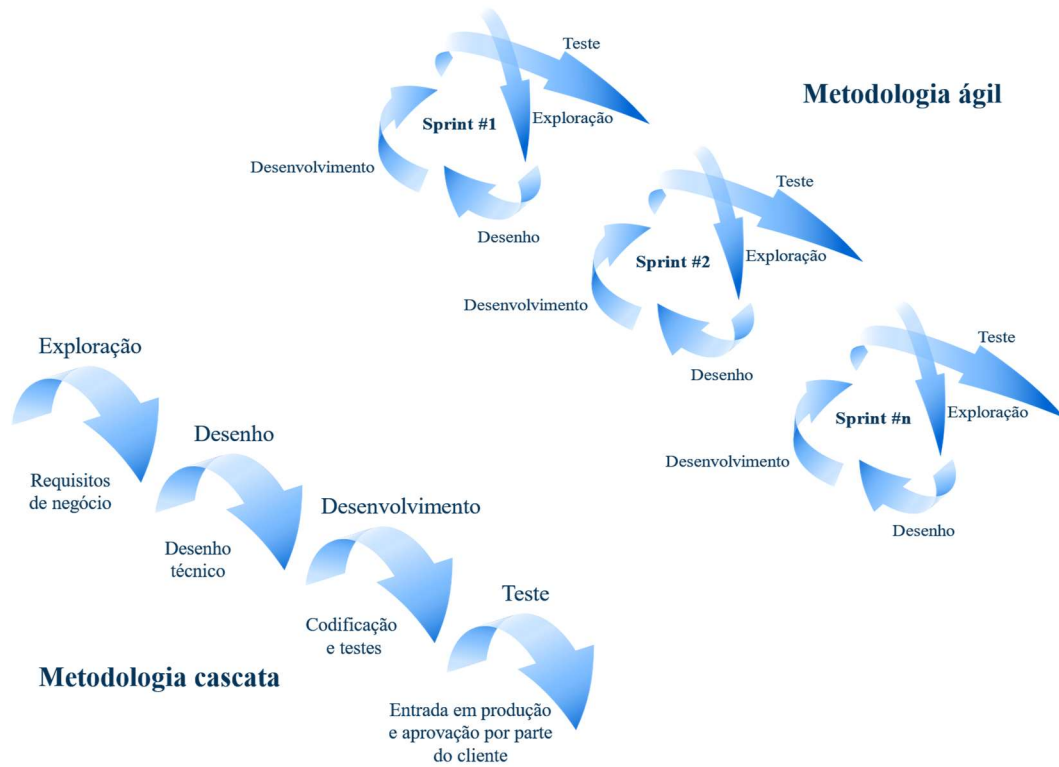


Figura 1.1 Ciclo de vida de Waterfall vs. Agile SDLC.

Fonte: Adaptado de (STH, 2019)

A metodologia *ágil* aplica o mesmo ciclo que é utilizado em *cascata*, porém com iterações mais curtas do ciclo de atividades. Assim, traduz-se numa entrega de valor e *feedback* contínuo (Cois, Yankel, & Connell, 2015).

De forma a suportar as várias etapas de desenvolvimento de *software*, Jobs et al. (2016) defendem que é necessário criar uma infraestrutura com ambientes distintos, mas semelhantes (de forma a que os processos e automatismos não sejam impedimento na entrega de *software*). Estes autores defendem que devem existir quatro ambientes: Desenvolvimento; Qualidade; Aceitação; e Produção. Regra geral, o ambiente de aceitação é um espelho do ambiente de Produção. Contudo, o nome e a quantidade de ambientes podem variar de organização para organização, com base nas regras de negócio estabelecidas por estas. Quando existe um encadeamento destes ambientes, com conceitos de *build/release/deploy* associados, pode-se afirmar que se está perante um *pipeline* de desenvolvimento de *software* que suporta o SDLC (Virmani, 2015). No âmbito do *DevOps*, o *pipeline* é automatizado, surgindo os conceitos de *Continuous Integration* e *Continuous Deployment*, que visam colmatar as lacunas existentes nas metodologias cascata e ágil (Cois et al., 2015).

1.2 DevOps

Não obstante ser muito popular atualmente, o *DevOps* ainda não se encontra bem definido, pois alguns autores identificam-no como sendo uma metodologia de desenvolvimento de *software*, outros como sendo uma abordagem de desenvolvimento de *software*, outros como sendo um método para a gestão de projetos de desenvolvimento de *software*, outros como sendo uma cultura organizacional, e outros ainda como sendo um movimento dos profissionais de TI, como se apresenta nos parágrafos seguintes.

A definição mais consentânea encontrada na literatura é a de que *DevOps* é uma nova abordagem de entrega de aplicações informáticas (*software*), através da colaboração entre as equipas de desenvolvimento e operações, em oposição à abordagem tradicional em que o desenvolvimento e as operações estão separados nos seus silos organizacionais. Embora a palavra ágil não surja muito associada a esta abordagem, a verdade é que o primeiro nome dado a esta abordagem pelos seus criadores, Patrick DuBois e Andrew Clay Schafer, foi “*Agile Infrastructure*” (Debois, 2008).

DevOps deriva de *Dev + Ops (Development + Operations)*, o que remete para a unificação de equipas, transformando os silos existentes num conjunto de equipas que trabalham em prol da organização e não em torno da sua atividade dentro dela, ligando assim todas as diretrizes que já existem no desenvolvimento de *software*. Nas metodologias tradicionais, a área de desenvolvimento está preocupada apenas com uma coisa (“desenvolver”) para o desenvolvimento contínuo de novas *features* e correção de problemas em produção. Enquanto que operações é a área responsável pelo aprovisionamento de *hardware* e *software*, e está apenas preocupada com a estabilidade e a manutenção dos sistemas acima de tudo. O caminho para produção deve ser desimpedido e sem obstáculos, de forma a que as operações consigam realizar o seu trabalho através de *releases* mais rápidas (A. Ravichandran et al., 2016; Fallis, 2013; Kim, Humble, Debois, & Willis, 2016).

1.2.1 Pilares do DevOps

Em 2010 Wills (Wills, 2010) propôs a caracterização do *DevOps* baseada em quatro pilares: cultura (*culture*); automatização (*automation*); medição (*measurement*); e partilha (*sharing*). A sigla CAMS surge desta quatro expressões (*Culture, Automation, Measurement e Sharing*) (Humble & Molesky, 2011). Mais tarde, Jez Humble adicionou a estes quatro pilares o pilar *Lean* (L), passando a sigla a ser CALMS (Riley, 2014). Ambas as siglas são referenciadas na literatura, mais profissional do que académica, o

que realça a natureza eminentemente profissional do movimento. De seguida apresentam-se os cinco pilares do CALMS.

1.2.1.1 Cultura (Culture)

A implementação do *DevOps* traz consigo a proposta de uma nova cultura e mentalidade organizacional, que incentiva a empatia e a colaboração entre as diferentes equipas envolvidas nos processos e de desenvolvimento, e a entrega de *software* evitando os silos organizacionais (Hamunen, 2016; Humble & Molesky, 2011; Laihonen, 2018; Luz, 2018; Riley, 2014). O *DevOps* propõe que esta nova cultura seja intrínseca a todos os elementos da organização, desde a administração aos programadores, passando pelas operações. As pessoas e processos são a prioridade, mas devem ser suportados por metodologias e ferramentas que permitam, de forma contínua e com ciclos de feedback, garantir um processo de melhoria contínua. Toda a organização deve ser um ecossistema cujo resultado é a entrega contínua de valor para o cliente final. A união das equipas deve ser a preocupação número um, aliada à boa comunicação, de forma a que o modelo de negócio flua naturalmente mais rápido, para não criar silos empresariais. Os canais de comunicação possuem um papel crucial na adoção e implementação da cultura, pois estes coordenam e conduzem a organização para os seus objetivos (Walls, 2013).

1.2.1.2 Automatização (Automation)

A automatização dos processos associados ao desenvolvimento e à entrega de *software* é a principal tarefa das organizações que querem adotar o *DevOps*. Os processos que se podem automatizar são diversos. Desde o processo de compilação, testes, disponibilização e instalação (*build*, testes, *release* e *deploy*), até à monitorização, o que permite um *feedback* mais rápido e reduzir os tempos de espera, por exemplo, com alarmísticas para as equipas de operações (Hamunen, 2016). A automatização permite, por um lado, eliminar tarefas monótonas no sentido que são transformadas em ações automáticas, evitando o erro humano, e melhorando a consistência da infraestrutura (Lwakatare, Kuvaja, & Oivo, 2016). Por outro lado, permite libertar recursos humanos da área de TI, tipicamente com competências valiosas (Trigo et al., 2010), para a realização de outras tarefas, que criem valor para a organização (Laihonen, 2018). De facto, os testes automatizados retiram uma grande parte do esforço das equipas de testes para que estas se possam focar em melhorar e criar testes passíveis de serem automatizados, dentro do âmbito dos vários tipos de testes já referidos. Os testes automáticos são os que

proporcionam a integração das práticas de *Continuous Integration* e *Continuous Deployment*, pois só através destes é que é possível obter feedback de forma a que o *software* possa prosseguir para produção (caso os testes tenham sido validados com sucesso).

1.2.1.3 Lean

O *lean* surgiu no setor da indústria automóvel, através da Toyota (*Toyota Production System*). Suporta a produção em larga escala da procura e da melhoria contínua, tendo por base a redução de desperdícios, de diferentes naturezas, como, por exemplo, de tempo e de dinheiro, por forma a entregar mais valor ao cliente (Krafcik, 1988). A cultura *DevOps* está fortemente associada aos processos ágeis de desenvolvimento de *software*, que pressupõem o desenvolvimento de *software* com base no *feedback* do cliente e a entrega de *software* em pequenas iterações e de forma regular (Kim et al., 2016). As metodologias *lean*, focadas na melhoria contínua e sistemática, são um pilar importante do *DevOps* (Kim et al., 2016), tornando o processo de desenvolvimento e de entrega de *software* mais robusto, eficaz e eficiente.

1.2.1.4 Medição (Measurement)

A abordagem *DevOps* sugere que a medição e controlo das equipas de desenvolvimento e de operações sejam feitas em simultâneo, sendo as equipas premiadas segundo os mesmos parâmetros, em que o objetivo comum é a produção de valor para o cliente. O sucesso da equipa de desenvolvimento é medido em função da aplicação que está em produção, utilizando a equipa de desenvolvimento o *feedback* da equipa de produção para a tomada de decisão quanto a novas melhorias e funcionalidades (Lwakatare et al., 2016), possibilitando, assim, a criação de aplicações mais estáveis (Humble & Molesky, 2011). A grande inovação ao nível da medição e controlo do processo de desenvolvimento e de entrega de *software* é o mesmo ser visto de forma holística, em que todas as equipas envolvidas devem colaborar de forma a enriquecer estes dados e a sua qualidade.

1.2.1.5 Colaboração (Sharing)

As equipas envolvidas no processo de desenvolvimento de *software* tradicionalmente não têm uma relação saudável com as equipas de operações: o desenvolvimento está mais preocupado com a entrega do *software*; enquanto que as operações querem garantir que existe estabilidade nas aplicações em produção. A cultura *DevOps* propõe a melhoria

desta relação, com a ideia de que todos devem colaborar, evitando os silos organizacionais, e que as lições aprendidas por uns sejam partilhadas com todos, possibilitando melhorias na cultura (Hamunen, 2016; Humble & Molesky, 2011; Kim et al., 2016; Laihonon, 2018; Luz, 2018; Riley, 2014). Desta forma, a cultura *DevOps* garante uma maior e mais eficiente comunicação, envolvendo todos as equipas no processo (do início ao fim). Como assegura mais responsabilidade a todas as equipas, estas têm de colaborar mais eficientemente. Apesar de não ser esperado que os *developers* executem o trabalho das operações, e vice-versa, é possível que estes colaborem de forma a corrigirem os eventuais problemas mais rapidamente. O maior desafio é quebrar estas barreiras entre as equipas, sendo importante a utilização de ferramentas que facilitem a comunicação. A produção de documentação de forma colaborativa de suporte a todo o processo de entrega e desenvolvimento é outro aspeto importante, pois permite que novos elementos sejam integrados na equipa de forma mais eficiente (Walls, 2013).

1.2.2 Princípios

Existem princípios básicos que devem ser seguidos de forma a que a curva de implementação do *DevOps* seja o mais linear possível (Kim et al., 2016):

- Testar sempre que for possível, de modo a garantir a integridade de todo o ciclo de desenvolvimento e da própria aplicação;
- Melhorar continuamente dos processos implementados, com a adoção de novas técnicas e ferramentas que visem criar uma maior qualidade do produto para que as entregas sejam ainda mais rápidas;
- Automatizar tudo o que for possível, para evitar erro humano e investimento desnecessário em recursos humanos;
- Trabalhar de forma a criar uma única equipa para trabalhar em torno dos mesmos objetivos, eliminando assim os silos existentes;
- Separar o processo por áreas e atividades, mas não de forma disruptiva que vá contra o princípio anterior, mas de forma a que se saiba onde começa e termina cada uma das fases do ciclo;

Estar sempre a par das novas tecnologias de forma a que se consiga criar *Minimum Viable Products* (MVP), com novas soluções que melhorem o modelo de negócio.

1.2.3 Práticas

Existem várias práticas que têm de se ter em conta quando se aborda *DevOps*, tais como, *Continuous Integration (CI)*, *Continuous Delivery* e *Continuous Deployment (CD)* (Shahin, Ali Babar, & Zhu, 2017), as quais são descritas de seguida:

- *Continuous Integration* é a prática mais comum das três: visa integrar o trabalho em progresso, de forma a que toda a equipa tenha *feedback* constante do desenvolvimento paralelo. Consiste em estar ligado continuamente a um repositório de código, de forma a detetar novas alterações (*webhook*) num determinado ramo (*branch*). Quando são detetadas novas alterações, é lançado um processo onde o código é compilado, podendo ou não ser testado através de testes unitários para uma primeira validação, de forma a confirmar que as alterações efetuadas estão executáveis.
- *Continuous Delivery* é a prática que garante que o código compilado está pronto para ser disponibilizado nos diversos ambientes da organização. Depois de passar todos os controlos de qualidade que visam garantir que o risco do *deploy* falhar é mais baixo (numa determinada máquina), pode então ser disponibilizada a nova alteração no ambiente em questão. Nesta prática, o código é disponibilizado depois do sucesso nos dois primeiros ambientes. Para produção existe um *gatekeeper* que decide se as alterações avançam para produção. Esta prática pressupõe obrigatoriamente o uso de um passo manual no processo.
- *Continuous Deployment* é a prática que visa eliminar o passo manual referido anteriormente. Desta forma, sempre que um ciclo é concluído com sucesso, isso significa que existem novas alterações em produção. Tal também significa que as alterações efetuadas por uma equipa podem ser instaladas independentemente do estado das alterações de outra equipa. Deste forma, é possível aumentar a produtividade e diminuir os ciclos de *release*.

Estas três práticas são as mais comuns e seguidas pela comunidade *DevOps* e, se bem implementadas, são o caminho para o sucesso na adoção da cultura *DevOps*. Em linhas gerais, pode-se afirmar que CI/CD leva à produção de novas versões de *software* (*releases*) mais rapidamente e com mais frequência, de forma a eliminar os obstáculos que existem entre o desenvolvimento e a produção. O cliente é beneficiado, pois obtém um acesso mais rápido e frequente a novas funcionalidades.

1.2.4 Ciclo DevOps

O ciclo de *DevOps* apresentado na Figura 1.2 ilustra todos os processos inerentes ao desenvolvimento de uma aplicação.

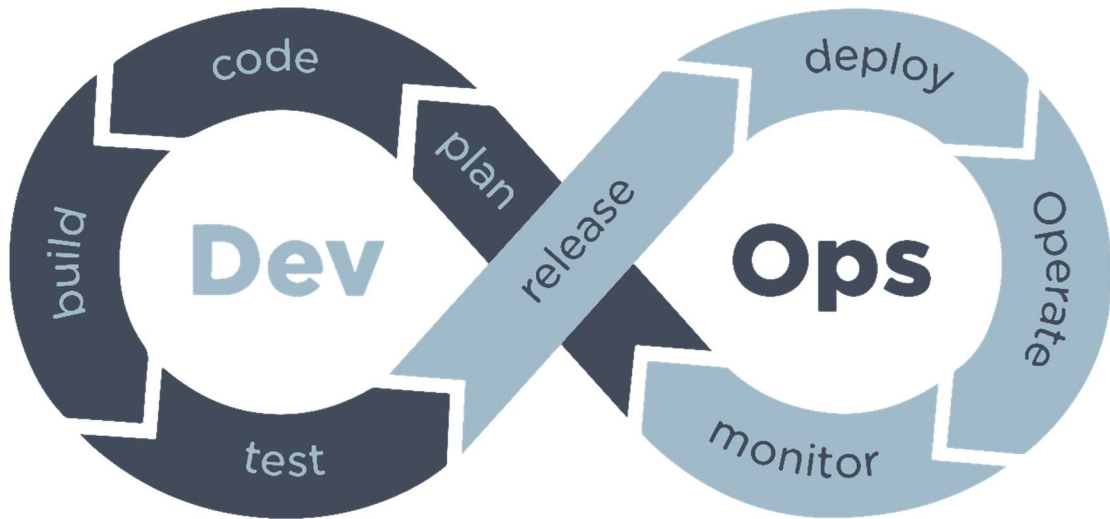


Figura 1.2 Ciclo DevOps.

Fonte: Kornilova (2017)

O planeamento é considerado a primeira fase, seguida da codificação. Estas duas fases são normalmente definidas pelas equipas de desenvolvimento, que executam e a equipa de negócio que define os requisitos a implementar. A fase de *build*, *test*, *release* e *deploy*, são as fases que por norma são automatizadas através de práticas de CI/CD. Nestas fases o código desenvolvido anteriormente é compilado, testado e instalado nos ambientes até chegar ao cliente final. A fase de *build* é onde acontece a compilação do código, e também, alguns testes iniciais, como por exemplo testes unitários. A fase seguinte é a de testes, e é onde são efetuados testes unitários, de aceitação, integridade, performance e não funcionais. Destes testes resulta a validação de código, serviços, escalabilidade segurança, disponibilidade, entre outros. A fase de *release* está diretamente relacionada com a de *deploy*, enquanto que *release* é o ato de disponibilizar para o público um conjunto de alterações, *deploy* é a execução do processo de instalação num determinado ambiente. As duas fases finais são de operações e monitorização, onde as equipas garantem a integridade e manutenção das aplicações. Também são recolhidas métricas para os próximos desenvolvimentos de forma a assegurar uma melhoria constante. Através deste ciclo e com *feedback* constante em cada fase é possível entregar *software* ao cliente final com mais valor.

Ao processo de *build* está inerente o uso de um repositório de código, um sistema de controlo de versões, que permite que os *developers* disponibilizem e visionem as suas alterações, para que estas possam seguir até ao cliente final. Estes sistemas são os responsáveis pela simplicidade do desenvolvimento em larga escala. Fundamentalmente existem dois tipos de sistemas de controlo de versões: distribuídos e centralizados. Os sistemas centralizados têm o código centralizado num determinado servidor e todas as alterações são efetuadas diretamente lá, um exemplo é o *Subversion* (SVN). Os sistemas distribuídos, por outro lado, usam cópias locais do repositório em cada computador de cada *developer*, o que garante um histórico de alterações para que, quando as alterações forem subitadas ao servidor, exista um menor risco de surgirem problemas no código. Um dos exemplos mais utilizados mundialmente é o GIT, que é a base de um dos maiores repositórios que existem na *cloud*, o *GitHub*.

1.2.4.1 Ferramentas de Suporte ao ciclo

O ciclo *DevOps* integra uma série de ferramentas distintas. Em cada uma das fases também são usadas diferentes ferramentas que suportam todo o ciclo.

Para o planeamento podem-se diversas ferramentas, como é o caso das ferramentas do *Office* da *Microsoft*, que contêm ferramentas de planeamento e de gestão de projetos. Uma das ferramentas mais utilizadas mundialmente, o *Jira*, visa suportar todo o SDLC e tem por base métodos ágeis, como o *Scrum* ou o *Kanban*, utilizando quadros colaborativos com os seguintes estados base: *To Do*, *In Progress* e *Done*. É também usada para planeamento e análise dos requisitos (Bass, Weber, & Zhu, 2015).

Para o *build* está implícito um repositório, como já referido anteriormente. Através de ferramentas de integração, como, por exemplo, o *Jenkins*, com o repositório pode-se configurar a prática de *continuous integration*. O processo de *build* é normalmente definido pelos *developers*, onde estes escolhem, consoante a tecnologia, qual o melhor processo para a aplicação em desenvolvimento. O *Jenkins* é uma ferramenta *open-source* que permite diversas customizações, o que garante uma grande versatilidade, e o torna numa das ferramentas mais utilizadas para este efeito. Devido a essa versatilidade esta ferramenta também pode ser utilizada para a execução de *deploys*, apesar de existirem ferramentas mais direcionadas para o efeito, como, por exemplo, o *Ansible* e o *XL Deploy*, ou outras ferramentas “*low-code*”, que permitem a automatização mais célere. Tal, para grandes organizações, pode ser um problema, pois os sistemas *legacy* que estas

possuem podem não ser compatíveis com estas ferramentas, e para isso necessitam de desenvolver o seu próprio processo. Com esse desenvolvimento existe um maior controlo de toda a infraestrutura, o que permite uma maior agilidade no processo de *build, deploy e release*. Para *release*, pode recorrer-se a um ARA (*Application Release Automation*), que permite um controlo de *releases* sem codificação de uma plataforma ou infraestrutura (Bass et al., 2015).

A escalabilidade aplicacional é aplicada de forma contínua, tirando partido de ferramentas de containerização, como é o caso do *Docker*. A portabilidade é vista como o pilar deste tipo de ferramentas. Esta ferramenta visa utilizar e partilhar os recursos de uma máquina, isolando aplicações em *containers*¹.

Nas fases de operação e monitorização são utilizadas muitas ferramentas que possibilitam o *feedback* contínuo e devolvem alarmísticas, para a equipa ser capaz de responder a eventuais falhas ou problemas detetados. Exemplos destas ferramentas são o *Zabbix* e o *Nagios* (Bass et al., 2015)

1.2.5 Abordagem tradicional vs. DevOps

Na Tabela 1.1 apresenta-se uma lista das características da entrega de *software* do modelo tradicional vs. *DevOps*.

Tabela 1.1 Entrega de software: Tradicional vs. DevOps

Característica	Tradicional	DevOps	Referência
Comunicação	Silos	Colaborativa	(Walls, 2013)
Definição de requisitos de <i>software</i>	Definidos apenas no início do projeto	Podem ser alterados ao longo do projeto	(Huckabee, 2015)
<i>Deploy</i> em produção	Tipicamente manual	CI/CD	(Kim et al., 2016)
Automatização de processos	Podem existir processo automatizados	SDLC totalmente automatizado	(Erich, Amrit, & Daneva, 2017)
Colaboração entre equipas	Produto final com os requisitos desenvolvidos no início do projeto	Produto iterativo com valor crescente	(Erich et al., 2017)
Equipas de desenvolvimento	Preocupadas apenas em desenvolver os requisitos e entregá-los	Colaboram ao longo de todo o SDLC para uma maior entrega de valor	(Riungu-Kalliosaari, Mäkinen, Lwakatare, Tiihonen, & Männistö, 2016)
Equipas de operações	Preocupadas apenas em manter as aplicações estáveis	Gestão colaborativa com as equipas de desenvolvimento	(Riungu-Kalliosaari et al., 2016)

¹ unidade padrão de *software* que empacota o código e todas as suas dependências, para que o aplicativo seja executado de maneira rápida e confiável, de um ambiente de computação para outro (Docker, 2019)

1.3 Benefícios do DevOps

Existem vários géneros de vantagens resultantes da adoção do *DevOps*, os quais podem ser agrupadas em benefícios técnicos, culturais e de negócio (Kim et al., 2016). Os benefícios técnicos estão relacionados com as práticas de CI/CD e com a velocidade de resolução de problemas. Os benefícios culturais contribuem para a melhoria da comunicação e para a criação de laços mais fortes entre os colaboradores das diferentes equipas e, conseqüentemente, para se ter colaboradores mais motivados, que contribuam para alcançar a missão e a visão da organização. Os benefícios de negócio estão relacionados com o cliente final e com a velocidade de entrega de valor, garantindo ainda uma maior estabilidade do negócio e mais espaço para inovar.

Os diversos benefícios são o resultado da boa aplicação dos princípios e práticas descritos anteriormente. Versionamento de código, integração contínua, testes contínuos e *deploy* contínuo, tudo com base numa monitorização contínua. Isto permite dar lugar a um princípio base do *lean*, a melhoria contínua (Kim et al., 2016). A cultura *DevOps* deve ser vista como um acelerador do desenvolvimento de *software*, que se baseia em metodologias *ágeis* de desenvolvimento. Podem adaptar-se as ferramentas aos processos ou vice-versa. Esta cultura permite uma abordagem híbrida, em que podem ser privilegiados alguns princípios em detrimento de outros, para que seja possível adotar gradualmente todas as práticas.

Automatizar é um dos maiores benefícios de que se pode tirar partido. Assim é possível providenciar as práticas já descritas, e utilizar os mesmos métodos em todos os ambientes de forma a libertar recursos humanos (Laihonen, 2018).

A entrega de valor rápida e contínua é um dos grandes benefícios dos clientes (Shahin et al., 2017). O *feedback* que pode ser recolhido dos clientes através de implementações de recolha de métricas para a construção de *dashboards*, é também uma das mais-valias (Shahin et al., 2017).

O *DevOps* permite o aumento da performance e da produtividade organizacional do departamento de TI, redução de custos no ciclo de vida de desenvolvimento de *software*, melhoria na eficiência e eficácia operacional, e um maior alinhamento com o negócio entre as equipas de desenvolvimento e de operações (Forsgren, Humble, Kim, Brown, & Kersten, 2017; Luz, 2018).

Na Tabela 1.2 é apresentada uma síntese dos benefícios de DevOps.

Tabela 1.2 Benefícios da cultura DevOps.

Benefícios	Referência
Melhoria da comunicação entre equipas	(Amaradri & Nutalapati, 2016; Lwakatare et al., 2019; Senapathi, Buchan, & Osman, 2018)
Maior partilha de conhecimento entre equipas	(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)
Monitorização contínua	(Amaradri & Nutalapati, 2016; Shahin et al., 2017)
Entrega de valor contínua	(Lwakatare, 2017; Shahin et al., 2017)
<i>Feedback</i> 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)

1.4 Desafios e barreias à adoção

Bass et al. (2015) questionam a razão do *DevOps*, que procura melhorar a comunicação entre as equipas de desenvolvimento e de operações, ainda não ser uma prática generalizada. Segundo estes autores, o principal problema é a resistência à mudança. Vista como uma das principais questões nas organizações, é necessário treinar e motivar as pessoas para que a adoção seja feita sem entraves e haja vontade de transformar e melhorar os processos atuais, de forma a que a visão da empresa seja atingida mais rapidamente.

Outra barreira é a falta de comunicação, dado que as empresas, por norma, trabalham em silos. Negócio, Desenvolvimento e Operações, cada uma destas áreas está apenas preocupada com os problemas da sua área e na procura de soluções para eles. Para que isto mude, é necessário que cada equipa assuma responsabilidades novas e diferentes, de

forma a que o objetivo final, entrega contínua de valor, seja o mesmo para todas as áreas (Riungu-Kalliosaari et al., 2016).

A dimensão da organização é vista como um desafio, sendo mais fácil implementar a cultura e práticas *DevOps* em organizações com poucos colaboradores e equipas pequenas, como numa *startup*, do que em grandes organizações multinacionais. Independentemente da dimensão da organização, um desafio que se coloca é o tipo de abordagem para a implementação a seguir: *top-down* ou *bottom-up*. A primeira, responsabiliza a administração, que tem de inculcar a cultura aos restantes colaboradores da organização. Se, por outro lado, for escolhida uma abordagem *bottom-up*, são os programadores os responsáveis por apresentar as vantagens da adoção aos seus responsáveis, e assim sucessivamente até chegar à administração.

Quando se considera a adoção da cultura *DevOps* é necessário ter em conta os benefícios inerentes, como, por exemplo, a mais rápida e contínua entrega de valor ao cliente final, contra o risco de surgirem problemas na implementação. É normal que estes aconteçam, o que torna necessário que a equipa implementadora esteja treinada para minimizar os riscos, e que seja capaz de criar planos de contingência válidos.

É necessário ter em conta a constante evolução e mutação dos processos inerentes à cultura *DevOps*, dado que podem envolver a mudança de práticas, ferramentas e definições, e as organizações têm de estar preparadas para se ajustarem sempre que necessário (Riungu-Kalliosaari et al., 2016).

Na Tabela 1.3 apresenta-se uma síntese de benefícios à adoção de DevOps.

Tabela 1.3 Barreiras à adoção de DevOps

Barreiras	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; Riungu-Kalliosaari et al., 2016)
Mudança das ferramentas	(Lwakatare, 2017; Lwakatare et al., 2019; Riungu-Kalliosaari et al., 2016)
Dimensão da organização	(Amaradri & Nutalapati, 2016)
Falta de competências técnicas	(Lwakatare et al., 2019)

Estas barreiras culturais, técnicas e de negócio podem ser ultrapassadas de diversas formas. As barreiras culturais podem ser vistas como um inibidor da adoção da cultura e devem ser transpostas através da melhoria da comunicação entre as equipas, colocando-as a trabalhar de forma mais centralizada e conjunta, como, por exemplo, realizando o desenvolvimento com as operações, por forma a criarem laços e a partilharem responsabilidades superando os problemas decorrentes de *releases* e implementações. Quanto às barreiras técnicas, são as mais fáceis de superar por serem aquelas em que apenas é necessário dotar as equipas de formação adequada para desempenharem as novas funções ou utilizarem novas ferramentas que são impostas pela adoção de *DevOps*. As barreiras no negócio são as mais difíceis de superar, por não ser possível mudar o tamanho da organização de forma a superar esta barreira. A solução as ultrapassar pode passar por criar equipas que contenham elementos de diversas áreas, que trabalhem em prol de um projeto conjunto, de forma a que a entrega de valor dos elementos seja focada nos requisitos impostos pelo negócio.

2 METODOLOGIA

A metodologia de investigação adotada para suportar o desenvolvimento deste projeto foi a metodologia estudo de caso, do tipo descritivo (Runeson & Höst, 2009), pois o objetivo da investigação conduzida foi o de estudar o fenómeno *DevOps* dentro do seu contexto, procurando uma melhor caracterização do mesmo.

O estudo de caso é a investigação qualitativa mais utilizada em sistemas de informação, sendo uma metodologia apropriada para a compreensão das interações entre as inovações relacionadas com a tecnologia da informação e os contextos organizacionais (Darke, Shanks, & Broadbent, 1998).

2.1 Estudo de caso

Segundo (Yin, 1989) um estudo de caso é um estudo empírico que investiga fenómenos contemporâneos referindo-se a um evento, uma entidade, um indivíduo ou uma unidade de análise. É através da recolha de evidências do contexto da vida real que é possível esclarecer o estudo em causa.

2.2 Etapas do estudo de caso

Na Figura 2.1 apresenta-se um diagrama com as etapas para a condução da investigação com a metodologia de “estudo de caso”.

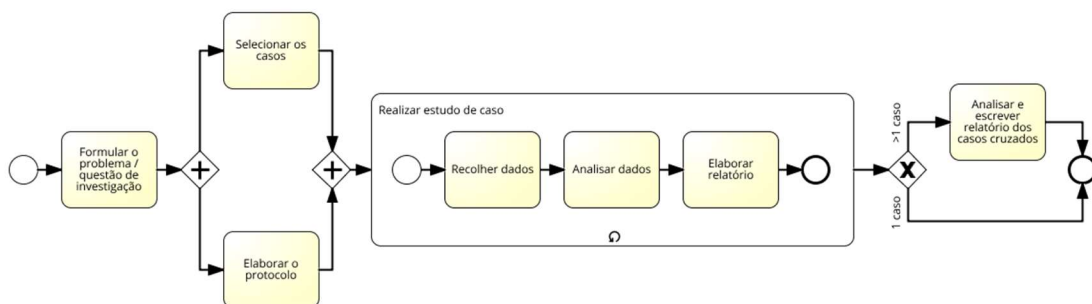


Figura 2.1 Etapas da metodologia de investigação “Estudo de Caso”

Fonte: Adaptado de Yin (2009)

De seguida apresentam-se as etapas e o trabalho realizado em cada uma para a condução da investigação.

2.2.1 Formulação do problema

O objetivo da investigação é o estudo do fenómeno *DevOps*, caracterizado no capítulo de revisão de literatura, com o intuito de adquirir um conhecimento abrangente sobre o mesmo, incluindo aspetos sobre a implementação, ferramentas e técnicas utilizadas, benefícios e barreiras à sua utilização e sucesso da sua adoção.

2.2.2 Seleção dos casos

Na etapa de definição da unidade de estudo, também designada de unidade em análise, escolhem-se os casos que se pretendem estudar. Aqui a abordagem pode ser a de estudar um caso único ou múltiplos casos (Yin, 2009). Para o trabalho de investigação apresentado neste relatório foi selecionado um único caso.

O caso selecionado para este estudo foi uma empresa europeia da área das telecomunicações. Esta entidade distribui serviços de internet, telefone, televisão e telemóvel. Também é responsável pela sua infraestrutura. No ano de 2018 teve um volume de negócios superior a 1500 milhões de euros, e conta com mais de 1800 colaboradores.

A escolha desta organização em particular deveu-se ao facto de a mesma estar a passar por um processo de adoção da cultura *DevOps*, e do acesso facilitado do autor do trabalho à mesma.

2.2.3 Elaboração do protocolo

O protocolo contém os elementos de recolha de dados e a definição da conduta a ser adotada para a sua aplicação, e deve incluir as seguintes secções: visão geral do estudo de caso; procedimentos de campo; questões do estudo de caso; e a forma como irão ser reportados os resultados (Yin, 2009).

A primeira etapa já foi apresentada anteriormente, na secção da questão de investigação e de seleção dos casos, e a última será apresentada no próximo capítulo. Assim, apresentam-se de seguida, o procedimento de trabalho de campo e as questões de estudo com a apresentação do guião criado para recolha de dados.

2.2.3.1 Procedimentos de campo

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

- Observação direta no local onde a equipa de implementação desempenha as suas funções;
- Realização de entrevistas aos participantes no fenómeno em estudo.

A observação direta foi efetuada ao nível dos processos, dos comportamentos e das ferramentas utilizadas com vista a avaliar o sucesso da adoção da cultura *DevOps*.

As entrevistas foram realizadas a um grupo restrito de pessoas relevantes para o estudo de caso, que têm conhecimento efetivo sobre a cultura implementada, os seus moldes, pré e pós implementação.

O grupo de entrevistados encontra-se caracterizado na Tabela 2.1.

Tabela 2.1 Entrevistados

Nome	Área	Função	Experiência (anos)
Entrevistado A	Equipa <i>DevOps</i>	Implementador	3
Entrevistado B	Equipa <i>DevOps</i>	Implementador	11
Entrevistado C	Equipa <i>DevOps</i>	Implementador	2+
Entrevistado D	Equipa <i>DevOps</i>	Implementador	1+
Entrevistado E	Equipa <i>DevOps</i>	Implementador	11
Entrevistado F	Equipa <i>DevOps</i>	<i>Scrum Master</i> , Arquiteto <i>software</i>	9
Entrevistado G	Equipa <i>Configuration Management</i>	Garantir funcionalidade de todo o <i>SDLC</i>	20
Entrevistado H	Equipa <i>Configuration Management</i>	Garantir funcionalidade de todo o <i>SDLC</i>	14
Entrevistado I	Gestão	Liderar as equipas de <i>DevOps</i> e <i>Configuration Management</i>	15
Entrevistado J	Teste Automáticos	Garantir a implementação dos testes automatizados ao longo do <i>SDLC</i>	3
Entrevistado K	Negócio	<i>Product Owner</i> ; Voz da equipa perante os superiores na organização	18
Entrevistado L	Gestão	Gerir o serviço de equipas de TI relacionadas com o <i>SDLC</i> na organização (<i>DevOps</i> , Testes e <i>Configuration Management</i>)	20
Entrevistado M	Gestão	Gestor de projeto	15

Este grupo perfaz um total de 13 elementos da mesma organização, mas de áreas distintas, de forma a garantir uma maior complementaridade de opiniões. A equipa de implementação da cultura, que foi responsável pela alavancagem da maior parte das

iniciativas de *DevOps*, consegue ter uma visão mais completa sobre o conceito, as técnicas, as ferramentas e as práticas.

O *scrum master* tem um papel de excelência na equipa. Definiu todos processos e ferramentas a serem utilizados, sendo agora coordenados por ele. As pessoas relativas ao suporte do SDLC, são as mais ligadas aos procedimentos anteriores e conseguem ter uma visão mais centralizada no pré-*DevOps* e no pós-*DevOps*.

O responsável por estas equipas, tem visão global de todos os processos e ajuda na tomada de decisões e aconselhamento. É um elemento externo à equipa de implementação da cultura que é responsável por trazer os testes automatizados para a pipeline de CI/CD.

O responsável ligado ao negócio, conhece o caminho seguido, sabe o que é pretendido alcançar, e é visto como a voz da equipa de implementação para todas as iniciativas *bottom-up* para conseguir chegar à administração.

O responsável das equipas do departamento onde está a ser liderada a iniciativa da mudança, nomeadamente a equipa de implementação, *configuration management* e testes, tem uma visão geral do panorama, como também uma visão das restantes equipas.

O *manager* da área de *DevOps*, conhece a organização e o projeto, com conhecimentos orientados para a implementação da cultura.

2.2.3.2 Guião

O guião desenvolvido (ver apêndice) foi utilizado para suporte às entrevistas, de forma a que todos os entrevistados respondessem às mesmas questões com o propósito de delimitar a entrevista apenas ao essencial. Através de questões de resposta fechada, escolha múltipla ou escala de *Likert*, e sempre com uma questão subjacente à fundamentação da “resposta”, foi possível garantir um maior contexto da resposta dada pelo entrevistado. O guião foi estruturado em nove secções:

- Secção I: Definição *DevOps*;
- Secção II: Pré-*DevOps*;
- Secção III: Implementação de *DevOps*;
- Secção IV: Práticas *DevOps*;
- Secção V: Técnicas e Ferramentas de *DevOps*;
- Secção VI: Benefícios *DevOps*;
- Secção VII: Gestão da mudança;

- Secção VIII: Barreiras *DevOps*;
- Secção IX: Pós-Implementação de *DevOps*.

2.2.4 Realizar o estudo de caso

A etapa de realização do estudo de caso, como se pode ver na Figura 2.1 é composta por três atividades, as quais são referidas de seguida.

A recolha de dados foi feita na organização selecionada para o estudo de caso, tendo decorrido entre os meses de maio e setembro.

Foi feita a observação direta ao longo de todo o período. Este método permitiu a identificação dos processos e ferramentas utilizadas. Também auxiliou na descrição global do projeto. Através do acesso a documentação, foi possível identificar os processos e a relação entre eles.

Para além da observação direta foram também realizadas 13 entrevistas a vários *stakeholders* do projeto. A cinco elementos da equipa de implementação, a dois da equipa de *configuration management*, ao *scrum master*, ao *product owner*, ao *team leader* das equipas referidas anteriormente, a um elemento da equipa de testes automatizados, ao gestor de serviço, e a um manager relacionado com o projeto.

A avaliação e análise de resultados é apresentada no capítulo três.

2.2.5 Elaboração do relatório de casos cruzados

A atividade não foi realizada dado não fazer sentido no presente contexto (de caso único).

3 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 guião (ver apêndice).

Nesta apresentação de resultados são apresentados excertos das entrevistas realizadas durante a condução do estudo de caso. Nestes excertos, os entrevistados são sempre identificados com a letra que lhes foi atribuída na Tabela 2.1.

3.1 Definição *DevOps*

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

3.1.1 O que é o *DevOps*

Apesar de não existir apenas uma resposta “correta”, *DevOps* pode ser visto de diversas formas. Desde a aproximação das equipas, melhoria da comunicação, entregas mais rápidas e com maior qualidade, melhoria dos processos e redução de custos. Neste sentido, segundo o líder das equipas de *DevOps* e *Configuration Management*:

“É uma metodologia organizacional que tem como grande objetivo proporcionar a gestão do SDLC da forma mais competente, ágil e eficiente, reduzindo assim os custos globais. Tem como principais pilares a standardização dos processos, a automação dos mesmos e a aproximação organizacional dos intervenientes.” (I)

O *DevOps* não deve ser visto apenas como a automatização dos processos e monitorização dos mesmos, pois *DevOps* é muito mais do que isso, como defende um elemento da equipa de testes automatizados:

“*DevOps* é a automação e monitorização da construção do software em todas as fases.” (J)

3.1.2 Melhor abordagem (a nível organizacional) para a adoção de *DevOps*

Na Figura 3.1 é possível verificar que 46,2% dos entrevistados considera que a melhor abordagem para a adoção da cultura *DevOps* é a *Bottom-Up*, 38,5% considera que é a *Top-Down*, e 15,4% considera que existem outras abordagens capazes de alavancar a cultura. 7,7% crê que depende de cada organização, e os outros 7,7% consideram *Bottom-Up*, mas com coadjuvação da organização.

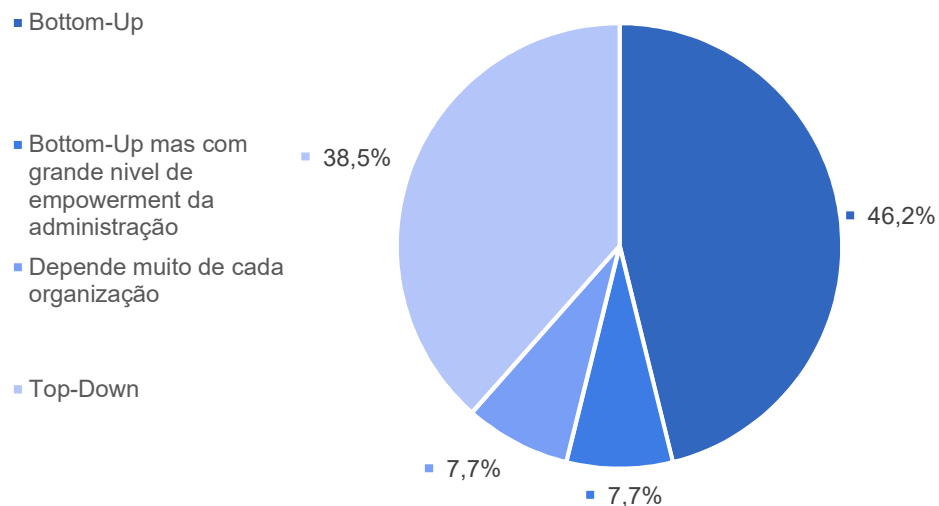


Figura 3.1 Melhor abordagem (a nível organizacional) para a adoção de DevOps?

Não existe uma forma correta de se implementar esta cultura, ambas as perspetivas são válidas e têm vantagens diferentes dependendo da maturidade da organização.

A abordagem *Bottom-Up* representa a colaboração dos intervenientes no processo de mudança de forma a alcançar a administração através dos diversos patamares hierárquicos.

Um dos elementos da equipa de implementação defende que:

“DevOps serve para “ajudar” o ciclo de vida do software e quem o constrói, por isso essas pessoas devem acompanhar o processo para o mesmo estar o mais afinado e infalível. Isto vai fazer com que a entrega seja mais rápida e a administração vai acabar por aceitar e entrar no “barco””. (D)

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

“Com cultura bottom-up irá ser encontrada mais resistência à adoção à medida que se avança na hierarquia. Torna-se muito mais simples/rápido a adoção de novas metodologias/tecnologias se tivermos o enforcing do topo da hierarquia primeiro.” (F)

3.1.3 Fases com maior risco na implementação de DevOps

Cada uma das fases listadas na Figura 1.2 representa uma etapa do SDLC. Desde o planeamento do desenvolvimento, até à monitorização em produção dos sistemas desenvolvidos.

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 61,5% dos entrevistados considerado que a fase com maior risco de implementação é a fase de planeamento, seguida da fase de *release* com 38,5%. 30,8% consideram que a fase com mais risco de implementação é a de testes. Para as restantes fases, 23,1% para a codificação e compilação. 15,4% consideram a fase de *deploy*, enquanto que 7,7% considera a fase de operação e monitorização.

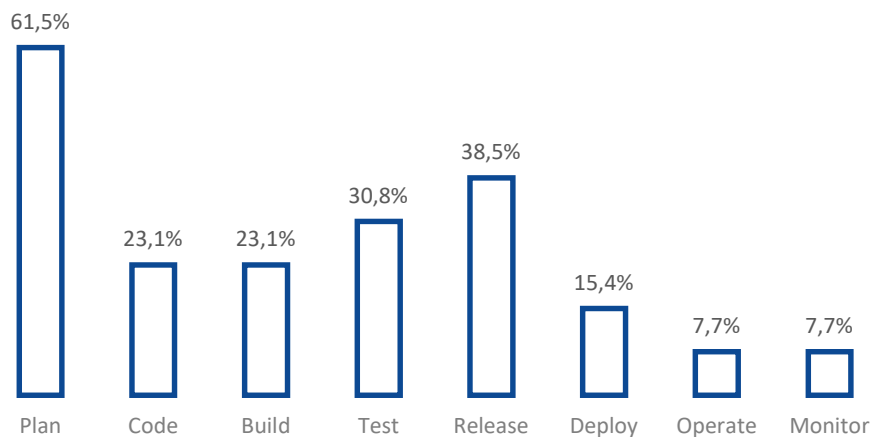


Figura 3.2 Fases com maior risco na implementação de DevOps

O planeamento é uma fase de extrema importância, conferindo assim o maior risco de implementação de acordo com dois dos elementos da equipa de implementação,

“Sem um bom planeamento acabam por vezes por haver imprevistos comprometedores, acarretando custos extra e atrasos. Frequentemente a fase de planeamento é erradamente um pouco desvalorizada devido a eventuais pressões e urgências. A desvalorização do passo anterior, pode levar a código “spaghetti”.” (E)

“Quando as alterações são geridas de forma ágil, é necessário ter em conta um conjunto de fatores, especialmente quando a organização vem de uma metodologia cascata. Este processo era definido através de interação humana e passa a ser parcialmente automatizada, o que pode gerar falta de confiança no processo de release.” (A)

O gestor de projeto afirma que:

“Os testes automáticos têm de ser fiáveis de forma a não comprometer a entrega automática em produção das funcionalidades, caso contrário, apenas estaremos a colocar rapidamente problemas em produção! Da mesma forma, o processo de release é essencial para garantir que dependências são cumpridas nas passagens do SDLC.” (M)

3.1.4 Práticas de DevOps mais impactantes (positivamente) para a implementação de DevOps

A Figura 3.3 apresenta resposta dos entrevistados quanto às práticas mais impactantes (de forma positiva) para a implementação de DevOps. No topo foi escolhida a fase de Entrega Contínua (*Continuous Delivery*), com 76,9% dos entrevistados a considerarem que esta é a prática mais impactante. Logo abaixo, com 69,2%, a prática de integração contínua, que está diretamente relacionada e dependente da anterior. Desenvolvimento colaborativo e testes automatizados com 61,5%, seguido de implantação contínua (*Continuous Deployment*) e Planeamento com 38,5%. 30,8% respondem que a fase do planeamento e feedback contínuo do cliente são das mais impactantes da implementação de DevOps. Por último, com 7,7%, a considerar que a prática de monitorização contínua é das mais impactantes.

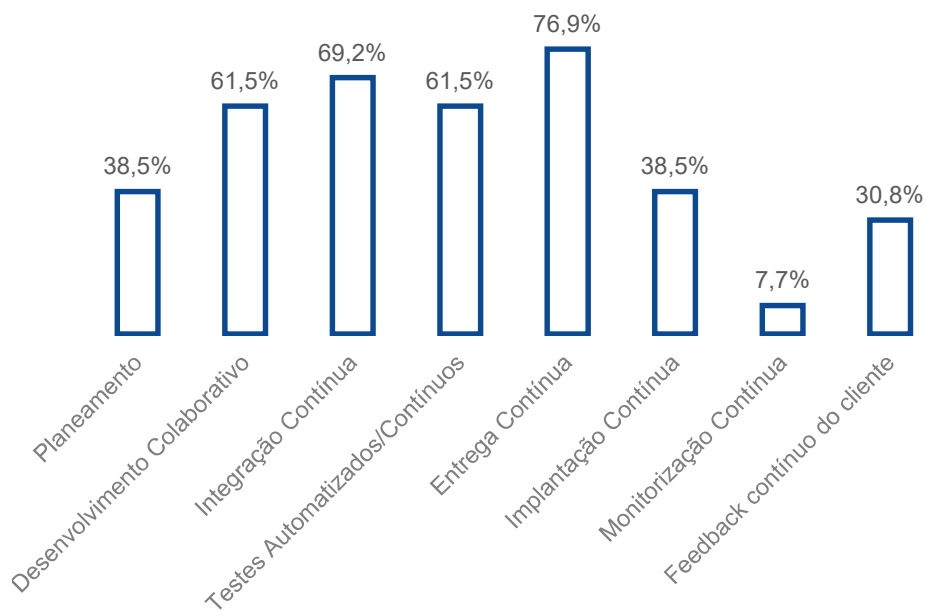


Figura 3.3 Práticas de DevOps mais impactantes (positivamente) para a implementação de DevOps

O entrevistado A, pertencente à equipa implementadora, defende que:

“As práticas de CI/CD são as que mais impacto têm, porque conseguem anular processos manuais e tempo de forma a conseguir-se uma entrega mais rápida”. (A)

Já o entrevistado B considera que:

“O DevOps é transversal entre equipas. Isso significa que para atingir o sucesso é importante que todos os stakeholders estejam sempre envolvidos no SDLC. A integração e testes contínuos permitem descobrir eventuais bugs mais cedo no ciclo e a entrega contínua leva a entregas mais frequentes e pequenas com menos risco.” (B)

O *scrum master* acredita que:

“Em projetos que não tenham maturidade de forma geral nos processos, implementar os processos de CI/CD/CDD e Desenvolvimento colaborativo, permite demonstrar em todos os níveis no cliente (desde gestão com aumento na rapidez e fiabilidade da entrega) como permitir ao desenvolvimento focar-se no desenvolvimento. E permitir as equipas de operação uma monitorização dos ambientes com processos automatizados.” (F)

A perspetiva do elemento da equipa de testes automatizados é:

“Com um desenvolvimento em conjunto, torna-se mais rápido e mais fácil superar as dificuldades. Com testes automáticos, diminui a probabilidade de erro na execução dos testes de deverão ser efetuados ao desenvolvimento em curso.” (J)

3.2 Pré-DevOps

Na segunda seção do guião, 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.

3.2.1 Motivações para a implementação do *DevOps*

De forma geral, 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. Como refere um dos entrevistados, a grande motivação foi:

“Problemas com as entregas e problemas de comunicação entre equipas, dado que os Devs têm uma linguagem mais específica que as Ops não entendem na totalidade.” (C)

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

Outro dos elementos da equipa de implementação afirmam que as motivações para a adoção de *DevOps* foram:

“Problemas de comunicação, implementação de boas práticas.” (B)

Um dos elementos da equipa de *Configuration Management* considera que as motivações foram relativas à deteção preliminar de erros asseguradas de passagens a produção mais rápidas.

“Deteção dos erros mais rápida e passagens a produção mais rápidas e robustas.” (G)

3.2.2 Ferramentas utilizadas para gerir o SDLC anteriores à implementação de 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. Anteriormente, eram utilizadas várias ferramentas que são consideradas como válidas numa implementação *DevOps*, como é o caso do orquestrador *Jenkins* para processos de automatização de algumas compilações e instalação de produtos. Para toda a gestão do SDLC, era utilizado o *CA Harvest SCM*² com uma grande customização da ferramenta como refere um entrevistado: “*CA SCM Harvest + grande camada customização JAVA, Shell, SQL.*” (I).

3.2.3 Processos semelhantes a DevOps anteriores à sua implementação

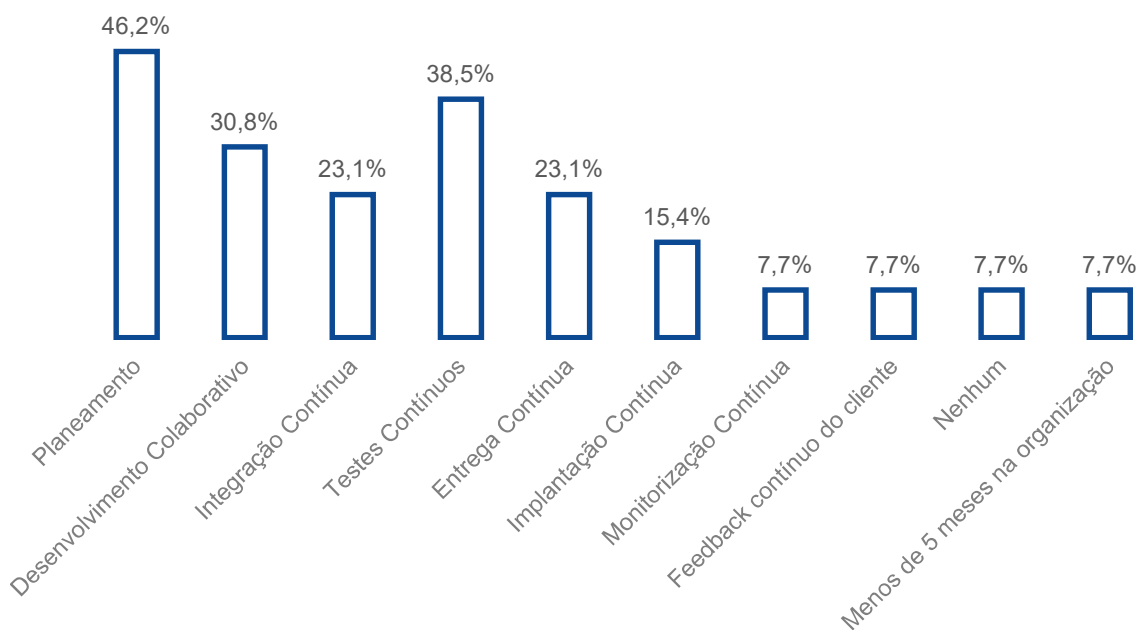


Figura 3.4 Processos da organização semelhantes a um processo inerente à cultura *DevOps*

Antes desta iniciativa de *DevOps* poderiam já existir processos inerentes a esta cultura, como é o caso do planeamento e testes automatizados, referido por 46,2% e 38,5% dos entrevistados, respetivamente. Integração contínua e implantação contínua, apesar de

² O *CA Harvest Software Change Manager (CA Harvest SCM)* providencia recursos orientados aos processos para a gestão de equipas de desenvolvimento de software numa organização (Broadcom, 2019).

serem práticas de *DevOps*, por serem referidos muitas das vezes como processos automáticos, podem suscitar dúvidas, visto que a organização já possuía alguns destes processos automáticos, mas não em forma de *pipeline* automatizada. Apesar de não serem processos de *DevOps*, são processos semelhantes que conferiram uma curva de aprendizagem menor para o uso das práticas enunciadas.

3.3 Implementação de *DevOps*

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

3.3.1 Alternativas à adoção de *DevOps*

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

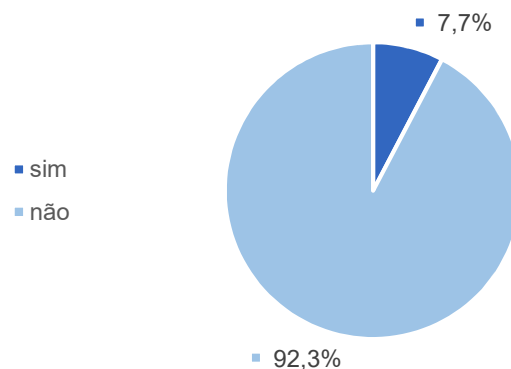


Figura 3.5 Alternativas à adoção de *DevOps*

92,3% dos entrevistados considera que não foi ponderada outra alternativa à adoção da cultura *DevOps*. Apenas 7,7% refere uma, Agile, mas esta não pode ser vista como uma alternativa, visto que se trata de uma metodologia de desenvolvimento de *software*, podendo ser utilizada inclusive na cultura *DevOps*.

3.3.2 Objetivos da implementação de *DevOps*

Os entrevistados referiram na sua maioria 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, automatização (CI/CD) e containerização, o segundo é cultural, novo modelo colaborativo com equipas integradas.” (K)

O gestor das equipas de TI relacionadas com a gestão do SDLC menciona que os objetivos desta implementação são:

“Possibilitar entregar rapidamente funcionalidades ao utilizador final sem comprometer a qualidade e diminuir o esforço de todos os participantes no SDLC.” (L)

Dois dos elementos da equipa de implementação afirmam que:

“Manter um nível de qualidade na entrega de software de forma a prever e resolver qualquer tipo de erros.” (C)

“Agilizar e automatizar o ciclo de vida do software.” (D)

O *scrum master* defende como principal objetivo o acelerar de processos:

“Investigação de novas metodologias, ferramentas, tecnologias e processos de forma a melhorar processos entre 100-2000% no tempo de entrega, diminuindo o tempo de falha e correção sendo detetado preventivamente antes de avançar para ambientes produtivos.” (F)

O elemento M, na qualidade de gestor de projetos refere como objetivo a investigação e utilização das melhores práticas e ferramentas:

“Tem como principal objetivo fazer investigação, desenvolvimento e implementação de modo a evoluir o processo de ciclo de vida da fábrica no paradigma de Continuous Integration/Deployment/Delivery/Testing até à produção, guiando a organização nas best-practices DevOps e tools do mercado.” (M)

3.3.3 Melhores práticas na implementação de DevOps

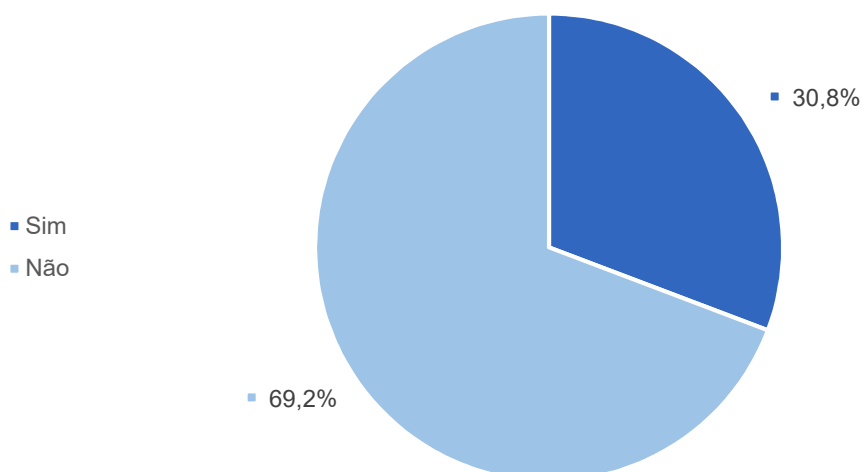


Figura 3.6 Melhores práticas na implementação de DevOps

69,2% dos entrevistados afirmam que não foi seguido nenhum guia (ver Figura 3.6), no entanto, o elemento L refere que:

“Baseado na experiência de implementação de CI/CD em projetos semelhantes, fóruns e eventos de principais fornecedores de tecnologia e eventos da comunidade DevOps, onde a partilha de experiências foi uma grande ajuda na identificação da estratégia a implementar.” (L)

Também ligado às melhores práticas, a opinião do elemento M vai de encontro às práticas da organização, e em como promovê-las garante a alavancagem da organização, no sentido da implementação de *DevOps*:

“Seguimos as boas praticas na organização em sintonia com o que pretendemos melhorar com os novos desafios.” (M)

3.3.4 Participação de entidades externas

Existiu a necessidade da participação de parceiros de negócio neste projeto. Para a implementação de ferramentas de análise estática de código (SCA – *static code analysis*), bem como o fornecimento de licenciamento de software pago. As ferramentas de SCA permitem evidenciar erros de sintaxe, vulnerabilidades e duplicações de código presentes num projeto.

3.3.5 Custo estimado da implementação de *DevOps*

Não pode ser estimado um valor para o custo global do projeto de implementação, pois este contempla várias fases e intervenientes. O Elemento A, da equipa de implementação considera que:

“Em grande parte, como são utilizadas ferramentas open source, as despesas são mais ligadas ao esforço da equipa de implementação.” (A)

O *scrum master* tem a seguinte opinião:

“Para a primeira implementação foram usadas apenas ferramentas free ou community, de forma a demonstrar que o processo funcionava. Com a sua respetiva adoção e migração de projetos para este novo processo, nos vários pontos em que a tecnologia é identificado como "critica" no processo, é promovido o processo para (no mínimo) oferecer suporte 24/7, ou no ideal funcionalidades que apenas são disponibilizadas em versões pagas e o cliente como um todo se encontra maduro para adoção das mesmas.” (F)

3.3.6 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.

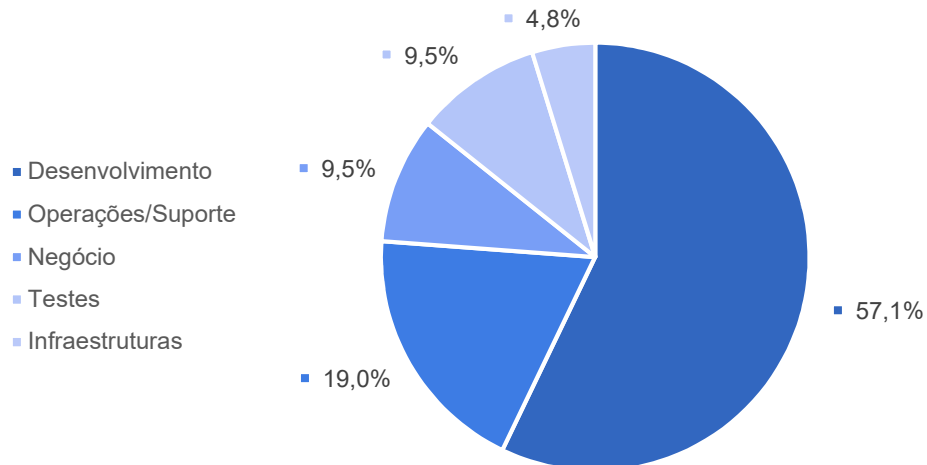


Figura 3.7 Áreas com maior mudança organizacional/cultural

A área onde se verificou maior mudança foi no desenvolvimento, com a maioria dos entrevistados a confirmarem essa mudança:

“Antes a área não tinha necessidade de interagir com as restantes e trabalhavam de forma muito isolada.” (A)

“Tendo o aval de Top-bottom, foi iniciado o processo no "desenvolvimento", sendo depois passado para o processo seguinte de "suporte a ambientes". À medida que o processo amadurece em cada fase, vamos avançando para a esquerda (na parte de planificação) e para a direita na parte de Release Management.” (F)

Na perspetiva do elemento da equipa de testes automatizados:

“O investimento em testes automáticos, leva a uma mudança na organização de investimento em testes manuais. A nível de infraestrutura, com a adoção de cointainers, é libertado o processo burocrático de pedido e criação de ambientes específicos. O desenvolvimento tem regras de entrega para permitir usufruir da pipeline implementada.” (J)

Um elemento da equipa de implementação considera que todas as equipas:

“Passaram a ter de adotar outra cultura em termos de entrega de software e passou a existir uma maior aproximação para deteção e correção de erros.” (E)

3.3.7 Tipologia do processo implementado

A Figura 3.8 apresenta qual a tipologia de processo implementado, segundo os entrevistados.

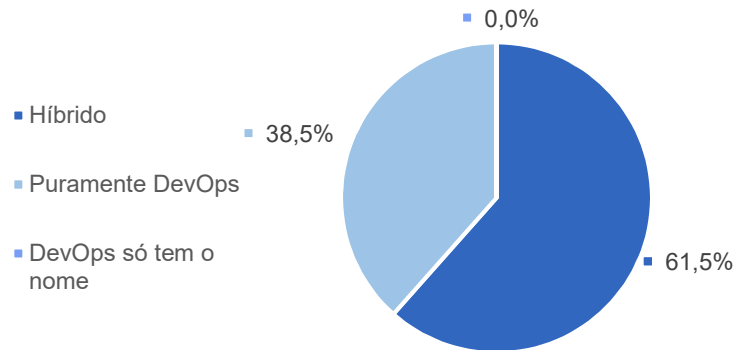


Figura 3.8 Tipologia do processo implementado

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

“Na realidade, é raro uma adoção DevOps ocorrer com sucesso se for “puro DevOps”, relegando completamente o mundo legacy. Com uma adoção híbrida, em que se vai adotando consoante a rapidez da transformação dos projetos para o novo processo DevOps, existe uma maior evangelização de todas equipas envolvidas e uma “suavização” na adoção e demonstração de resultados.” (F)

O *product owner* afirma que:

“O processo de Continuous Integration em implementação é fundamentalmente DevOps.” (K)

Contudo, não é um apenas está prática que garante um processo puramente *DevOps*, pois se não contempla as restantes práticas corretamente implementadas, não pode ser considerado *DevOps* puro.

O elemento L observa que:

“É um processo de DevOps em que a génese do conceito está a ser implementado, no entanto, a responsabilidade não é deixada nas mãos dos developers e sim, é definido e implementado um processo que permita ao desenvolvimento tirar os benefícios de um processo de DevOps.” (L)

3.4 Práticas *DevOps*

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

3.4.1 Práticas implementadas

Através da observação dos processos e ferramentas, e da entrevista com o *scrum master*, foram identificadas as seguintes práticas implementadas pela organização:

- Planeamento;
- Desenvolvimento colaborativo;
- *Continuous Integration*;
- *Continuous Delivery*;
- Testes contínuos;
- Monitorização contínua.

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

3.4.2 Principais fatores influenciadores do sucesso da implementação de *DevOps*

Os entrevistados consideram que os principais fatores influenciadores de sucesso são:

- Espírito de equipa e entreajuda;
- Qualidade da infraestrutura;
- Apoio da gestão de topo;
- Concordância da equipa;
- Abertura para a mudança;
- Experiência;
- Comunicação;
- Definição dos objetivos e planeamento;
- Evangelização;
- Tecnologias que habilitem as práticas;
- Análise de impacto;
- Automatização;
- Tomada de decisão.

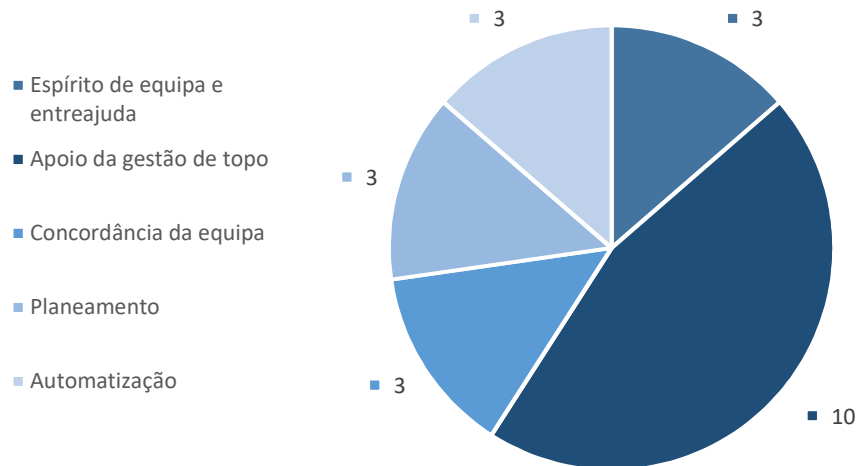


Figura 3.9 Top 5 fatores influenciadores de sucesso

A Figura 3.9 mostra os cinco fatores influenciadores de sucesso mais identificados pelos entrevistados. Destacando o Apoio da gestão de topo, com dez dos entrevistados e salientarem este fator. Os restantes fatores, espírito de equipa e entreaajuda, concordância da equipa, planeamento e automatização foram referidos por três entrevistados cada.

Os participantes referiram que estes fatores se verificam na organização.

3.5 Técnicas e Ferramentas de DevOps

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

3.5.1 Implementação de DevOps

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

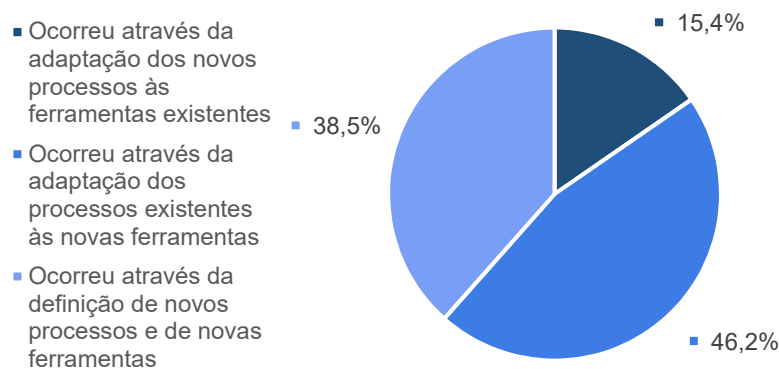


Figura 3.10 Implementação de DevOps

Os processos e ferramentas divergem de organização para organização. Apesar de ser uma cultura *DevOps*, não existe um guia das ferramentas a serem utilizadas. Neste sentido, a organização já dispunha de várias ferramentas que suportam uma cultura *DevOps*. Como, por exemplo, o *Jenkins*, e assim foi apenas necessário adaptar novos processos às ferramentas. Noutro sentido, houve a necessidade de adotar novas ferramentas para os processos já existentes, de forma a garantir que os processos existentes funcionarem nas novas ferramentas. Houve também necessidade de criar processos e adotar novas ferramentas para sustentar as práticas de *DevOps* implementadas.

3.5.2 Ferramentas implementadas

Na Figura 3.11 estão representadas as categorias das ferramentas implementadas.

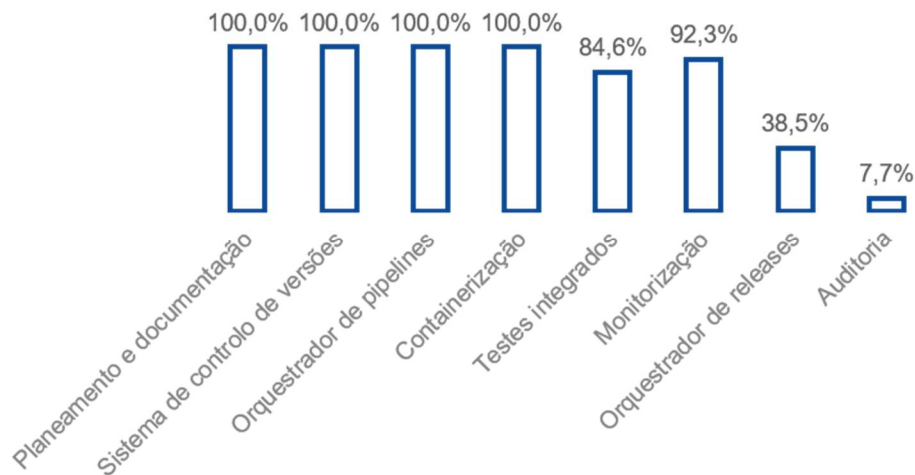


Figura 3.11 Categorias das ferramentas implementadas

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.

As ferramentas implementadas pela organização foram:

- *Jira*
- *Confluence*
- *Gitlab*
- *Jenkins*
- *Docker*
- *Junit*
- *Selenium*
- *Kubernetes*
- *Maven/Gradle*
- *JMeter*
- *Kibana*
- *ElasticSearch*
- *Metricbeat*
- *Artifactory*
- *Logstash*
- *Kiwuan*
- *Codacy*
- *SonarQube*

Estas ferramentas permitiram a adoção da cultura *DevOps* nas categorias acima descritas. O *Jira* permite a gestão de todo o planeamento da organização, desde a organização dos itens de trabalho até à gestão das *releases*. Dentro da mesma temática, o planeamento, o *confluence* permite gerir toda a documentação associada a um projeto. O *Gitlab* é um sistema de controlo de versões adotado pela organização. A infraestrutura aplicacional é suportada pelo *Docker* e *Kubernetes*, sendo o primeiro o motor da containerização e o segundo o orquestrador. O *Junit*, o *Selenium*, e o *JMeter* permitem efetuar vários tipos de testes automatizados. O *Jenkins* permite a orquestração de processos de *Build* e *Deploy*. O *Kibana*, o *Elasticsearch*, o *Metricbeat* e o *Logstash* são uma pilha de ferramentas que funcionam de forma a coletar informação, sobre todo o processo de desenvolvimento e apresentar *dashboards* com a mesma. O *Maven* e o *Gradle* são tecnologias utilizadas para compilação. E por fim, todo o código que é entregue no sistema de controlo de versões é auditado através das ferramentas, *Codacy*, *Kiwuan* e *SonarQube*.

3.5.3 Utilização da *Cloud*

Não foi considerada pela Administração da organização em estudo nenhuma plataforma *Cloud*, apenas foi discutida essa possibilidade por alguns elementos, mas nada que tivesse o impacto suficiente para avançar e ser tido em conta.

3.5.4 Sistema de controlo de versões

A adoção de um sistema de controlo de versões é um elemento chave na adoção da cultura *DevOps*, como mostra a Figura 3.12.

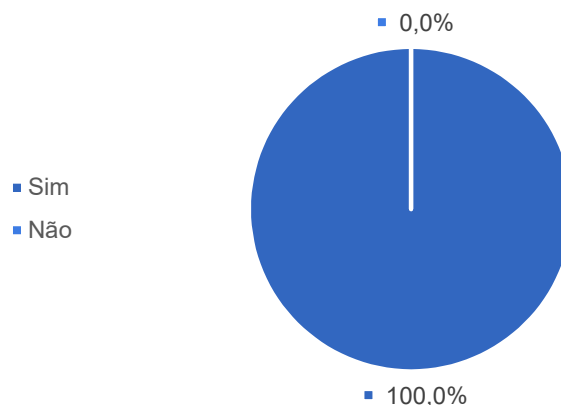


Figura 3.12 Sistema de controlo de versões facilitou a adoção de *DevOps*

Hoje em dia um sistema de controlo de versões é essencial a qualquer organização, independentemente da mesma praticar ou não *DevOps*.

O elemento A afirma que:

“Para qualquer organização, um sistema de controlo de versões é uma mais valia, pois garante que os developers não se atropelam uns aos outros, e que o código está sempre a acessível a todos de uma forma central. Também habilita a prática de Continuous Integration.” (A)

O *scrum master* referiu que este processo veio substituir a ferramenta antiga.

“Aumentou o processo de trabalho colaborativo, retirando-o do Harvest (processo colaborativo restritivo).” (F)

O *product owner* vai ainda mais além referindo que:

“Sem sistema de controlo de versões, não há DevOps (tudo deve ser “as a code”).” (K)

Um dos elementos da equipa de *Configuration Management* considera que:

“Foi a tecnologia que abriu mais a porta a uma abordagem DevOps.” (H)

3.6 Benefícios *DevOps*

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

3.6.1 Principais benefícios da adoção de *DevOps*

Os entrevistados consideram que os principais benefícios da adoção de *DevOps* são:

- Entregas mais rápidas;
- Eficiência;
- União equipas 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*.

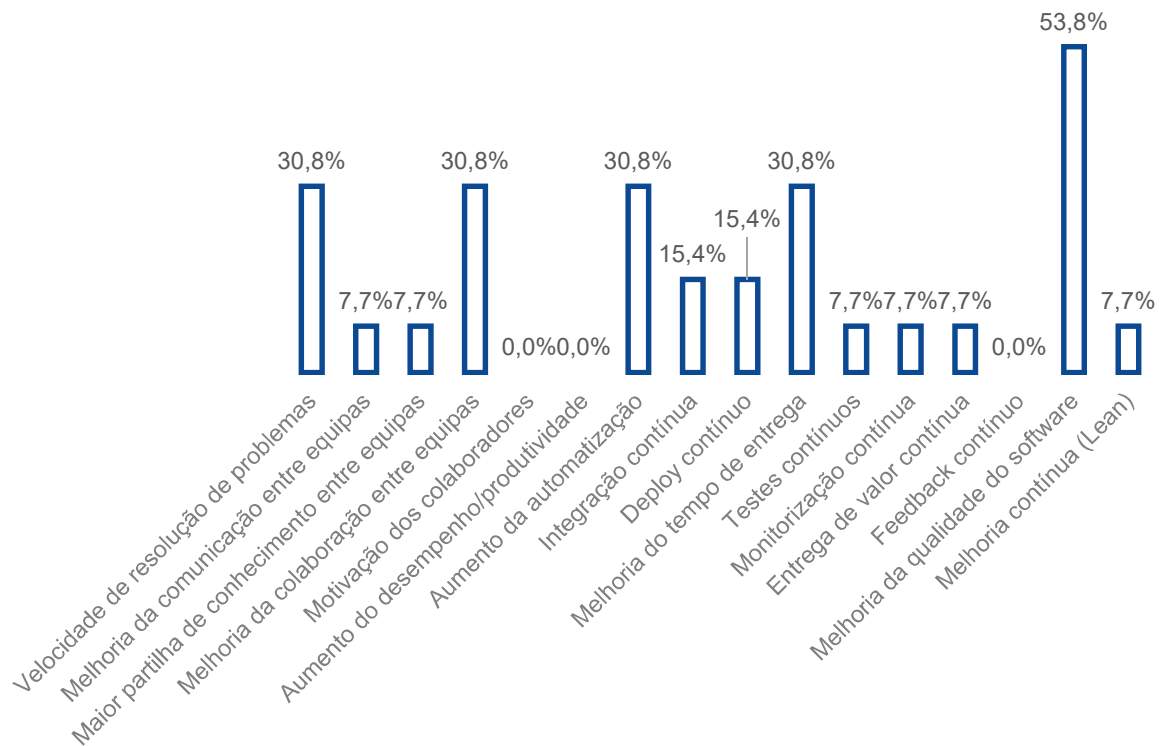


Figura 3.13 Principais benefícios da adoção de DevOps

Como se pode ver na Figura 3.13, 53,8% dos entrevistados indicam que o maior benefício é a melhoria da qualidade de software, seguido do aumento do desempenho e produtividade com 38,5%. Logo atrás, com 30,8%, é considerado o aumento da velocidade de resolução de problemas, melhoria da comunicação entre as equipas, aumento da automatização, e a melhoria do tempo de entrega.

3.6.2 Colaboração entre equipas

Como se pode ver na Figura 3.14, 84,6% dos entrevistados consideram que as equipas passaram a ter uma forma mais colaborativa de trabalhar.

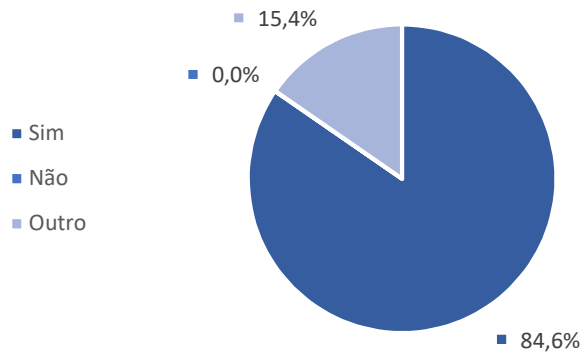


Figura 3.14 As equipas passaram a trabalhar de forma mais colaborativa, após a implementação?

Os restantes 15,4%, que correspondem a dois entrevistados, indicam que não, mas referem que se começam a notar alguns progressos:

“Em algumas ocasiões parece estar mais vincada a abordagem DevOps.” (H)

Por outro lado, um elemento da equipa de implementação refere que:

“A implementação de DevOps, por si só, não produz um ambiente de trabalho mais colaborativo. Para tal, deve existir investimento nas pessoas, na formação das pessoas, e no incentivo à comunicação e participação com outras equipas. Tentando sobretudo promover a colaboração na resolução de temas, ao invés da procura de “descartar” temas para outros.” (E)

3.6.3 Correções de emergência em produção

A Figura 3.15 apresenta a frequência das correções de emergência de sistemas em produção, em que 1 significa “menos frequente” e 7 “mais frequente”. A Figura 3.15 mostra que, os entrevistados consideram, numa grande maioria, que as correções de emergência de sistemas em produção diminuíram com a adoção de DevOps.

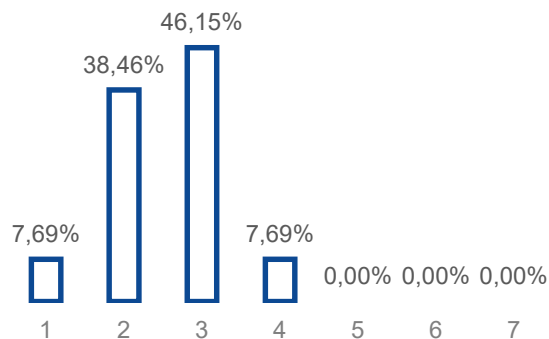


Figura 3.15 Frequência das correções de emergência em produção

3.6.4 Satisfação do cliente e equipas

Os gráficos seguintes (figuras 3.16 a 3.18) apresentam a satisfação do cliente e das equipas, relativamente à adoção da cultura *DevOps*, tendo sido utilizada uma escala *Likert*, em que 1 significa “diminuiu” e 7 “aumentou”.

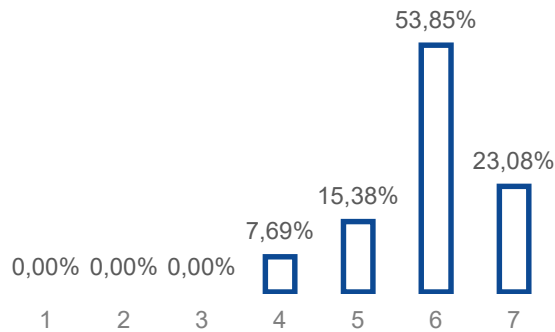


Figura 3.16 Satisfação do cliente

De acordo com os entrevistados (ver Figura 3.16), a satisfação do cliente aumentou.

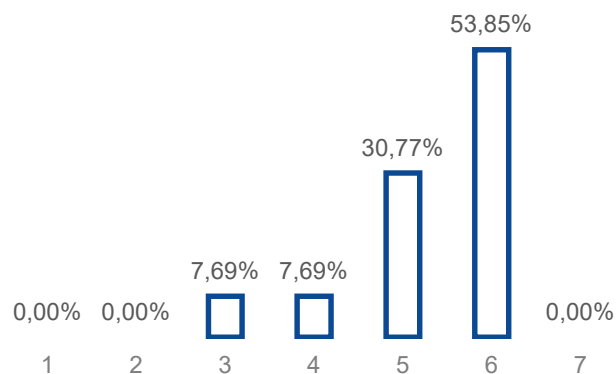


Figura 3.17 Satisfação da equipa de desenvolvimento

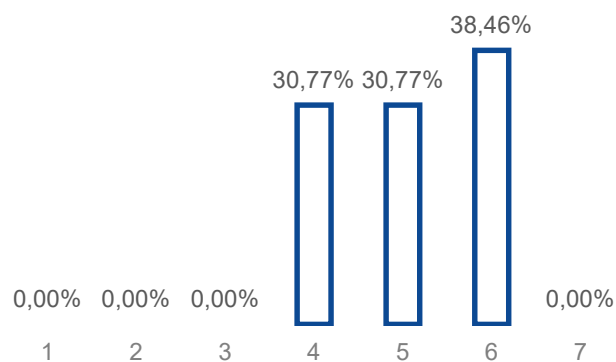


Figura 3.18 Satisfação da equipa de operações

A satisfação das equipas de desenvolvimento e operações aumentou, como mostram as Figura 3.17 e Figura 3.18.

3.6.5 Satisfação da organização

Para aferir a satisfação da organização, também foi utilizada uma escala de Likert, em que 1 significa “diminuiu” e 7 “aumentou”. Também a satisfação da organização como um todo, aumentou desde a Administração até aos desenvolvedores, como mostra a Figura 3.19.

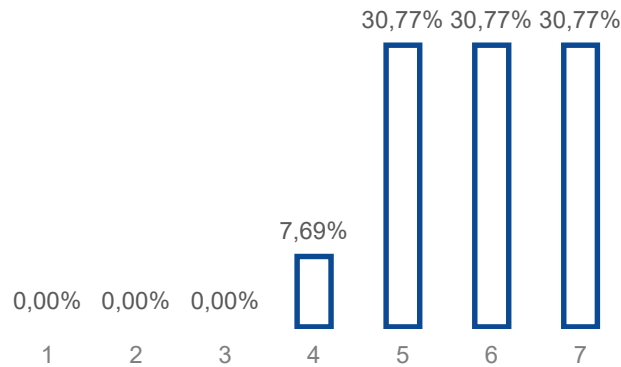


Figura 3.19 Satisfação da organização

3.6.6 Frequência de entregas

A frequência das entregas, avaliada também numa escala de *Likert* de 1 a 7, em que 1 significa “diminuiu” e 7 “aumentou”, também aumentou, mas de forma não muito expressiva, com 69,22% dos entrevistados a considerar que aumentaram (posições 5, 6 e 7 do gráfico apresentado na Figura 3.20). No entanto, 23,08% observam que as entregas se mantiveram constantes e 7,69% que chegaram a diminuir.

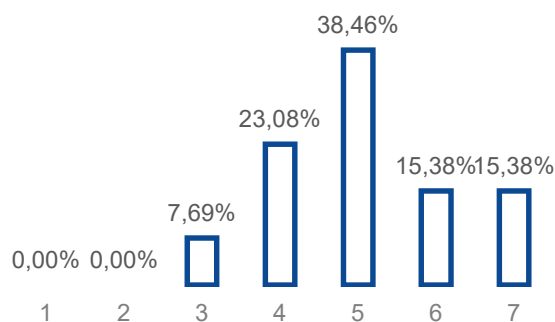


Figura 3.20 Frequência das entregas

3.6.7 Automatização

De forma unívoca, os participantes referem atentam que a automatização aumentou (ver Figura 3.21). Este é um resultado esperado, pois com a implementação das práticas de *DevOps* é necessária sempre esperada a automatização de processos.

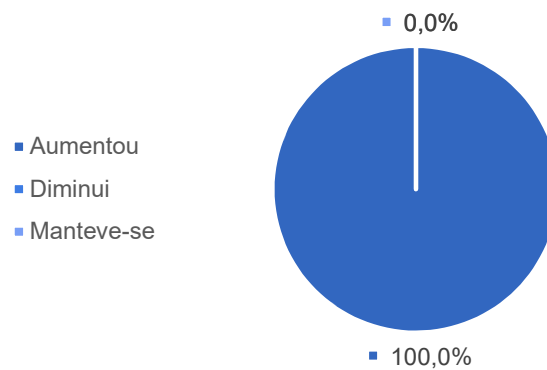


Figura 3.21 Automatização

3.7 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 de gestão de mudança, incluindo a formação das equipas, a alteração das infraestruturas e a reorganização dos departamentos.

3.7.1 Mudança para a cultura *DevOps* como *stakeholder*

A mudança para a cultura *DevOps* verificou-se relevante, como mencionam os participantes. 92,3% afirmam que se revelou pertinente e 7,7% positiva (ver Figura 3.22).

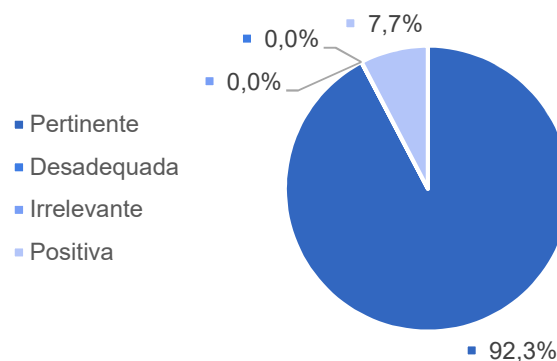


Figura 3.22 Importância da mudança para a cultura *DevOps* na perspectiva de stakeholder

Um elemento da equipa de implementação e um da equipa de *configuration management*, respetivamente, referem que a adoção da cultura veio reduzir os erros que aconteciam em produção:

“Esta implementação melhorou a comunicação e colaboração entre equipas. A médio prazo vai também contribuir para a uniformização de ambientes diminuindo deste modo erros em Produção.” (B)

“Existia necessidade de aumentar a velocidade com que as entregas chegam a produção e de diminuir os erros que chegam a produção.” (G)

O *Scrum master* e um dos elementos da equipa de testes automatizados, respetivamente, mencionam que esta adoção veio refrescar e unificar os processos:

“O processo era muito necessário e tornou-se “refreshing” na alteração de forma de trabalho.” (F)

“Foi uma mudança que permitiu efetuar não só o levantamento de execução de processos como unificar a mesma num todo.” (J)

3.7.2 Formação para a equipa responsável pela implementação de DevOps

A formação para a implementação de *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 como formação oficial, que garante certificação, é o melhor numa perspetiva de ir aprendendo à medida que se vai necessitando e avançando no projeto. O elemento A, da equipa de implementação, e o elemento L, gestor do serviço das equipas relacionadas com a gestão do SDLC mencionam, durante a entrevista, que existiu autoformação.

“A equipa tem o apoio de organização e é treinada, muito com base em self-learning, para conseguir dar várias respostas viáveis à resolução de um problema.” (A)

“Sendo um tema “novo”, a equipa tem feito self-learning e participado em ações de parceiros de modo a se partilhar best-practises.” (M)

3.7.3 Formação das equipas envolvidas

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

“Formação elucidativa dos novos processos e ferramentas a utilizar (fornecida pela própria equipa implementadora).” (I)

3.7.4 Alterações na infraestrutura da organização

A infraestrutura da organização não sofreu alterações, pelo menos a nível de *data-centers*, contudo foi necessário reorganizar a forma como se operava nesse nível.

Foi criado o cluster de *Kubernetes* que suporta a containerização, como referem os elementos K e L, respetivamente:

“Implementação de clusters de Kubernetes e containerização.” (K)

“Criação do cluster Kubernetes.” (L)

Algumas aplicações que dão suporte ao SDLC também tiveram a necessidade de novas máquinas para as suportar, como refere o elemento G:

“Algumas das aplicações precisaram de máquinas novas.” (K)

3.7.5 Alterações estruturais ou reorganização de departamentos da organização

Não existiu reorganização dos departamentos da organização, porém, foi criada a equipa de *DevOps*, que permitiu endereçar os novos desafios para o desenvolvimento do novo SDLC, como referem os elementos K e L:

“Foi criada uma equipa DevOps (desenvolvimento + Operações).” (K)

“Surgiu o equipa de projeto de DevOps para estas implementações.” (L)

Para além da criação desta equipa, as restantes tiveram a necessidade de se adaptar aos novos processos como referem dois elementos:

As equipas adaptaram-se aos novos processos sem necessidades de alterações. (G)

Foi necessário adaptar as pessoas e fazê-las pensar de uma forma diferente. (J)

O *team leader* refere que a alteração estrutural dos departamentos é vital para a continuidade do projeto *DevOps*:

“Atualmente não, mas será vital numa nova fase de aprofundamento da metodologia DevOps.” (I)

3.7.6 Gestão da mudança

A mudança de processos e ferramentas deve ser gradual e adaptativa, para que os intervenientes possam integrar o processo e partilhar opiniões de forma a melhorá-lo. Deve começar-se sempre por um modelo MVP (*Minimum Viable Product*), de forma a

provar que o conceito funciona e é que possível retirar benefícios dos novos processos e ferramentas, como referem os elementos B, G e I:

“A mudança foi gerida migrando as várias aplicações/equipas em pequena escala (1 a 1). 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 equipas.” (B)

“Fez-se a prova de conceito apenas em alguns produtos e só depois se começou a tentar massificar.” (G)

“O processo foi sendo afinado de forma a ser menos disruptivo, mais conciliador e assim com melhores resultados. Não foi de todo o caminho "mais curto".” (I)

3.8 Barreiras DevOps

Nesta secção, são apresentados os resultados relativos às barreiras da adoção da cultura *DevOps*, tendo sido estudadas questões como, a identificação dos principais obstáculos, a resistência à mudança para a nova realidade, nova realidade no que toca à interação entre equipas, e a influência do tamanho da organização.

3.8.1 Principais obstáculos à adoção de DevOps

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

Relativamente à primeira questão, os entrevistados salientam os obstáculos que têm maior impacto para o sucesso do projeto são os referidos na Figura 3.23.

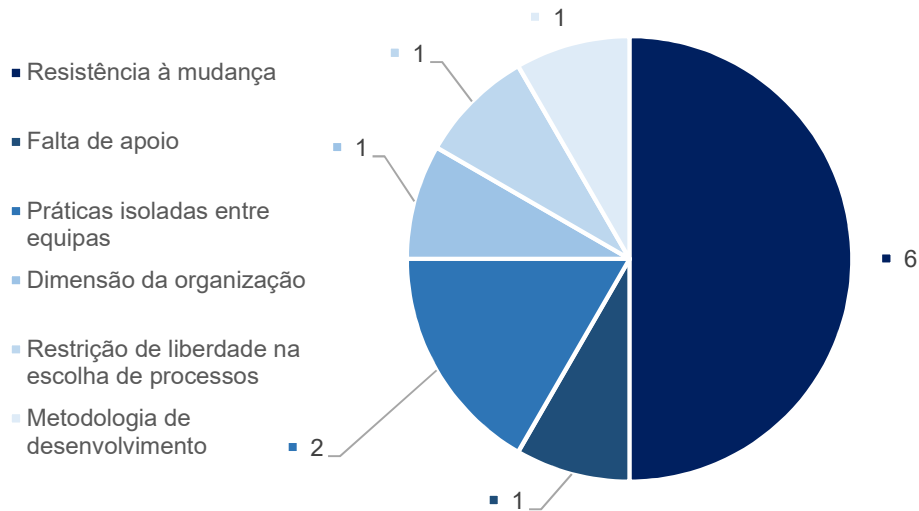


Figura 3.23 Principais obstáculos à adoção de DevOps

- Resistência à mudança;
- Falta de apoio;
- Práticas isoladas entre equipas;
- Dimensão da organização;
- Restrição de liberdade na escolha de processos;
- Metodologia de desenvolvimento.

A resistência à mudança foi considerada pela maioria dos entrevistados como a maior barreira, como refere um dos entrevistados da equipa de implementação:

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

Na segunda pergunta, com base na revisão de literatura efetuada, foi pedido aos entrevistados que selecionassem os mais relevantes, que podem influenciar negativamente o sucesso do projeto, cujos resultados se apresentam na Figura 3.24.

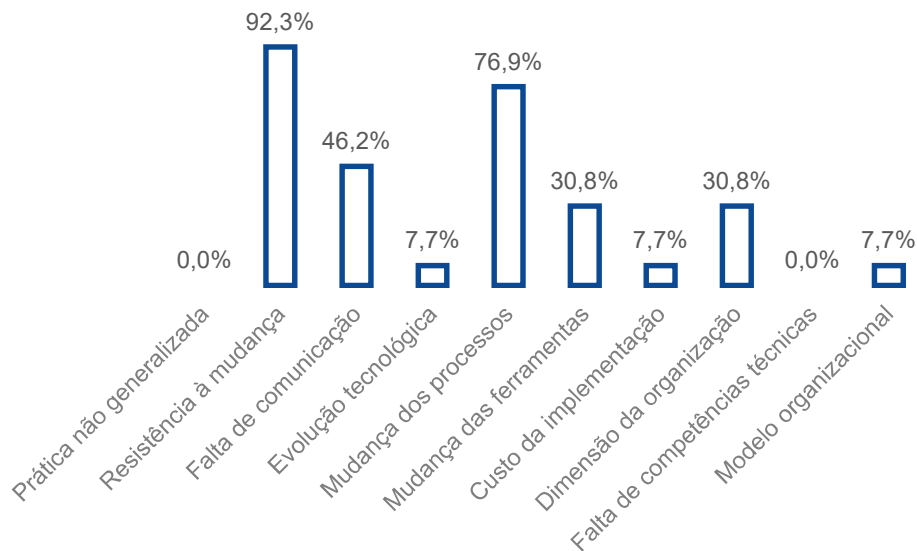


Figura 3.24 Principais obstáculos à adoção de DevOps

Para além da identificação, foi também pedido que justificassem as suas escolhas, tendo obtido as seguintes justificações em relação:

- Ao tamanho da organização e os custos associados como refere o elemento A:

“Quando se trata de uma grande organização, com vários milhares de pessoas, o tamanho é sem dúvida um dos problemas, pois todos têm de estar alinhados para que tudo corra conforme o planeado. O custo das ferramentas, como por exemplo cloud, normalmente está sempre presente.” (A)

- À mudança dos processos, referida pelo elemento B e G:

“Ao mudar os processos, haverá equipas que não estando familiarizadas com os mesmos, terão relutância em adotá-los, porque não os compreendem ou não vêm relevância nestes. Isso tudo pode ser aumentado se não houver comunicação entre todos os stakeholders envolvidos.” (B)

“A alteração de processos que já estão enraizados na empresa é difícil de fazer, pois as equipas têm medo de que os novos processos não funcionem de acordo com as suas necessidades.” (G)

- Ao poder de cada equipa perante a organização a nível hierárquico pode influenciar todo o processo de adoção, como refere o *scrum master*:

“Os obstáculos reduzem-se, no final, ao "poder" de cada equipa, e como estas mudanças podem causar uma diminuição deste "poder", e aumento de outras equipas que possam ser "concorrentes".” (F)

O *product owner* refere que:

“Uma das principais barreiras é a alteração das posições e da organização das equipas que mexe com poder e orçamentos. Segundo, será a resistência à adoção de uma cultura mais colaborativa (responsabilidades partilhadas).” (K)

3.8.2 Resistência à mudança

A mudança acarreta sempre a possibilidade de existir resistência. Neste sentido, 53,8% dos entrevistados referem que existiu resistência à mudança, enquanto que 46,2% considera que não (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 equipas.

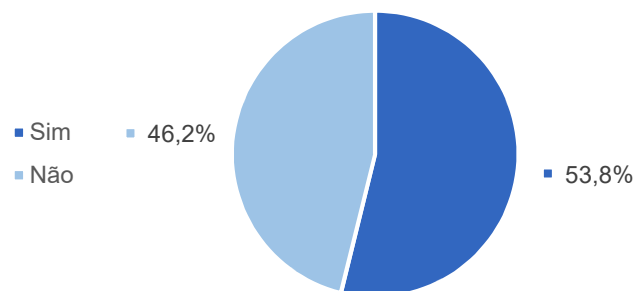


Figura 3.25 Resistência à mudança

Um dos entrevistados descreve que a mudança foi bastante atribulada, referindo que:

“De todas as maneiras possíveis. Só faltou violência física.” (I)

3.8.3 Melhoria da comunicação entre as equipas com a implementação do *DevOps*

Existiu uma clara melhoria de comunicação entre as equipas, como refere a maioria dos entrevistados, 76,9% (ver Figura 3.26). A melhoria da comunicação conduz à consequente quebra de barreiras impostas pelos processos antigos.

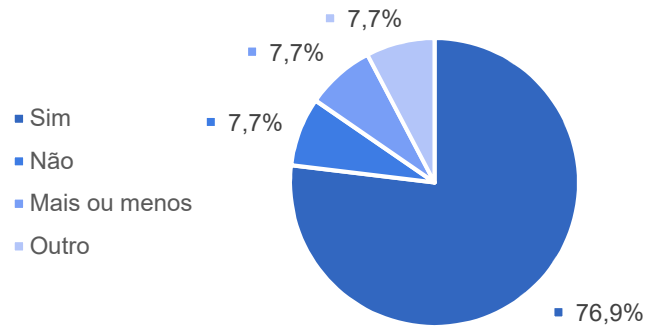


Figura 3.26 Melhoria da comunicação entre equipas com a implementação do DevOps

3.8.4 Dimensão da organização como barreira

Na Figura 3.27 é apresentado o gráfico que representa a dimensão da organização como barreira da adoção da cultura, sendo 1 pouco relevante e 7 muito relevante.

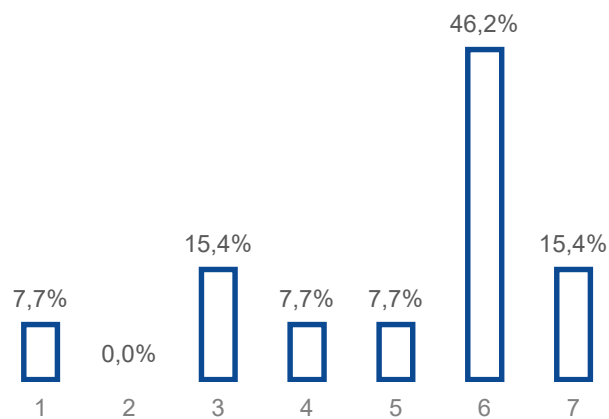


Figura 3.27 Dimensão da organização como barreira

A implementação de *DevOps* teve por base algumas iniciativas que ocorreram em algumas equipas de desenvolvimento, relacionadas com CI, como refere um elemento da equipa de implementação:

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

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

Com outra perspectiva, de resistência à mudança, o *team leader* menciona que existe uma relação de proporcionalidade entre resistência e a dimensão da organização:

“A resistência à mudança de uma organização é proporcional à sua dimensão/complexidade, tal como a inércia é proporcional à massa.” (I)

3.9 Pós-Implementação de 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.

3.9.1 Cumprimento dos objetivos

Na Figura 3.28 é possível verificar que a maioria dos entrevistados consideraram que o *DevOps* cumpriu os objetivos estipulados.

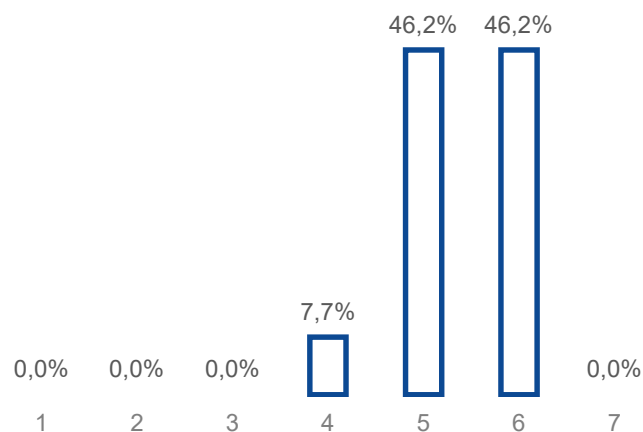


Figura 3.28 Cumprimento dos objetivos

Todos os objetivos que foram propostos com a adoção da cultura *DevOps*, têm-se revelado satisfatórios, como refere um dos entrevistados:

Até à data tudo o que foi proposto, foi conseguido. (J)

Outro entrevistado, refere que ainda é um processo em crescimento, muito devido ao tamanho da organização.

“Devido à grande dimensão da empresa, o processo encontra-se sempre em crescimento, e com este crescimento traz novas equipas e novos processos de adoção”. (F)

3.9.2 Satisfação como stakeholder

A Figura 3.29 indica que 84,6% dos entrevistados revelam-se satisfeitos com os objetivos atingidos.

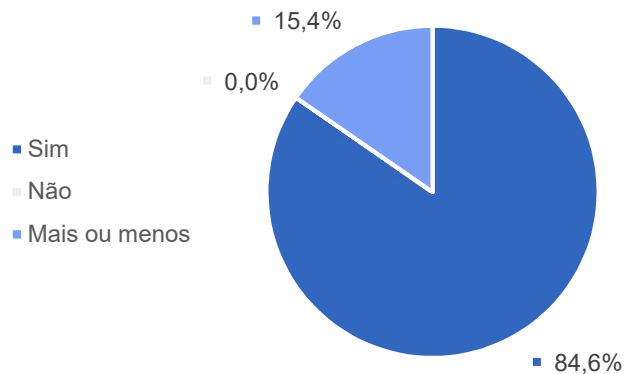


Figura 3.29 Satisfação como stakeholder

Contudo, os restantes 15,4% 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 estanque. Tem margem para evoluir.” (H)

Os elementos A e G referem que as entregas mais rápidas e qualidade do software, são uma das fontes de satisfação:

“Era necessária esta intervenção, para que a qualidade do software fosse maior, de forma a acompanhar o mercado com entregas mais rápidas.” (A)

“Os objetivos de automatização e de entregas mais rápidas estão a ser atingidos.” (G)

Um elemento da equipa de implementação refere que a comunicação e as entregas de código melhoraram:

“A comunicação entre equipas melhorou. A entrega de código tornou-se mais simples.” (B)

O *scrum master* sublinha que:

“Como implementador da metodologia, é bom ver o processo a dar frutos e a resolver os maiores problemas do cliente.” (F)

O gestor de projeto salienta que surgiram melhorias a nível de automatismos e de comunicação:

“Estamos a ter mais automatismos implementados e menos ruído entre pessoas.” (M)

No entanto, o elemento I, *team leader*, admite que ainda falta derrubar algumas barreiras, apesar de se encontrar satisfeito com o atingido, referindo que:

“Satisfeito porque dadas as circunstâncias, foi (e é diariamente) uma vitória. Insatisfeito, porque nos falta derrubar muitas mais barreiras para muitas mais vitórias.” (I)

3.9.3 Critérios para avaliação do sucesso da implementação de *DevOps*

Segundo os entrevistados, os melhores critérios para avaliar o sucesso de uma implementação *DevOps* são os seguintes apresentadas na Figura 3.30:

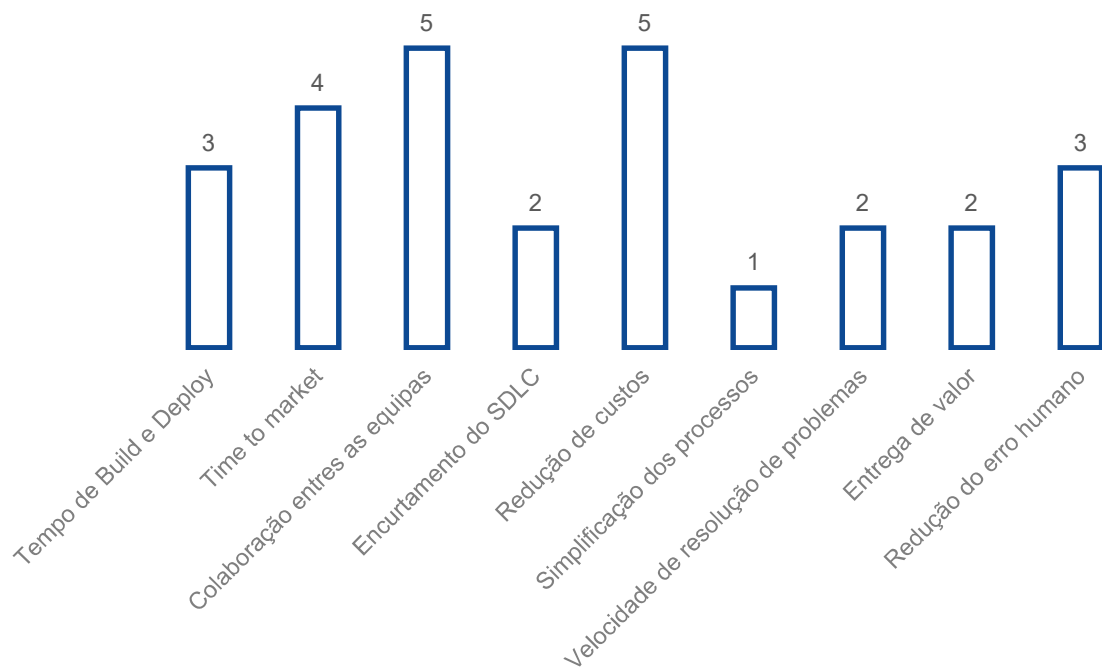


Figura 3.30 Critérios de sucesso de implementação de *DevOps*

- Tempo de *Build e Deploy*;
- *Time to market*;
- Relação da colaboração entres as equipas;
- Encurtamento do SDLC;
- Redução de custos;
- Simplificação dos processos;
- Velocidade de resolução de problemas;
- Entrega de valor;
- Redução do erro humano.

3.9.4 Avaliação formal do sucesso da implementação de 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 estava a ser benéfico para a organização.

3.9.5 Aplicação de DevOps

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

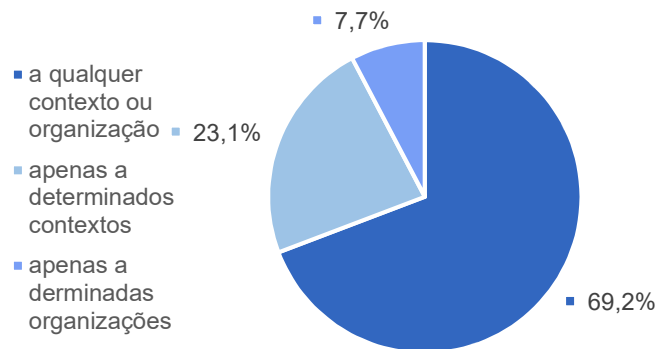


Figura 3.31 Aplicação de DevOps

Os entrevistados são da opinião que *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 elementos da equipa de *Configuration Management* refere que:

“Pode ser visto como um processo universal, porque tem conceitos aplicáveis a qualquer tipo de negócio, contudo está muito estereotipado às TI.” (H)

3.9.6 Sucesso da adoção de DevOps na organização

A maioria dos entrevistados considera que houve sucesso na adoção da cultura, como mostra a Figura 3.32.

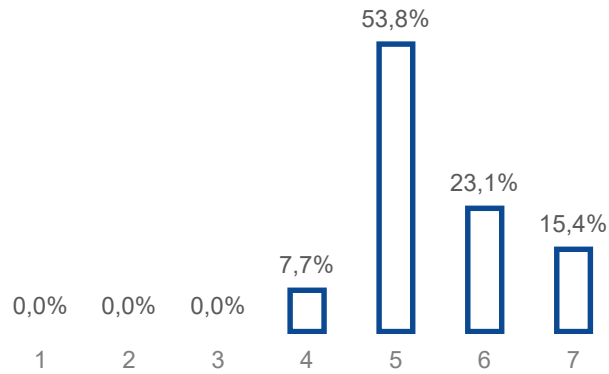


Figura 3.32 Sucesso da adoção de DevOps na organização

Este sucesso começa cada vez mais a contagiar toda a organização e estrutura organizacional, no qual todos querem abraçar esta “aventura”. O *team leader* afirma que:

“Estamos perto de atingir o chamado turning point, em que são os próprios stakeholders a exigir a sua inclusão (tais são as vantagens) no processo de mudança.” (I)

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

3.10 Considerações finais

Este estudo de caso permitiu caracterizar em que ponto estava a organização, antes e depois de adotar a cultura *DevOps*; Compreender as motivações e objetivos da mudança, o pré e o pós-*DevOps* da organização; Verificar o processo adotado dentro das diferentes fases de adoção; Quais as práticas e princípios implementados e quais os processos e ferramentas existentes que foram reutilizados na adoção. O estudo permitiu, ainda, verificar as áreas onde se verifica maior mudança, tanto a nível organizacional como de infraestrutura.

Com análise das entrevistas, é possível compreender se a organização partilha dos mesmo princípios, benefícios e barreias da adoção de *DevOps*, identificados na revisão da literatura.

3.10.1 Adoção do *DevOps*

Como já referido, não existe nem um guião, nem um caminho certo, para a adoção de *DevOps*. Esta adoção deve ser efetuada através de um estudo dos processos que devem sofrer alterações, bem como das ferramentas a serem adotadas. A organização já possuía uma mente aberta no que toca à adoção de novas tecnologias. Neste sentido, como a

organização já utilizava algumas ferramentas relacionadas com *DevOps*, foi possível uma adoção mais rápida.

A adoção de *DevOps* pode incidir sobre duas perspetivas, *top-down* ou *bottom-up*. Como referido anteriormente, não existe uma forma correta da melhor abordagem, ambas são válidas. Deve ser um processo acompanhado por todos os elementos relacionados com o desenvolvimento de *software*. O importante é existir sintonia, entre os envolvidos, do que será alvo da implementação.

Os objetivos foram maioritariamente cumpridos, mas, como se trata de um processo de melhoria contínua, ainda existe algum caminho a ser percorrido. No entanto, a evolução do processo, pode estar condicionada devido à barreira da dimensão da organização. Os entrevistados classificaram a adoção como relevante e positiva, sublinhando os benefícios conseguidos.

Os entrevistados referiram que a adoção de *DevOps* pode ser avaliada com base em alguns critérios, como referido na secção 3.9.3. Apesar de não ter sido levada a cabo uma avaliação formal, conclui-se que a adoção ocorreu com sucesso, com base nas respostas às questões da secção seis (benefícios *DevOps*) e nove (pós-implementação), como por exemplo, o aumento da frequência das entregas, a diminuição das correções em produção, a elevada satisfação da organização e dos *stakeholders*.

3.10.2 Benefícios do *DevOps*

Relativamente aos benefícios evidenciados na Tabela 1.2 e comparando com os benefícios reunidos pelos entrevistados, concluímos que as opiniões destes vão de encontro à revisão da literatura, tendo sido identificados pelos entrevistados alguns benefícios a acrescentar aos da Tabela 1.2, como são os casos do aumento da fiabilidade do processo e a unificação de processos.

Foi também pedido aos entrevistados que, dos benefícios identificados na revisão de literatura, escolhessem os que consideravam mais benéficos. Melhoria da qualidade do *software*, melhoria dos tempos de entrega, aumento do desempenho e produtividade organizacional, maior colaboração e aumento da velocidade de resolução de problemas, foram os mais escolhidos. De um modo geral, os entrevistados confirmam que existiu um aumento da comunicação entre as equipas, bem como a melhoria dos tempos de entrega e resolução de problemas. Referindo também o aumento da satisfação do cliente e equipas, desenvolvimento, operações e equipas relacionadas com o SDLC. A frequência

das entregas também aumentou o que vem salientar o benefício anteriormente referido. De forma geral, a satisfação da organização também aumentou.

A implementação das práticas de *Continuous Integration* e *Continuous Delivery*, que a organização adotou vêm reforçar alguns dos benefícios. Diretamente relacionado, está a resolução de problemas, visto que estas práticas possuem automatismos direcionados à redução do erro humano, diminuindo assim os problemas.

Estes benefícios são obtidos através das práticas implementadas pela organização. O pensamento e o princípio de melhoria contínua estão, posteriormente à adoção, subjacentes à cultura da mesma. Assim, é possível garantir a continuidade do sucesso da organização e uma envolvimento maior no processo de tomada de decisões por todas as equipas relacionadas com o desenvolvimento de software.

3.10.3 Barreiras à adoção do *DevOps*

Existem várias barreiras, referidas anteriormente, que inibem ou atrasam a adoção de *DevOps*. A resistência à mudança foi referida na revisão da literatura como uma barreira. Esta barreira foi também uma das que os entrevistados mais destacaram, e metade dos entrevistados considerou que existiu resistência à mudança por parte dos envolvidos. Com a mudança de processos, é necessário dotar as equipas de conhecimentos para manusear as ferramentas e compreender os procedimentos.

A falta de comunicação é outra significativa barreira, isto porque, por norma, anteriormente à adoção, as equipas trabalham sob a forma de silos. Com a adoção de *DevOps*, a organização passa a ter mais comunicação, como referem os entrevistados. Consequentemente, houve a quebra desta barreira, dando lugar a um ambiente mais colaborativo. Desta forma, é também possível quebrar a barreira do negócio que, apesar de ser difícil devido à dimensão da organização, pode ser ultrapassada através da criação de equipas com vários elementos de diferentes áreas. Levando a uma entrega de valor mais rápida e contínua. Esta barreira continua por quebrar, visto que a organização ainda não atingiu esse nível de maturidade, no entanto, estão a ser levada a cabo algumas iniciativas neste sentido.

As barreiras técnicas podem ser quebradas através da formação da equipa de implementação para aplicação de novas técnicas e processos. Também deve existir formação para as equipas envolvidas, de forma a clarificar a utilização dos processos e ferramentas implementadas. Como já referido na análise de resultados, a equipa de

implementação apenas teve formação com base em *self-learning* e criação de ambientes para testar o que foi aprendido. Desta forma, foi possível identificar o caminho a seguir, bem como as ferramentas a adotar. No entanto, a falta de formação certificada pode levar ao aumento do tempo de implementação e à não utilização total das ferramentas implementadas. A formação das equipas envolvidas foi meramente explicativa a nível de utilização das ferramentas e processos definidos, identificando-se suficiente para o efeito. A organização conseguiu quebrar a maioria das barreiras mesmo antes da adoção desta cultura, tudo através de uma boa visão e decisões estratégicas que culminam com a correta adoção de tecnologias de forma a habilitar as práticas de *DevOps*.

CONCLUSÃO

Este trabalho de projeto teve como objetivo analisar e compreender o processo de adoção de *DevOps* de uma organização, com destaque para os princípios, práticas, benefícios e barreiras.

Para isso realizou-se um estudo de caso, de uma organização de grande dimensão, da área das telecomunicações, que está atualmente a passar por um processo de migração para a cultura *DevOps*.

Através deste estudo de caso, foi possível verificar o “antes” e o “depois” da adoção de *DevOps* na organização, e estudar a sua adoção nas diferentes vertentes.

O processo de desenvolvimento existente até à adoção da cultura *DevOps* era puramente *waterfall*, no qual existia um ciclo de vida estruturalmente bem definido. Desde o desenvolvimento até produção, passando pelos ambientes de testes e pré-produção, o ciclo era gerido por diferentes *gatekeepers*.

A mudança para a cultura *DevOps* está associada a processos de desenvolvimento ágeis, o que garante que os processos estão mais incorporados na organização e são mais automatizados, conferindo assim à organização a utilização de práticas de *Continuous Integration* e *Continuous Delivery*. Estas práticas garantem que os processos são executados de forma mais eficiente, são mais autónomos e com menor intervenção humana, o que leva a uma diminuição de falhas. Foram estabelecidos alguns princípios que foram bem implementados e permitiram a adoção das práticas referidas anteriormente.

Como contributos deste trabalho destaca-se a revisão de literatura efetuada, com enfoque na comparação entre as metodologias tradicionais e as ágeis, identificação das características, benefícios e barreiras do *DevOps*, da qual resultou a escrita do artigo intitulado “*DevOps – fundamentos e perspetivas*”, submetido e aceite na 19.^a Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI’2019).

Ainda como contributos, de referir os resultados do estudo de caso dos quais se destaca, por exemplo, no que toca às práticas, a identificação da *continuous integration* e *continuous deployment*, como mais relevantes. Os benefícios, (ex: a melhoria da qualidade de software) e as barreiras (ex: resistência à mudança). De destacar, ainda,

como principal fator influenciador do sucesso na adoção/implementação de *DevOps*, à semelhança de outras iniciativas nas organizações, o apoio da gestão de topo.

Como principais limitações, destaca-se a dificuldade em agendar e redigir as entrevistas com os participantes e a falta de disponibilidade de tempo (diretamente proporcional ao nível hierárquico dos mesmos).

Em trabalhos futuros, perspectiva-se a repetição deste estudo de caso noutras organizações para se poder cruzar os resultados de diferentes organizações, e obter desta forma resultados ainda mais detalhados.

Como conclusão, de referir que não existe uma definição de *DevOps*, cada organização adota a cultura de forma diferente e relativa ao seu setor de atividade e/ou em relação às práticas que mais a favorecem. *DevOps* visa *releases* mais frequentes, alta automatização e *software* confiável. A adoção da cultura *DevOps*, acelera o alcance dos objetivos definidos pela organização.

REFERÊNCIAS BIBLIOGRÁFICAS

- 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/wp-content/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 de 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., Soto-Acosta, 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

APÊNDICE. GUIÃO

Adoção da cultura DevOps

* Required

Definição DevOps

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 - A iniciativa de promover a ordem de mudança deve partir da administração, incentivando os seus subordinados a continuar essa mudança pelos níveis hierárquicos inferiores. Bottom-Up - A iniciativa de promover a ordem de mudança deve partir da equipa implementadora, aplicando e disseminando os benefícios de uma boa implementação pelos níveis hierárquicos superiores.

Mark only one oval.

- Top-Down
- Bottom-Up
- Other: _____

<https://docs.google.com/forms/d/13zTT7EWeog6lirMkqw3IRJsW-ZGWb8GGVEeq-IjSQ8/edit>

1/21

Adoção da cultura DevOps

* Required

Definição DevOps

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 - A iniciativa de promover a ordem de mudança deve partir da administração, incentivando os seus subordinados a continuar essa mudança pelos níveis hierárquicos inferiores. Bottom-Up - A iniciativa de promover a ordem de mudança deve partir da equipa implementadora, aplicando e disseminando os benefícios de uma boa implementação pelos níveis hierárquicos superiores.

Mark only one oval.

- Top-Down
- Bottom-Up
- Other: _____

3. Por favor fundamente a sua resposta.

Ciclo DevOps



4. Das fases listadas na imagem acima, quais considera serem as que apresentam maior risco na implementação de DevOps? *

Neste contexto, como maior risco subentende-se que as outras fases possam ficar comprometidas.
Check all that apply.

- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor

5. Por favor fundamente a sua resposta.

6. Tendo em conta as práticas de DevOps abaixo enunciadas, relacionadas com o ciclo de desenvolvimento de software, quais considera serem as mais impactantes (positivamente) para a implementação de DevOps? *

Check all that apply.

- Planeamento
- Desenvolvimento Colaborativo (Collaborative Development)
- Integração Contínua (CI - Continuous Integration)
- Testes Automatizados/Contínuos (CT - Continuous Testing)
- Entrega Contínua (CDD - Continuous Delivery)
- Implantação Contínua (CD - Continuous Deployment)
- Monitorização Contínua (Continuous Monitoring)
- Feedback contínuo do cliente e otimização (Continuous customer feedback and optimization)
- Other: _____

7. Por favor fundamente a sua resposta.

Pré-DevOps

8. De forma geral, antes da implementação de DevOps, quais eram as ferramentas utilizadas na sua empresa para gerir o ciclo de vida de desenvolvimento de software (SDLC)? *

9. Já existia, antes da implementação, algum processo da organização semelhante a um processo inerente à cultura DevOps? *

Check all that apply.

- Planeamento
- Desenvolvimento Colaborativo (Collaborative Development)
- Integração Contínua (CI - Continuous Integration)
- Testes Automatizados/Contínuos (CT - Continuous Testing)
- Entrega Contínua (CDD - Continuous Delivery)
- Implantação Contínua (CD - Continuous Deployment)
- Monitorização Contínua (Continuous Monitoring)
- Feedback contínuo do cliente e otimização (Continuous customer feedback and optimization)
- Other: _____

Implementação de DevOps

10. Na sua organização, foram consideradas outras alternativas à adoção de DevOps? *

Mark only one oval.

- Sim
- Não

11. Se sim, quais?

12. Foi seguido algum guia de boas práticas de implementação de DevOps? *

Mark only one oval.

Sim

Não

13. Se sim, qual ou quais?

14. Houve a necessidade de participação de entidades externas à organização para suportar o processo de implementação? *

Mark only one oval.

Sim

Não

15. Por favor fundamente a sua resposta.

16. Qual foi o custo estimado da implementação do DevOps (ex.: custo associado às licenças de novas ferramentas adquiridas, formação, alteração de processos, etc.)?

17. Por favor fundamente a sua resposta.

18. Qual(ais) a(s) área(s) em que se verificou uma maior mudança organizacional/cultural? *

Ex: Desenvolvimento, Operações, Infraestruturas, Negócio...

19. Por favor fundamente a sua resposta.

20. Considerando a literatura, o processo implementado é? *

Mark only one oval.

- Puramente DevOps
- Híbrido
- De DevOps só tem o nome
- Other: _____

21. Por favor fundamente a sua resposta.

Práticas DevOps

22. A prática de Planeamento foi implementada? *

Mark only one oval.

- Sim
- Não
- Other: _____

23. A prática de Desenvolvimento Colaborativo (Collaborative Development) foi implementada? *

Mark only one oval.

- Sim
- Não
- Other: _____

24. A prática de Integração Contínua (CI - Continuous Integration) foi implementada? *

Mark only one oval.

- Sim
 Não
 Other: _____

25. A prática de Testes Contínuos (CT - Continuous Testing) foi implementada? *

Mark only one oval.

- Sim
 Não
 Other: _____

26. A prática de Entrega Contínua (CDD - Continuous Delivery) foi implementada? *

Mark only one oval.

- Sim
 Não
 Other: _____

27. A prática de Implantação Contínua (CD - Continuous Deployment) foi implementada? *

Mark only one oval.

- Sim
 Não
 Other: _____

28. A prática de Monitorização Contínua (Continuous Monitoring) foi implementada? *

Mark only one oval.

- Sim
 Não
 Other: _____

29. A prática de Feedback contínuo do cliente e otimização (Continuous customer feedback and optimization) foi implementada? *

Mark only one oval.

- Sim
- Não
- Other: _____

30. Quais são, na sua opinião, os principais fatores influenciadores do sucesso da implementação de DevOps? Indique pelo menos três fatores. *

Ex: Apoio da gestão de topo

31. Na sua organização verificaram-se todos esses fatores? *

Técnicas e Ferramentas de DevOps

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

Mark only one oval.

- 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 de novas ferramentas
- Other: _____

33. Quais das seguintes ferramentas foram implementadas na sua organização? *

Check all that apply.

- Planeamento e documentação (Ex: Jira, Confluence)
- Sistema de controlo de versões (Ex: Git, SVN, TFVC)
- Orquestrador de pipelines (Ex: Jenkins, Bamboo)
- Containerização (Ex: Docker, Kubernetes, OpenSifht)
- Ferramentas de testes integrados (Ex: JUnit, Selenium)
- Orquestrador de releases (Ex: XL Release, CA Release Automation)
- Monitorização (Ex: Nagios, Zabbix, Kibana, Grafana)
- Other: _____

34. Por favor identifique as ferramentas implementadas.

35. Foi considerada alguma plataforma Cloud para suporte da infraestrutura? *

Mark only one oval.

- Sim
- Não

36. **Se sim, qual ou quais?**

Ex.: Azure, AWS, Google Cloud Platform, outra

37. **Se foi adotado um sistema de controlo de versões, este facilitou a adoção de DevOps?**

Ex: Git, TFVC, outro

Mark only one oval.

Sim

Não

38. **Por favor fundamente a sua resposta.**

Benefícios DevOps

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

40. Selecione os principais benefícios da adoção do DevOps (no máximo 3):

Check all that apply.

- Melhoria da colaboração entre equipas
- Aumento da velocidade de resolução de problemas
- Aumento ao nível da Automatização
- A entrega de valor rápida e contínua
- Feedback contínuo
- Aumento de produtividade e desempenho
- Other: _____

41. Considera que as equipas passaram a trabalhar de forma mais colaborativa, após a implementação?

Mark only one oval.

- Sim
- Não
- Other: _____

42. As correções de emergência em produção passaram a ser? *

Mark only one oval.

	1	2	3	4	5	6	7	
Menos frequentes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Mais frequentes

43. A satisfação do cliente? *

Mark only one oval.

	1	2	3	4	5	6	7	
Diminuiu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Aumentou

44. A satisfação da equipa de desenvolvimento? *

Mark only one oval.

	1	2	3	4	5	6	7	
Diminuiu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Aumentou

45. A satisfação da equipa das operações? *

Mark only one oval.

	1	2	3	4	5	6	7	
Diminuiu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Aumentou

46. A satisfação da sua organização? *

Mark only one oval.

	1	2	3	4	5	6	7	
Diminuiu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Aumentou

47. A frequência de entregas? *

Mark only one oval.

	1	2	3	4	5	6	7	
Diminuiu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Aumentou

48. A automatização:

Mark only one oval.

- Aumentou
- Diminuiu
- Manteve-se
- Other: _____

Gestão da mudança

49. Como stakeholder, considera que a mudança foi? *

Mark only one oval.

- Pertinente
- Desadequada
- Irrelevante
- Other: _____

50. Por favor fundamente a sua resposta.

51. Houve formação para a equipa responsável pela implementação de DevOps? *

Mark only one oval.

- Sim
- Não

52. Se houve formação, por favor descreva em que áreas e a sua importância.

53. Houve formação para as equipas envolvidas na mudança (desenvolvimento, operações, etc.) focada nos novos processos e ferramentas? *

Mark only one oval.

- Sim
 Não

54. Se houve formação, por favor descreva em que áreas e a sua importância.

55. Houve necessidade de alterações na infraestrutura da organização? *

Mark only one oval.

- Sim
 Não

56. Por favor fundamente a sua resposta.

57. Houve necessidade de alterações estruturais ou reorganização de departamentos da organização? *

Mark only one oval.

- Sim
 Não

58. Por favor fundamente a sua resposta.

59. Como foi gerida a mudança? O processo revelou-se adequado?

Barreiras DevOps

60. Quais considera serem os principais obstáculos à adoção de DevOps?

61. **Selecione os principais obstáculos à adoção de DevOps (no máximo 3)? ***

Check all that apply.

- Resistência das equipas de Desenvolvimento
- Resistência das equipas de Operações
- Manutenção das ferramentas
- Custo da implementação
- Other: _____

62. Por favor fundamente a sua resposta.

63. Considera que existiu resistência à mudança por parte dos envolvidos no projeto? *

Mark only one oval.

- Sim
- Não
- Other: _____

64. Se sim, como se manifestou essa resistência?

65. A interação entre as equipas envolvidas no processo de implementação de DevOps levou à melhoria da comunicação e consequente quebra de barreiras existentes entre elas? *

Mark only one oval.

- Sim
- Não
- Other: _____

66. A dimensão da organização foi uma barreira para a adoção de DevOps? *

Mark only one oval.

	1	2	3	4	5	6	7	
Pouco relevante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito relevante

67. Por favor fundamente a sua resposta.

Pós-Implementação de DevOps

68. Os objetivos estipulados com a adoção de DevOps foram atingidos? *

Mark only one oval.

	1	2	3	4	5	6	7	
Não foram atingidos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Foram totalmente atingidos

69. Como stakeholder ficou satisfeito? *

Mark only one oval.

Sim

Não

Other: _____

70. Por favor fundamente a sua resposta.

71. Com base em que critérios se pode/deve avaliar o sucesso da implementação de DevOps? *

Ex: Redução de custos

72. Na sua organização houve uma avaliação formal da implementação do DevOps. Se sim, que critérios foram usados?

73. Considera que DevOps é adequado a *

Mark only one oval.

- qualquer contexto ou organização
- apenas a determinados contextos
- apenas a determinadas organizações
- Other: _____

74. Por favor fundamente a sua resposta.

75. Em geral, como classifica o sucesso da adoção de DevOps na sua organização? *
Mark only one oval.

	1	2	3	4	5	6	7	
Fracasso total	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sucesso total

76. Por favor fundamente a sua resposta. *
