DEVOPS-AN OPEN ACCESS JOURNAL
Open Journal

DevOps-An Open Access Journal

Review

DevOps Adoption: Insights from a Large European Telco

António Trigo[1,2*]; João Varajão[2]; Leandro Sousa[1]

[1]*Coimbra Business School Research Centre, Instituto Superior de Contabilidade e Administração de Coimbra (ISCAC), Polytechnic Institute of Coimbra, Quinta Agrícola – Bencanta, Coimbra, Portugal*

[2]*Centro ALGORITMI, University of Minho, Campus de Azurém, Guimarães, Portugal*

[*]***Corresponding author***

António Trigo

Coimbra Business School Research Centre, Instituto Superior de Contabilidade e Administração de Coimbra (ISCAC), Polytechnic Institute of Coimbra, Quinta Agrícola – Bencanta, Coimbra, Portugal; Centro ALGORITMI, University of Minho, Campus de Azurém, Guimarães, Portugal; E-mail: antonio.trigo@gmail.com

Article information

Received: January 13th, 2023; **Revised:** April 2nd, 2023; **Accepted:** April 29th, 2023; **Published:** May 5th, 2023

Cite this article

Trigo A, Varajão J, Sousa L. DevOps adoption: Insights from a large European Telco. 2023; 2(1). doi: https://doi.org/10.70705/ppp.doaj.2023.v02.i01.pp26-39

ABSTRACT

Information technology professionals often refer to DevOps as a cultural or professional movement that presents a new approach to software delivery through collaboration between the development (Dev) and the operations (Ops) teams. It is an expanding phenomenon, but its adoption in organizations is still at an embryonic stage, requiring research to clarify the benefits, challenges, and barriers to its adoption. With this objective in mind, it was carried out an in-depth study in a large telecommunications (Telco) company that decided to undertake a process of migration to DevOps. The case study involved several stakeholders and covers the DevOps' ex-ante, adoption, and ex-post. Several important facets of DevOps are addressed in this paper, including: 1) practices (continuous delivery and continuous integration were considered the most relevant in the company), 2) benefits (from benefits stand out the improvement in the software quality and faster delivery, with fewer production failures), 3) barriers (the most significant obstacle was the resistance to change, in several dimensions), 4) success factors (the main influencing factors were top management support, implementation process and applied technology); and 5) others aspects (e.g., motivations and tools). Results provide academics and professionals with an integrated view of the condi- tions of DevOps adoption and its outcomes within organizations.

Keywords

DevOps; Adoption; Success factors; Barriers; Benefits; Principles; Lifecycle; Practices; Motivations; Objectives; Processes; Tools; Outcomes; Success; Case study.

INTRODUCTION

Companies focused on software development need to improve their project management prac- tices continuously so that their products/services achieve higher levels of quality and reach the market faster. The main objective is to enhance efficiency, efficacy and satisfy stakeholders (e.g., customers), as these are fundamental for project and product success (Varajão, 2018).

It is in this context that the phenomenon of agile development emerged not many years ago, looking to involve the customer in the work of the software development (Dev) team, which in agile is characterized by short development cycle iterations, continuous releases, and rapidly evolving requirements (Drury-Grogan et al., 2017). In this way, agile practices influence both team motiva- tion (McHugh et al., 2011) and client satisfaction (Nuottila et al., 2016).

With the agile development and the need to frequently put new (partial) versions of software into production (consequently turning operations (Ops) also agile), another phenomenon emerged (Bobrov et al., 2020; Luz et al., 2019): DevOps. The DevOps main goal is to integrate the entire software development life cycle (SDLC) process, from the requirements enactment to putting software into production (Humble & Farley, 2010), through the implementation of a deployment pipeline (Bobrov et al., 2020; Lwakatare et al., 2019). This integration is desirably done through automatic procedures (e.g., related to testing), aiming to ensure that software meets the quality requirements. In addition to the automation of software delivery, DevOps also looks to ensure, through automated processes (such as alarming), that errors detected in software during produc- tion quickly come to Dev teams' attention. Thus, errors can be corrected, and a new version of the software promptly put into production.

For these mechanisms to work, the Dev and Ops teams need to collaborate and support each other. This type of involvement between the members of the different teams makes DevOps be interpreted as a new movement of Information Technology (IT) professionals, in which "sharing" is the basic principle, meaning that everyone works towards the same goal. For DevOps to be successful, it must be part of the organizational culture (Humble & Farley, 2010; Kornilova, 2017; Walls, 2013).

There are several recent studies in the literature about DevOps (Díaz et al., 2018; Luz et al., 2018; Lwakatare et al., 2019; Senapathi et al., 2018), focusing on aspects such as the culture and principles (Luz et al., 2019), automation (Luz et al., 2019; Senapathi et al., 2018), skills and collaborations (Hemon et al., 2020), job satisfaction factors, risk and work conditions (Hemon et al., 2020), or benefits (Bolscher & Daneva, 2019; Senapathi et al., 2018). However, it is still missing an overall picture of the phenomenon, focusing on the conditions occurring before the adoption (e.g., motivations), during the adoption (e.g., practices put in place), and effects of the adoption (e.g., benefits). Consequently, more research is needed to clarify the main aspects of DevOps adoption under an integrated perspective.

To help fill the literature gap, we carried out an in-depth case study of a large telecommunications (Telco) company that decided to migrate to DevOps culture. Our main goal was to produce a holistic overview of the process by observing and interviewing several stakeholders along the DevOps ex- ante, adoption, and ex-post. As a result, fundamental facets of DevOps are addressed in this paper, including motivations, practices, processes, tools, changes, benefits, barriers, and success factors.

Overall, our study identifies, analyses, and integrates a critical mass of research on the adoption of the DevOps culture in organizations together with empirical evidence, offering a solid basis for researchers and professionals interested in the implementation of DevOps.

The paper is organized as follows. The next section summarizes the relevant literature on DevOps adoption. The research method is described next. Then, the key findings are presented, and the results are discussed. Finally, we conclude with implications from this study for practice and research, limitations, and some highlights for further research.

2. Background—DevOps

The most consensual definition found in the literature is that DevOps is a new approach to software delivery that aims at speeding up the delivery of quality software through collaboration between the Dev and the Ops teams – as opposed to the traditional approach where development and operations are separated in their organizational silos (Bass et al., 2015; De França et al., 2016; Jabbari et al., 2018; Lwakatare et al., 2019; Smeds et al., 2015).

Although the word agile might not seem too associated with this approach, the first name given to it was "Agile Infrastructure" (Debois, 2008). DevOps derives from Dev + Ops (Development + Operations), which refers to the unification of teams, transforming the existing silos into a set of teams that focus work on the organization and not on the activity within it, thus linking all the steps that already exist in software development (including delivery).

In classical approaches, development is concerned with the continuous development of new features and correction of problems in production. At the same time, operations are responsible for the provisioning of hardware and software and are concerned with the stability and maintenance of systems above all else. In DevOps, in addition to that concerns which continue to be necessary, the focus is on ensuring an obstacle-free path that allows the automated placing of the developed software in production, supporting the frequent and fast delivery of new software releases (Davis & Daniels, 2016; Kim et al., 2016; Ravichandran et al., 2016).

2.1. Principles

Wills (2010) proposed DevOps' characterization based on four principles: Culture, Automation, Measurement, and Sharing, creating the CAMS acronym (Humble & Molesky, 2011). Later, Jez Humble added the Lean (L) principle to these four principles, changing the acronym to CALMS (Riley, 2014), where:

Culture (C): DevOps brings a cultural change to the organization, which encourages empathy and collaboration between the different teams involved in developing and delivering software, avoiding the organizational silos. The entire organization must be an ecosystem that results in the con- tinuous delivery of value to the end customer (Hamunen, 2016; Kim et al., 2016; Laihonen, 2018; Riley, 2014; Walls, 2013).

Automation (A): the automation of the processes associated with the development and delivery of software is the main task of organizations that want to adopt DevOps, being possible to automate various processes, such as compilation, testing, availability, and installation (build, tests, release and deploy; Hamunen, 2016). On the one hand, automation allows eliminating monotonous tasks in the sense that they are transformed into automatic actions, avoiding human error, and improving the consistency of the infrastructure (Lwakatare et al., 2016). On the other hand, it allows freeing up human resources in the IT area, typically with valuable skills (Trigo et al., 2010), to realize other tasks that create value for the organization (Laihonen, 2018).

Lean (L): Lean methodologies, focused on continuous and systematic improvement, are an important pillar of DevOps (Kim et al., 2016), making the software development and delivery process more robust, effective, and efficient.

Measurement (M): the DevOps approach suggests that the Dev and Ops teams' assessment and control should be done simultaneously, with both teams being awarded according to the same parameters (the common goal is to produce value for the customer). The Dev team's success is measured according to the software in production, using the Ops team's feedback for decision- making on new improvements and features (Lwakatare et al., 2016), thus enabling the creation of more stable applications (Humble & Molesky, 2011).

Sharing (S): the DevOps culture proposes the improvement of the relationship between Dev and Ops teams, with the idea that everyone should collaborate (avoiding the organizational silos) and that the lessons learned by some are shared with all (enabling improvements in culture) (Hamunen, 2016; Humble & Molesky, 2011; Kim et al., 2016; Laihonen, 2018; Riley, 2014).

Although there are other related principles (e.g., customer-centric

action, end-to-end responsibility, continuous improvement, etc.), CALMS synthesizes DevOps' main pillars. This culture allows a hybrid approach, in which some principles can be privileged over others to be possible to adopt the practices gradually. To note also that tools can be adapted to processes or vice versa (Ebert et al., 2016).

2.2. Practices

Several practices need to be taken into account when addressing DevOps, such as Continuous Integration (CI), Continuous Delivery (CDE), and Continuous Deployment (CD) (Lima & Vergilio, 2020; Shahin et al., 2017).

Continuous Integration (CI) is the most common practice of the three: it aims to integrate work in progress so that the entire team has constant feedback from development. It consists of being continuously connected to a code repository to detect changes in each branch. When new changes are detected, a process is launched, where the code is compiled and may be tested through unit tests for a first validation, to confirm that the changes made are viable. A good example is GitHub Webhooks, which enable automated code deployment, validation, and testing (Benson et al., 2019).

Continuous Delivery (CDE) is the practice that ensures that the compiled code is ready to be made available to the organization's various clients. After passing all the quality controls, which aim to reduce the risk of deployment failures, the new change can then be made available to production, where there is a gatekeeper to decide if the changes go into production. This practice requires a manual step in the process.

Continuous Deployment (CD) is the practice that aims to eliminate the manual step mentioned above. Thus, whenever a cycle is successfully completed, new changes go into production. This also means that changes made by one team can be installed regardless of the status of changes made by another team. In this way, it is possible to increase productivity and shorten release cycles.

These three practices, usually called CI/CD practices, are the most common and followed by the DevOps community. If well implemented, they are the path to success in adopting the DevOps culture (Capizzi et al., 2019; Forsgren et al., 2019). In general terms, it can be said that CI/CD leads to the production of new software versions (releases) more quickly and more frequently to eliminate the obstacles that exist between development and production. The customers benefit from this because they get faster and more frequent access to new functionalities.

2.3. Lifecycle

The DevOps lifecycle involves several phases inherent to developing an application (Joost et al., 2017; Kornilova, 2017): plan, code, build, test, release, deploy, operate, and monitor.

Planning is the first phase, followed by codification, which is carried out with the Dev and business teams' participation (that defines the requirements). Next, there is the build phase, where the code is compiled, and some initial tests, such as ex-unit tests, are made. It follows the testing, where unit tests, acceptance, integrity, performance, and non-functional tests are performed. These tests result in the validation of code, addressing scalability, security, availability, among other aspects. Next, the release phase is directly related to the deployment phase. While the release is the act of making available a set of changes, deployment includes the execution of the installation process in an environment. To sum up, in these phases, the code developed is compiled, tested, and installed in the environments (of development, quality, or acceptance) until it reaches the final client (production environment). The build, test, release and deploy phases are usually automated through CI/CD practices (Joost et al., 2017).

The two final phases are operations and monitoring, where teams ensure the integrity and maintenance of applications. Metrics are also defined for the next developments to support evaluation and continuous improvement. This cycle and constant feedback in each phase are expected to deliver software to the end customer with value-added.

2.4. Benefits

There are several types of benefits resulting from DevOps' adoption, which can be translated into technical, cultural, and business benefits (Kim et al., 2016). The technical benefits are related to CI/CD practices and the speed of problem-solving. Cultural benefits contribute to the improvement of communication and the creation of stronger ties between the collaborators of the different teams and, consequently, to have more motivated collaborators who contribute to achieving the organi- zation's mission and vision. The business benefits are related to the final customer and value delivery speed, ensuring greater business stability and more room for innovation. The various benefits are the result of the application of the principles and practices described before. Code versioning, continuous integration, continuous testing, and continuous deployment, all based on continuous monitoring, allow giving way to a fundamental Lean principle, which is a continuous improvement (Kim et al., 2016).

Automating is one of the most significant benefits (Erich et al., 2017) since it reduces errors and frees up human resources (Laihonen, 2018). Fast and continuous value delivery is another sig- nificant benefit for customers (Senapathi et al., 2018; Shahin et al., 2017). The feedback that can be collected from customers is also one gain (Shahin et al., 2017).

Furthermore, DevOps contributes to increased performance and organizational productivity, reduced software development lifecycle costs, improved operational efficiency and effectiveness, and better business alignment and communication between Dev and Ops teams (Erich et al., 2017; Forsgren et al., 2017; Riungu-Kalliosaari et al., 2016; Walls, 2013).

2.5. Barriers

When considering adopting DevOps, it is necessary to ponder the inherent benefits against the costs and problems that arise from the adoption. It is usual for these to happen, requiring the training of the adoption team to minimize risks and create solid contingency plans. It is also needed to take into account the constant evolution of the processes inherent to the DevOps culture since they involve chang-

ing practices, tools, and definitions, and organizations must be prepared to adjust whenever necessary (Riungu-Kalliosaari et al., 2016).

Bass et al. (2015) question the reason for DevOps not being yet a widespread practice. According to these authors, the main problem is resistance to change, which is seen as one of the main challenges/ barriers. In DevOps, team members need to learn new ways of working and lose old habits (Shahin et al., 2017). It is required to prepare and motivate people so that adoption is done with fewer obstacles. The willingness to transform and improve current processes will enable achieving the company's vision more quickly.

Another barrier is the lack of communication, given that companies usually work in silos. Business, Dev teams, and Ops teams are typically concerned with the problems in their areas, not having a holistic view of the company and not worrying about other areas' problems. For this to change, each team must take on new and different responsibilities so that the final objective (continuous delivery of value) is the same for all areas (Riungu-Kalliosaari et al., 2016).

The size of the organization is also seen as a challenge. It is easier to implement the DevOps culture in organizations with few employees and small teams (such as start-ups) than in large multinational organizations (Amaradri & Nutalapati, 2016).

The level of maturity of the organization regarding software development can also be a barrier. This relates, for instance, to the use of agile development practices, automated testing, or existing technical environments. The lower this level, the more costs and efforts are required to establish the processes and the CI/CD pipeline (Elberzhager et al., 2017).

Since DevOps involves changing processes and putting into practice new ones, adopting new technologies and tools, human resources training, among many other aspects, the cost of invest- ment needs to be carefully considered. It might be a significant deterrent to the adoption (Elberzhager et al., 2017).

Finally, the lack of technical skills can also be an obstacle. One of the issues associated with DevOps is the lack of development skills of people in operations and vice versa; that is, to bring together people from different backgrounds to work in the same team, they have to learn about the area they do not master. For example, those connected to operations need to be aware of the different technical environments and respective pipelines. The operations elements need to have knowledge of programming and SDLC. If this learning does not occur, it becomes more compli- cated to create a mindset for different team members to share, communicate and collaborate (Smeds et al., 2015). Table 2 summarizes the barriers found in the literature.

To note that many of the identified barriers are related to organizational inertia dimensions (Besson & Rowe, 2012): negative psychology inertia (denial, fear of learning); socio-cognitive inertia (norms and values at the individual, group, organization, industry, and society levels); socio- technical inertia (technological and socio-technical path dependencies); economic inertia (eco- nomic path dependency); and political inertia (vested interests and alliances).

3. Research framework

DevOps adoption in organizations is non-trivial due to required technical, organizational, and cultural changes (Lwakatare et al., 2019). Thus, to fully understand DevOps, it is necessary to study the complete adoption process, including the ex-ante, adoption, and ex-post adoption (Varajão, 2018). Figure 1 presents the research framework, which is described in the next sections.

3.1. Adoption process

3.1.1. Ex-Ante

Before starting the adoption of DevOps, it is necessary to consider the internal and technological environment of the organization in order to evaluate the costs/benefits arising from the adoption (Elberzhager et al., 2017), being important to identify what practices (Bass, 2018; Jabbari et al., 2018), processes (Lwakatare et al., 2019) and technologies (Ravichandran et al., 2016; Smeds et al., 2015) the organization already has in place that can support the adoption of DevOps, as well as the motivations (Ravichandran et al., 2016) and objectives (Kim et al., 2016) set for the adoption since they will influence the process characteristics (De Feijter et al., 2018; Lwakatare et al., 2019; Muñoz et al., 2019). Therefore, to better understand the DevOps ex-ante, it is important to get answers to the following questions:

Q1. What is the organization's existing environment (status quo) concerning practices, processes, and tools (before DevOps adoption)?
Q2. What are the motivations for the adoption of DevOps? Q3. What are the objectives of the adoption of DevOps?

3.1.2. Adoption

With DevOps, new practices (Bass, 2018; Jabbari et al., 2018) and processes (Lwakatare et al., 2019) are introduced in software production and delivery, supported by new tools (Lwakatare et al., 2019). The adoption of DevOps leads to considerable changes, varying from the infrastruc- tures to the way the team members work (Lima & Vergilio, 2020; Shahin et al., 2017), requiring a careful change management process (Humble & Farley, 2010; Kim et al., 2016). Therefore, to get the answers to the following questions is important to understand the DevOps adoption:

Q4. Which practices are implemented? Q5. Which processes are implemented? Q6. Which tools are adopted?
Q7. How is change management held?

3.1.3. Ex-post

DevOps' adoption typically translates into significant changes (Shropshire & Sweeney, 2017), affecting different stakeholders (Ravichandran et al., 2016). Given its importance, it is fundamental to evaluate the success of the adoption (Hemon et al., 2020) in several dimensions (De Feijter et al., 2018; Lwakatare et al., 2016; Nuottila et al., 2016). Thus, the following questions arise:

Q8. Which changes occurred?
Q9. Are the stakeholders satisfied?
Q10. Is the adoption considered a success?

3.2. Main insights

From the adoption processes, it is possible to draw a set of lessons learned (Tripp & Armstrong, 2018) that, once systematized, can serve as a guide for future implementations (Luz et al., 2018; Smeds et al., 2015). One of the important aspects of studying the adoption process is to realize which of the implemented practices are the most impactful (Luz et al., 2019; Lwakatare et al., 2019). It allows other organizations that wish to adopt DevOps to know where to focus their efforts (Maroukian & Gulliver, 2020). Thus, the following question:

Q11. Which practices of DevOps are most impactful?

To increase the likelihood of success in adopting DevOps in an organization, it is necessary to identify the success factors for its adoption (Rafi et al., 2020; Van Belzen & Kusters, 2019). Thus, the following question:

Q12. What are the success factors of the DevOps adoption?

Since DevOps is a new approach to how the organization works, challenges and barriers to its implementation will inevitably arise (Riungu-Kalliosaari et al., 2016). Thus, the following question:

Q13. What are the barriers to DevOps adoption?

In any endeavor of adopting a new process or technology, the expectation is that the organization benefits from it (Au et al., 2003). Therefore, the following question:

Q14. What are the benefits of DevOps adoption?

4. Method

It was carried out a descriptive case study (Yin, 2013) since the objective of the research was to describe and discuss the DevOps adoption phenomenon within its context. According to Yin (1989), a case study is an empirical study that investigates contemporary phenomena by referring to an event, an entity, an individual, or a unit of analysis. Through the collection of evidence from the real-life context, it is possible to clarify the research questions. The research method is depicted in Figure 2, and the main phases are described next.

The research's main goal was to produce a holistic overview of the process, focusing on DevOps ex-ante, adoption, ex-post, and main insights. Figure 1 presents the research framework that resulted from the first phase (develop theoretical framework), which was based on the literature review.

A single case was chosen in the select case phase: a country's branch of a large European Telco. This company operates in Telco and Audio-visuals segments, offering fixed and mobile solutions, Internet, voice, and data for residential, personal, business, and wholesale markets and pay-tv, broadband, and cinema distribution and exhibition services. At the end of 2019, it had more than 9.7 million subscribers, a turnover of around 1.6 billion euros, and about 2000 employees.

This organization was selected because it decided to undertake a process of migration to DevOps and enabled us to follow the whole process from the very beginning to the end.

The design data collection protocol phase describes how the case study research will be con- ducted and should include the following sections (Yin, 2013): an overview of the case study, field procedures, case study questions, and how the results will be reported. Our study's field proce- dures were mainly of two types: direct observation; and interviews with participants in the phenomenon under study (see, Table 3). The direct observation was carried out at the level of practices, processes, and tools used to obtain insights into the DevOps adoption. The interviews were conducted with a group of people relevant to the case study, representing several stake- holders characterized in Table 3.

This group was selected because of having in-depth knowledge of the entire DevOps phases. It is made up of 13 members of the company, from different areas, to ensure complementary perspec- tives. The DevOps adoption team (interviewees A to F) was responsible for leveraging most DevOps initiatives. It has a complete view of the associated concepts, practices, processes, and tools. From this team, we can highlight the Scrum Master (interviewee F) since he was responsible for defining all the processes and tools to be used and the overall DevOps adoption process. In addition to the Dev team's elements, were also interviewed members from the Ops team (interviewees G, H, and J), which ensure the proper functioning of the SDLC cycle support structure. The person in charge of both teams (Interviewee I) has an overall view of all processes. He is responsible for bringing the automated tests into the deployment pipeline. Hierarchically above this manager, we find the IT department manager responsible for the SDLC (interviewee L) and managing several teams/human resources. He has an overall view of the work of the different IT teams. The product owner (Interview K) is the business- related manager responsible for managing the product backlog and assuring new features' concep- tual integrity. He is also responsible for the deliveries' final quality, being the "voice" of the DevOps team to the Administration. Finally, the Project Manager (Interviewee M) is responsible for managing and controlling products development across different IT teams.

Based on the research framework, a script was developed to support and focus the interviews to the essential. The script included closed-ended questions, multiple-choice, Likert scale, and open questions. It was structured into four sections: DevOps ex-ante; DevOps adoption; DevOps ex-post; Main insights (DevOps most impactful practices; DevOps success factors; DevOps barriers; and DevOps benefits). For more details, please see the appendix.

The data collection phase took place at the selected company between October 2018 and September 2019 through direct observation and individual interviews (the interviewees are identi- fied in

Table 3). During the direct observation of the teams (one of the researchers was always in the field, supporting the DevOps adoption process), which was carried out during all the adoption phases, notes were recorded, and documentation was gathered for analysis. These notes and documentation were important for complementing the theoretical model and defining the inter- views' script.

The interviews were conducted face-to-face in the last phase of the adoption process between July and September 2019. Each interview was recorded with the participant's consent and tran- scribed into text for later quantitative and qualitative (interpretative) analysis. The interviews were, on average, 45 minutes long. Regarding the closed questions, a descriptive statistical analysis (Witte & Witte, 2017) was made. In the case of the open questions, it was carried out a content analysis (mainly conceptual analysis) (Dey, 2003; Zhang & Wildemuth, 2009). The software tool used for supporting the analysis was MAXQDA 2020. Aiming to maximize the validity and reliability, the process was as follows. Firstly, one of the authors performed the initial coding. Secondly, the coding results were discussed by all the authors to reach an agreement. In the third step, and in a blind way, two authors categorized the responses. In the end, these results were again compared and discussed until the authors reached a final agreement.

5. Results—Ex-ante, adoption, and ex-post of DevOps

This section presents the obtained results regarding the DevOps main phases as identified in the theoretical research framework (see, Figure 1): ex-ante; adoption; and ex-post.

5.1. Ex-ante adoption

Regarding status quo (Q1), it was possible to verify, by direct observation, that some of the DevOps practices, processes, and tools already existed in the company before the DevOps adoption, which was also later confirmed by the interviewees. The company used continuous integration and con- tinuous delivery practices, but not in the form of an automated pipeline. Planning and automated testing processes were also in use. Concerning DevOps tools, the use of a central code repository, a compilation orchestrator, and deploy and monitoring tools were identified. One example is the use of Jenkins orchestrator for automation processes of some compilations and product installation. To sum up, before DevOps, the company already had some of the required practices, processes and tools, but was experiencing significant communication problems with both Devs and Ops teams. There were also opportunities for improvement by adopting new good practices.

In general, the main motivations (Q2) to adopt DevOps were: 1) the need to find a solution to communication problems; and 2) barriers among the company's SDLC teams. As one of the interviewees said, the principal motivation was: "To solve communication and delivery problems between teams, as Devs had a specific language that Ops do not fully understand" (C). Another adoption team member stated that the motivations for adopting DevOps were: "To solve communication problems and implement good practices" (B). One of the Configuration Management team members considers that the motivations were related to the preliminary detection of errors, secured from faster production runs: "Faster error detection and faster and more robust deliveries" (G).

When questioned about the objectives (Q3) of the DevOps initiative, the interviewees showed to have a common understanding, as they mostly referred to the speed of delivery and the reduction of failures as primary adoption objectives:

- The product owner stated that the adoption objectives were: "The first objective is technical: automation (CI/CD) and containment; the second is cultural: a new collaborative model with integrated teams" (K);
- The manager of the IT teams related to the SDLC mentioned that the objectives of this adoption were: "Enabling rapid delivery of functionality to the end-user without compromising quality, and reducing the effort of all SDLC participants" (L);
- Two of the members of the adoption team said that the objectives were: "Achieve a high level of quality in software delivery to avoid or resolve any errors" (C); and "Streamline and automate the software life cycle" (D);
- The Scrum Master mentioned as the primary objective the acceleration of processes: "Investigation of new methodologies, tools, technologies, and processes in order to improve processes between 100–2000% in delivery time, decreasing the failure and correction time (and detecting errors preventively before moving to productive environments)" (F);
- The project manager refers as objectives, the study and use of best practices and tools: "The main objective is to study DevOps phenomenon to evolve the life cycle process of the plant in the Continuous Integration/Deployment/Delivery/Testing paradigm to the production, guiding the organization in the DevOps best-practices and tools of the market" (M).

5.2. DevOps adoption

5.2.1. Practices, processes, and tools

Aiming to reach a higher level of automation and management maturity of the SDLC, the following practices (Q4) were implemented by the company: planning; collaborative development; contin- uous integration; continuous testing; continuous delivery; continuous deployment; continuous monitoring; continuous customer feedback; and optimization.

Several processes (Q5) were required regarding documentation management, version control, pipelines orchestration, containerization, integrated testing, monitoring, release orchestration, and auditing. Examples of supporting tools are (Q6) Jira, Confluence, GitLab, Docker, Kubernetes, Junit, Selenium, JMeter, Kibana, Elasticsearch, Metricbeat, Logstash, Maven, Gradle, Codacy, Kiwuan, and SonarQube. Respondents emphasized the version control system implemented, with all respon- dents reporting that it facilitated the adoption of DevOps:

- Developer A stated that: "For any organization, a version control system is a crucial asset because it ensures that developers don't run over each other and that the code is always centrally acces-

sible to all. It also enables the practice of Continuous Integration" (A);

- The Scrum Master said that this process replaced the old tool and that: "It increased the collaborative work process" (F);
- The Product Owner goes even further, emphasizing that: "No version control system, no DevOps (everything should be 'as a code')" (K);
- One of the Configuration Management team members considers that: "It was the technology that opened the door most to a DevOps approach" (H).

Processes and tools differ from organization to organization. Although it is a DevOps culture, there is no guide to the tools to be used. In this sense, before DevOps, the company already had several tools to support a DevOps culture, like Jenkins, for example. It was "only" necessary to define new processes to use some of the existing tools better. There was also a need to create processes and adopt new tools to support the DevOps practices implemented.

Each of the phases listed in Figure 3 represents one SDLC phase, from development planning to production monitoring of the developed systems. These phases have different DevOps adoption risks, depending on the organization in question. Figure 3 shows the phases that respondents consider to have more risk in the DevOps adoption.

Planning is a critical phase, thus having the greatest risk of adoption, as mentioned by almost 70% of the participants. According to two of the adoption team members: "Without good planning, there are sometimes compromising contingencies, resulting in extra costs and delays; Often the planning phase is wrongly undervalued due to possible pressures and urgencies; The devaluation of the previous step can lead to «spaghetti» code" (E).

The project manager added that "Automatic tests have to be reliable so as not to compromise the automatic delivery in production of the features; otherwise, we will only be putting problems into production quickly! Similarly, the release process is essential to ensure that dependencies are met in SDLC passages" (M).

Most respondents (61.5%) described the process as being hybrid. The Scrum Master argues that: "In reality, it is rare for a DevOps adoption to be successful if it is «pure DevOps,» completely forgetting the legacy world. With a hybrid adoption, where one adopts according to the speed of transformation of projects to the new DevOps process, there is a greater «evangelization» of all teams involved and a «smoothing» in the adoption and demonstration of results" (F).

5.2.2. Change management

Change always entails the possibility of resistance. In this sense, 53.8% of those interviewed say that there was resistance to change. One of the interviewees describes that the change has been quite stressful, stating that: "(Was stressful) in every possible way. . . all that was missing was physical violence (smile)" (I).

In DevOps, the change of processes and tools must be gradual and adaptive so that stake- holders can integrate the process and share opinions to improve it. A Minimum Viable Product (MVP) model should be the start to prove that the concept works and that it is possible to benefit from the new processes and tools, as mentioned by some interviewees. In this case (Q7):

- "The change was managed by migrating the various applications/teams gradually. The imple- mentation started with a «simpler» version of DevOps, which increased in complexity as the process matured and was being adopted by more teams" (B);
- "The «concept proof» was made with a few products and only after successful implementation was applied to the other products" (G);
- "The process was fine-tuned to be less disruptive, more conciliatory, and so with better results. It was not followed the «shortest path»" (I).

The Scrum Master and one of the members of the automated testing team, respectively, mention that this adoption has refreshed and unified the processes: "The process was necessary and became «refreshing» for changing the way of working" (F); and "It was a change that made it possible not only to gain awareness on the execution of processes but to unify them as a whole" (J).

DevOps training is essential to maximize value from the adoption process. In this sense, the company opted for a self-learning model. Although not considered official training (leading to certification), this model was considered the best in the perspective of "learning as you go along with the project." Even without formal training, it was crucial to "elucidative guidance on the new processes and tools to be used (provided by the implementation team itself)" (I).

5.3. Ex-post DevOps

5.3.1. Outcomes

After adopting DevOps, the interviewees felt the most changes (Q8) in the development, with 100% mentioning it. According to two elements of the DevOps team, there were changes in how the team members communicate, collaborate, and deliver the new releases: "Before DevOps, the team had no need to interact with the rest of the teams and worked very isolated" (A); "The Dev team had to adopt another culture in terms of software delivery, and there has been a greater approximation to the Ops team to detection and error correction" (E). However, one of the inter- viewees considered that DevOps did not contribute to the improvement of collaboration between teams and stated that: "DevOps implementation by itself does not produce a more collaborative working environment; This requires investment in people, in training, and in encouraging commu- nication and participation with other teams; Above all, it should try to promote collaboration in the solution of problems, rather than seeking to «discard» problems for others" (E). This interviewee, in his response, focuses on the issue of adopting a new culture. It is necessary to change the mindset of the team members to a new way of working. It becomes clear from this case that it is not enough to adopt the new working methodology, being also needed to embrace

it to benefit.

When asked whether there was an increase in automation in their work, all the participants (100%) unambiguously referred that automation has increased with DevOps adoption. From the perspective of an automated testing team element: "There was a shift from manual to automatic testing; At the level of infrastructure, with the adoption of containers, the bureaucratic process of ordering and creating specific environments is freed up; Dev team now must comply with rules that allow them to take advantage of the implemented pipeline" (J). About 30% of the interviewees also refer that there were changes in the operations.

Regarding software quality, performance, and productivity, one parameter that can be evaluated is the number of "emergency corrections." When asked if this kind of correction decreased or increased with DevOps adoption, most participants (92.3%) referred that it declined. One adoption team member and one configuration management team member mention the reduction of errors: "This implementation improved communication and collaboration between teams; In the medium term, it will also contribute to the standardization of environments, thus reducing errors in produc- tion" (B) and "There was the need to increase the speed with which deliveries go into production and to reduce the errors that go into production" (G). When asked whether there has been an increase in software delivery, 69.2% said yes.

The company's infrastructure has not changed at the level of data centers, but it was necessary to reorganize how they operate. "A Kubernetes cluster that supports containerization was created," as the K and L elements state. Some applications that support the SDLC also "required new machines" (G).

There was no reorganization of the company's departments. However, the "DevOps team was created," which allowed addressing the new challenges for developing the new SDLC, as the K and L participants refer. In addition to the creation of this team, the involved human resources needed to adapt to the new processes: "Teams have been adapted to the new processes" (G) and "It was necessary to adapt people and make them think differently" (J). The team Leader states that structural change in departments is vital for the DevOps project's continuity, specifically "for the phase of deepening the DevOps methodology" (I).

5.3.2. Stakeholders satisfaction

Most participants considered that DevOps achieved its objectives (on a scale that goes from 1 (not achieved) to 7 (totally achieved), more than 90% of participants answered 6 or 7).

All objectives that were proposed with the adoption of the DevOps culture have proved satisfac- tory (Q9), as one of the interviewees said: "To date, everything that has been proposed has been achieved" (J). Another interviewee says that it is still a growing process: "Due to the large size of the company, the process is always growing, and with this growth came new teams and new adoption processes" (F).

Most of the interviewees considered themselves satisfied with the results achieved (84.6%). However, the remaining 15.4% report that there is still some way to go, being DevOps adoption an evolutionary process. One of the interviewees states that "It satisfies, but it must be seen as an evolving process, not a closed one; It has a way forward to evolve" (H).

Elements A and G report that faster deliveries and software quality are one of the sources of satisfaction: "This intervention was necessary to make the quality of the software higher in order to keep up with the market with faster deliveries" (A) and "The faster automation and delivery goals are being achieved" (G). One adoption team element mentions that "Communication between teams has improved. Code delivery has become simpler" (B). Scrum Master states that "As a methodology implementer, it's good to see that the process is fruitful and solves the biggest customer problems" (F). The project manager points out that improvements have emerged in terms of automation and communication: "We're having more automatisms implemented and less noise among people" (M).

However, element I, the team leader, admits that there are still some difficulties, noting that: "Satisfied because, given the circumstances, it was (and it is daily) a victory; Dissatisfied, because we need to break down many more barriers to achieve even more victories" (I).

Change has been increasingly advantageous for the entire company. In Figure 4, it is possible to see that most of the stakeholders were satisfied with the DevOps adoption process, with only small differences between them. The Likert scale goes from 1 (not satisfied) to 7 (very satisfied).

5.3.3. Success

Most respondents consider that DevOps adoption was successful (Q10), as shown in Figure 5.

During this process, it was possible to observe that the newly created DevOps team was working more joyfully and collaboratively, achieving better performance than other teams in the company.

Other company elements began to notice it and led them to want to embrace this new "adven- ture." Later in the interview, the team Leader stated that: "We are close to achieving the so-called turning point, in which the stakeholders themselves demand their inclusion in the process of change" (I).

In the company, there was not a formal evaluation of the success of DevOps. However, during adoption, meetings were held to validate whether progress was being beneficial to the company.

6. Discussion—main insights

This section presents the research results regarding the main insights on the adoption process, as identified in the theoretical research framework (see, Figure 1), comprising: impactful practices, success factors, barriers, and benefits. These are also discussed considering the literature.

6.1. Impactful practices

Figure 6 shows the most impactful practices (Q11) of DevOps adoption in the company. Continuous delivery was chosen at the top, with 76.9% of respondents considering it. Right below, with 69.2%, is the continuous integration, and automated testing, and collaborative development, both with 61.5%.

It emerges from the direct observation and the interviews that the CI/CD practices that allow the automation of the SDLC were the most significant since they enabled the Dev team to put the developed work into production more quickly and with fewer errors. It is also worth mentioning the need to create automatic monitoring processes of the software in production associated with continuous monitoring. According to the interviewees:

- "CI/CD practices have the most impact because they can override manual processes and save time in order to achieve faster delivery" (A).
- "DevOps is transversal to all team members; This means, to achieve success, it is important that all stakeholders are always involved in the SDLC; Continuous integration and testing allow for possible bugs to be discovered earlier in the cycle, and continuous delivery leads to more frequent and smaller deliveries with less risk" (B).
- "In projects that do not have maturity in general processes, implementing the processes of CI/ CD/CDE and Collaborative Development, allows to demonstrate at all levels in the client (from management with increased speed and reliability of delivery) how to allow the development focus on development; and, allow operation teams to monitor environments with automated processes" (F).
- "With a joint development, it becomes faster and easier to overcome the difficulties; With automated tests, the probability of error in the execution of the tests is reduced to the ongoing development" (J).

Although the software development teams already used practices similar to those of CI/CD (for example, a version control system), the fact is that with DevOps that they were made official. One of DevOps' adoption's main objectives is to obtain a fully automated CI/CD pipeline (Caprarelli et al., 2019; Garg & Garg, 2019; Senapathi et al., 2018). Thus, our results are consistent with existing literature that points to CI/CD practices' use as the most impacting and beneficial (Caprarelli et al., 2019; Shahin et al., 2017).

6.2. Success factors

Aiming to identify the main success factors (Q12) of DevOps adoption, it was included in the interview script one open question asking up to three factors per participant. Figure 7 shows the identified factors, ordered from the most referenced to the least referenced.

The most mentioned factor for the success of adopting DevOps is the top management's support, with one of the interviewees saying that: "(in this case) there was a lot of top management support for the adoption" (B). The lack of top management protection is pointed out in many studies as one of the greatest challenges for the adoption of the DevOps culture (Hamunen, 2016; Lwakatare, 2017) since, in some cases, DevOps adoption is seen by the top management team as an "experience" rather than a new organizational approach (Hamunen, 2016). Furthermore, the lack of support and participation from top management in software development projects is usually pointed out as a cause for DevOps's adoption failure (Procaccino et al., 2005).

The second factor is related to the implementation process and applied technology (Ahmad et al., 2013). Participants emphasized the importance of the implementation team's expertise and the tools selected for the adoption process. This is in line with current literature since several studies mention the importance of using the DevOps process's right tools (Díaz et al., 2018; Ebert et al., 2016; Lwakatare, 2017).

Regarding the third factor, the participants stressed the importance of a clear definition of objectives and the team's selection to lead the process and manage the project. For instance, the project manager is responsible for managing team members' relationships to benefit the project team as a whole (Leonard & van Zyl, 2014). They also notice the importance of the participation of all team members in the decision-making process.

As for communication and cooperation, which appears as the fourth factor, the participants referred to the collaboration between team members and the spirit of mutual help throughout the adoption process. This success factor, related to two DevOps principles (culture and sharing), is referred to by Luz et al. (2018) as a sine qua non condition for the DevOps adoption. Although crucial for DevOps adoption, it can also be seen as a benefit that emerges from it (Debois, 2008; Hamunen, 2016; Humble & Molesky, 2011; Kim et al., 2016; Laihonen, 2018; Riley, 2014).

The participants also highlighted the importance of the training and competencies of the involved stakeholders. This success factor is associated with the second and third factors pre- sented because the planning, execution, and control of the adoption process can be strongly conditioned by the lack of technical and behavioral capabilities of both who is in charge and who is headed (Senapathi et al., 2018; Smeds et al., 2015). For example, the DevOps implementation team's leader's lack of competence may compromise the definition of a clear plan for adopting DevOps (Toh et al., 2019).

Other mentioned factors are the "ability to manage change" and the "culture of the organiza- tion." In the case of the studied company, these were two aspects carefully addressed. Finally, to mention the monitoring and follow-up of the adoption process, which is fundamental for assessing and evolving the whole process.

6.3. Barriers

Regarding the barriers (Q13) to DevOps adoption, interviewees were asked to identify up to three barriers from the list presented in Table 2, which they considered the most important taking into account their experience. There was also an open question about it. Figure 8 shows the partici- pants' answers, where the most mentioned barriers are resistance to change (92.3%), new processes (to learn and

follow) (76.9%), lack of communication (46.2%), new tools (to learn and use) (30.8%), and size of the organization (30.8%).

Ninety percent of the respondents mentioned that there was resistance to change on the part of those involved. As highlighted by interviewee A: "Resistance to change is usually where most companies are trapped, fear of the unknown continues to haunt organizations when it comes to adopting new methods, processes, and tools" (A). This result was expected, as the processes of adopting new ways of work usually face this barrier. Several DevOps studies in the literature also identify this barrier (Amaradri & Nutalapati, 2016; Bass et al., 2015; Lwakatare, 2017; Shahin et al., 2017).

The second most important barrier is the adoption of new processes (76.9%). This is somewhat related to the previous barrier since new processes require changes. Interviewees B and G state: "When changing processes, there will be teams that are not familiar with them, and will be reluctant to adopt them because they do not understand them or do not see their relevance. This can all be worsened if there is no communication between the involved stakeholders" (B); and "Changing processes that are already rooted in the company is difficult because the teams are afraid that the new processes will not work according to their needs" (G). Like resistance to change, it is also one of the most identified barriers in the literature (Lwakatare, 2017; Lwakatare et al., 2019; Riungu-Kalliosaari et al., 2016), so it is not a surprise to come in second place. It should be noted that one of the main consequences of the addition of the DevOps culture is the change of the software development and operation processes, which impacts how people work (Lwakatare, 2017; Riungu-Kalliosaari et al., 2016).

The lack of communication is another significant barrier (46.2%) because, before DevOps, Dev and Ops teams typically work in silos (Riungu-Kalliosaari et al., 2016). This barrier is also consistent with existing literature; Riungu-Kalliosaari et al. (2016) refer to it as an "inhibiting factor" and Katal et al. (2019) as a "major bottleneck." Although it is considered a barrier to the adoption of DevOps, it should be noted that overcoming it becomes a benefit to the organization.

Fourthly (30.8%), two barriers arise: the use of new tools and the organization's size. As with the new processes' barrier, the difficulty in using new tools can be minimized with proper training. The lack of training may lead to an increase in the adoption process duration and the non-use (in full) of the implemented tools. Regarding the size of the organization, interviewees A and I justified their opinion by saying that: and "When it comes to a large organization, with several thousand people, size is undoubtedly one of the problems, because everyone has to be aligned so that everything goes according to plan" (A); and "The resistance to change of an organization is propor- tional to its dimension/complexity, just as inertia is proportional to mass" (I). This result has been previously documented in the literature by Amaradri and Nutalapati (2016), which state that it is easier to implement the DevOps culture in organizations with few employees and small teams (e.g., a start-up) than in large multinational organizations.

A barrier whose position on the chart comes as a surprise concerning the existing literature (Lwakatare et al., 2019; Smeds et al., 2015) is the one associated with the lack of technical skills. This is perhaps associated with the fact that the interviewees have a strong technical background, which has led them to devalue this item. Nevertheless, it is indirectly valued when they point out the lack of training as a cause for some difficulty in adopting DevOps.

From the open question, two new relevant aspects were identified: the power of each team (Devs and Ops) in the company since it can influence the adoption process ("The obstacles are ultimately reduced to the «power» of each team; the changes can cause a decrease in the «power» of a team and an increase in the power of another team, what may be seen as «competition»" (F)); and the changing of teams' hierarchical position inside the company ("One of the main barriers is related to changing the positions and reorganization of the teams, that mess with extant power and budgets; Second, it may affect the adoption of a more collaborative culture (shared responsibilities)" (K)).

6.4. Benefits

Regarding the benefits (Q14) of DevOps adoption, participants were asked to identify up to three benefits, from the list presented in Table 1, which they considered the most important. As in the question of the barriers, there was also an open question. As shown in Figure 9, 54% of the Improved software quality Improved delivery (continuous and faster delivery) Faster problem-solving interviewees believe that the main benefits were improved software quality and improved delivery (continuous and faster delivery), followed by faster problem-solving (46%). Next, with 38%, the following benefits were identified: Cost reduction; Improved collaboration within teams; Increased automation; and Increased organizational performance and productivity.

DevOps is frequently presented as a new approach to software delivery, through collaboration between Dev and Ops teams, aiming at speeding up the delivery of quality software (Bass et al., 2015; Lwakatare et al., 2019). Thus, improved software quality and improved delivery were, as expected, recognized as important benefits. Practitioners' studies such as the "State of DevOps" by Puppet and Google also refer to these as main benefits, which include improved business outcomes (Forsgren et al., 2017) and improvement of software development and delivery (Forsgren et al., 2019).

The third main benefit is the faster resolution of problems, which was also expectable as it is typically associated with DevOps (Ebert et al., 2016; Erich et al., 2017). This benefit results from a set of practices and tools brought by the DevOps culture, such as better communication between teams, the existence of common infrastructure (CI/CD pipeline), the existence of monitoring tools that detect problems in the applications/systems so that administrators can take the appropriate actions, or even the existence of logging applications that allow isolating problems and identifying solutions (Ebert et al., 2016).

In the fourth position, there are evenly four benefits, also referenced in the literature, when discussing DevOps: Cost reduction (Erich et

al., 2017; Forsgren et al., 2017; Riungu-Kalliosaari et al., 2016; Walls, 2013); Improved collaboration within teams, related to the DevOps principle "sharing" and, according to literature, one of the major impacts of DevOps (Hamunen, 2016; Humble & Molesky, 2011; Kim et al., 2016; Laihonen, 2018; Riley, 2014; Riungu-Kalliosaari et al., 2016); Increased automation (Forsgren et al., 2019; De França et al., 2016; Lwakatare, 2017; Riungu- Kalliosaari et al., 2016); and Increased organizational performance and productivity (Forsgren et al., 2017; Luz et al., 2018; Lwakatare et al., 2019).

From the open question, the participants referred to some benefits regarding the SDLC, such as unification, simplification, and shortening of the SDLC, which add to the current knowledge. Finally, the benefit "more motivated and realized teams," notwithstanding identified in the extant literature (e.g., (Ebert et al., 2016; Lwakatare, 2017)), in our case, was not mentioned by the study's participants.

7. Summary

This paper presents a case study on DevOps' adoption in a large European telecommunications and media company. Given the opportunity to follow the adoption since its beginning, it was possible to characterize the ex-ante, adoption, and ex-post of DevOps; to understand the motiva- tions and objectives of the endeavor; to identify the implemented practices, processes, and tools; to verify the areas where there was the greatest change, both at the infrastructure and organiza- tional levels, and how change was managed; and to identify lessons learned. From direct observa- tion and the perspective of all the stakeholders, it was also possible to verify whether the company shares most of the benefits and barriers of DevOps adoption reported in the literature. A framework with the main insights is presented in Figure 10.

In the ex-ante, although the company already had some of the DevOps practices, processes, and tools, it was experiencing several problems with Devs and Ops teams. Thus, the main motivations and objectives for the DevOps adoption were related to solving the identified problems (e.g., communication problems between teams) and to the willingness to improve the software delivery by implementing best practices.

During the adoption process, taking into account the DevOps inherent principles, it was possible to realize that the software development and delivery became more automated with the imple- mentation of a CI/CD pipeline, which made the whole process leaner. Until then, team members worked in separate silos (development and operations). After adoption, the teams began to share more knowledge, having embraced a new culture of empathy and collaboration focused on continuous improvement.

Concerning new practices (which are associated with principles), in this case, the most relevant are those associated with CI/CD. The company implemented a new version control system, a new deployment pipeline, and automated testing, which is in line with previous studies (Kim et al., 2016; Laihonen, 2018; Lwakatare et al., 2019, 2016). To support these new practices, the company had to reorganize itself. Although it was not necessary to acquire/build new data centers, there was a need for reorganization, specifically through creating a Kubernetes cluster. In addition to these infrastructure changes, there was also the need to reorganize the workforce by creating a new team: a DevOps team.

In the adoption of the new organizational processes, it was possible to identify several success factors and barriers. In this case, and similarly to other adoption processes, top management's support was considered a (critical) success factor (Shao et al., 2016). So, without this support and sponsorship, it would be challenging to succeed. This is an aspect that is linked with another subject that emerges from the literature, which is how top management sees the adoption of DevOps, whether as an "experience" or as an "organizational change process" (Hamunen, 2016). If it is not the latter, the probability of success would be quite low. The planning, communication, and experience of the leading team are also critical success factors for the adoption.

Resistance to change presented itself as the main barrier. This reinforces the importance of top management support and the change management approach. There are other barriers associated with the resistance to change, such as adopting new processes and technologies. As mentioned before, the lack of communication between different team members must be the first obstacle to fall. If this does not happen, the DevOps adoption process may not even start.

An interesting aspect revealed by this study regarding the barriers are the power relations of the teams within the organization, which can affect the adoption of DevOps; furthermore, when carrying out an adoption process of the DevOps culture, it is necessary to choose well which teams will participate so that they have the strength within the organization to lead the process.

Overall, the adoption of DevOps allowed the company to fulfill the objectives, improving com- munication within teams and resulting in an optimized SDLC, leading to the improvement of the software quality, which was one of the major underlying motivations. This case study confirms the benefits of the adoption of the DevOps culture by software Devs and Ops teams, which can be extended to other areas of the organization (Amaradri & Nutalapati, 2016; Debois, 2008; Hamunen, 2016; Humble & Molesky, 2011; Kim et al., 2016; Laihonen, 2018; Luz et al., 2019, 2018; Lwakatare et al., 2019; Riley, 2014; Senapathi et al., 2018).

8. Conclusion

This research contributes to both theory and practice. For theory, our work presents a new frame- work for studying the adoption of DevOps. For practice, the results highlight several important aspects for the successful adoption of DevOps in companies, including processes and tools ("what is required"), success factors ("what influences the success"), barriers ("what may hinder adop- tion"), and benefits ("what benefits may be achieved").

The main limitation is that the research relates to only one case study, which affects its external validity and inhibits generalization. Another limitation relates to study participants. On the one hand, partici-

pants' diversity enabled us to gain richer and complementary insights into the phenomena. On the other hand, it was noticed that some of the participants lack an overall theoretical knowledge about DevOps, being their knowledge "only" the one that results empirically from their daily practice, resulting in different levels of awareness about the DevOps adoption. Moreover, in further studies, it would be important to include the perspectives of other stake- holders, such as the client. Finally, the last limitation is related to the still weak implementation in the studied case regarding continuous monitoring, automatic reporting of failures of systems in production, and auditing, since the bulk of the effort was focused on implementing the deploy- ment pipeline. Although it was possible to realize through direct observation and interviews that some work was done, more was expected in light of DevOps principles and practices.

The proposed framework gives a global vision that integrates the conditions and results of DevOps adoption from the perceptions of the participant actors. However, it does not incorporate the dynamics between elements within DevOps adoption (i.e., between practices, success factors, outcomes, benefits, stakeholders' satisfaction, etc.). As further work, an in-depth analysis of the links between the different elements and conditions of the DevOps lifecycle is recommended.

Furthermore, since our article does not present a systematic literature review, this kind of review is also a proposal for future research.

Our study's results and the mentioned limitations enlighten future work since there is the need to carry out more studies (e.g., replication studies or focused studies (for instance, addressing success factors or barriers, and relating them to the adoption success)) aiming to continue developing the theoretical body because DevOps is still an understudied subject in the research literature.

Funding
This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

REFERENCES

Ahmad, N., Tarek Amer, N., Qutaifan, F., & Alhilali, A. (2013). Technology adoption model and a road map to successful implementation of ITIL. Journal of Enterprise Information Management, 26(5), 553–576. https://doi.org/10.1108/JEIM-07-2013-0041

Amaradri, A. S., & Nutalapati, S. B. (2016). Continuous Integration, Deployment and Testing in DevOps Environment [Master Thesis]. Blekinge Institute of Technology. http://www.diva-portal.se/smash/get/ diva2:1044691/FULLTEXT02.pdf

Au, Y. A., Au, Y. A., & Kauffman, R. J. (2003). What do you know? Rational expectations in information technol- ogy adoption and investment. Journal of Management Information Systems, 20(2), 49–76. https://doi.org/10.1080/07421222.2003.11045764

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.

Bass, L. (2018). The Software Architect and DevOps. IEEE Software, 35(1), 8–10. https://doi.org/10.1109/MS.2017.4541051

Benson, R. M., Munsell, E., Bertrand, N., Baynton, M., Bollig, E. F., & McDonald, J. (2019). A multi-environment HPC-scale puppet infrastructure for compliance and systems automation. Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) - PEARC '19, 1–8. https://doi.org/10.1145/3332186.3332240

Besson, P., & Rowe, F. (2012). Strategizing information systems-enabled organizational transformation: A transdisciplinary review and new directions. The Journal of Strategic Information Systems, 21(2), 103–124. https://doi.org/10.1016/j.jsis.2012.05.001

Bobrov, E., Bucchiarone, A., Capozucca, A., Guelfi, N., Mazzara, M., Naumchev, A., & Safina, L. (2020). DevOps and its philosophy: Education matters! In B. Antonio, D. Dragoni, D. Schahram, L. Patricia, M. Manuel, R. Victor & S. Andrey (Eds.), Microservices (pp. 349–361). Springer International Publishing. https://doi.org/10. 1007/978-3-030-31646-4_14

Bolscher, R., & Daneva, M. (2019). Designing software architecture to support continuous delivery and DevOps: A systematic literature review. 14th International Conference on Software Technologies, ICSOFT 2019, 27–39.

Capizzi, A., Distefano, S., Araújo, L. J. P., Mazzara, M., Ahmad, M., & Bobrov, E. (2019). Anomaly detection in Devops toolchain. International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment (pp.37–51).

Caprarelli, A., Di Nitto, E., & Tamburri, D. A. (2019). Fallacies and pitfalls on the road to DevOps: A longitudinal industrial study. International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment (pp. 200–210).

Davis, J., & Daniels, R. (2016). Effective DevOps. O'Reilly Media. https://www.oreilly.com/library/view/effec tive-devops/9781491926291/

de Feijter, R., Overbeek, S., van Vliet, R., Jagroep, E., & Brinkkemper, S. (2018). DevOps competences and maturity for software producing organizations. Proceedings of International Conference on Business Process Modeling, Development and Support (BPMDS 2018) and of International Conference on Evaluation and Modeling Methods for Systems Analysis and Development (EMMSAD 2018). https://doi.org/10.1007/978-3-319-91704-7_16

de França, B. B. N., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by hearing multiple voices. Proceedings of the 30th Brazilian Symposium on Software Engineering, 53–62.

Debois, P. (2008). Agile infrastructure and operations: How infra-gile are you? Agile 2008 Conference, 202–207.

Dey, I. (2003). Qualitative data analysis: A user friendly guide for social scientists. Routledge.

Díaz, J., Almaraz, R., Pérez, J., & Garbajosa, J. (2018). DevOps in practice. Proceedings of the 19th International Conference on Agile Software Development Companion - XP '18, 1–3. https://doi.org/10.1145/3234152.3234199

Drury-Grogan, M. L., Conboy, K., & Acton, T. (2017). Examining decision characteristics & challenges for agile software development. Journal of Systems and Software, 131(September), 248–265. https://doi.org/ 10.1016/j.jss.2017.06.003

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software, 33(3), 94–100. https://doi.org/ 10.1109/MS.2016.68

Elberzhager, F., Arif, T., Naab, M., Süß, I., & Koban, S. (2017).

From agile development to DevOps: Going towards faster releases at high quality – experiences from an industrial context. Proceedings of the International Conference on Software Quality (SWQD 2017): Software Quality. Complexity and Challenges of Software Engineering in Emerging Technologies. (pp. 33–44). https://doi.org/10.1007/978-3-319-49421-0_3

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

Forsgren, N., Humble, J., Kim, G., Brown, A., & Kersten, N. (2017). State of DevOps Report 2017. DevOps Research and Assessment (DORA). http://www.berry kersten.nl/wp-content/uploads/2017-state-of-devops-report.pdf

Forsgren, N., Smith, D., Humble, J., & Frazelle, J. (2019). Accelerate state of DevOps 2019 report. DevOps Research and Assessment (DORA). https://services.goo gle.com/fh/files/misc/state-of-devops-2019.pdf

Garg, S., & Garg, S. (2019). Automated cloud infrastruc- ture, continuous integration and continuous delivery using docker with robust container security. 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 467–470.

Hamunen, J. (2016). Challenges in Adopting a Devops Approach to Software Development and Operations. Aalto University. http://urn.fi/URN:NBN:fi:aalto- 201609083476

Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). From agile to DevOps: Smart skills and collaborations. Information Systems Frontiers, 22(4), 927–945. https:// doi.org/10.1007/s10796-019-09905-1

Hemon-Hildgen, A., Rowe, F., & Monnier-Senicourt, L. (2020). Orchestrating automation and sharing in DevOps teams: A revelatory case of job satisfaction factors, risk and work conditions. European Journal of Information Systems, 29(5), 474–499. https://doi.org/ 10.1080/0960085X.2020.1782276

Humble, J., & Farley, D. (2010). Continuous delivery.

Pearson Education.

Humble, J., & Molesky, J. (2011). Why enterprises must adopt Devops to enable continuous delivery. Cutter IT Journal, 24(8), 6. https://www.cutter.com/article/ why-enterprises-must-adopt-devops-enable- continuous-delivery-416516

Jabbari, R., Bin Ali, N., Petersen, K., & Tanveer, B. (2018). Towards a benefits dependency network for DevOps based on a systematic literature review. Journal of Software: Evolution and Process, 30(11), e1957. https://doi.org/10.1002/smr.1957

Joost, V., Rigal, S., Wijnholds, G., & Lubsen, Z. (2017). Building software teams: Ten best practices for effec- tive software development. O'Reilly Media.

Katal, A., Bajoria, V., & Dahiya, S. (2019). DevOps: Bridging the gap between development and operations. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 1–7.

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps handbook : How to create world-class agility, reliability, and security in technology organizations. In The DevOps handbook (pp. 480). IT Revolution Press.

Kornilova, I. (2017). DevOps is a culture, not a role! Medium. Com. https://medium.com/@neonrocket/ devops-is-a-culture-not-a-role-be1bed149b0

Laihonen, P. (2018). Adoption of DevOps Practices in the Finnish Software Industry: an Empirical Study [Master's thesis]. Aalto University. https://aaltodoc. aalto.fi/handle/123456789/34400

Leonard, A., & van Zyl, D. (2014). Social relationships in IT project teams: Its role, complexity and the manage- ment thereof. International Journal of Information Systems and Project Management, 2(1), 21–39. https://doi.org/10.12821/ijispm020102

Lima, J. A. P., & Vergilio, S. R. (2020). Test case prioritiza- tion in continuous integration environments:

A systematic mapping study. Information and Software Technology, 121(May), 106268. https://doi. org/10.1016/j.infsof.2020.106268

Luz, W. P., Pinto, G., & Bonifácio, R. (2018). Building a collaborative culture. Proceedings of the 12th ACM/ IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '18, 1–10. https://doi.org/10.1145/3239235.3240299

Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. Journal of Systems and Software, 157 (November), 110384. https://doi.org/10.1016/j.jss.2019.07.083

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). Relationship of DevOps to agile, lean and continuous deployment. In Proceedings of International Conference on Product- Focused Software Process Improvement (PROFES 2016). https://doi.org/10.1007/978-3-319-49094-6_27

Lwakatare, L. E. (2017). DevOps adoption and implemen- tation in software development practice : concept, practices, benefits and challenges [Doctoral disserta- tion]. University of Oulu. jultika.oulu.fi/files/ isbn9789526217116.pdf

Lwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., Kuvaja, P., Mikkonen, T., Oivo, M., & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. Information and Software Technology, 114(October), 217–230. https://doi.org/10.1016/j.infsof.2019.06.010

Maroukian, K., & Gulliver, S. (2020). Leading DevOps practice and principle adoption. 9th International Conference on Information Technology Convergence and Services (ITCSE 2020), 41–56. https://doi.org/10. 5121/csit.2020.100504

McHugh, O., Conboy, K., & Lang, M. (2011). Using agile practices to influence motivation within IT project teams. Scandinavian Journal of Information Systems, 23(2), 59–84. http://hdl.handle.net/10379/3415

Muñoz, M., Negrete, M., & Mejía, J. (2019). Proposal to avoid issues in the DevOps implementation: A systematic literature review. In: Rocha, Á., Adeli, H., Reis, L., Costanzo, S. (Eds.), New Knowledge in Information Systems and Technologies, World-

CIST'19 2019. Advances in Intelligent Systems and Computing (pp. 666–677). Springer, Cham. https:// doi.org/10.1007/978-3-030-16181-1_63

Nuottila, J., Aaltonen, K., Kujala, J., Nuottila, J., Aaltonen, K., & 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

Procaccino, J. D., Verner, J. M., Darter, M. E., & Amadio, W. J. (2005). Toward predicting software development success from the perspective of practi- tioners: An exploratory Bayesian model. Journal of Information Technology, 20(3), 187–200. https://doi. org/10.1057/palgrave.jit.2000044

Rafi, S., Yu, W., & Akbar, M. A. (2020). RMDevOps. Proceedings of the Evaluation and Assessment in Software Engineering, 413–418. https://doi.org/10. 1145/3383219.3383278

Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). Practical DevOps. In DevOps for digital leaders. Berkeley, CA: CA Press. https://doi.org/10.1007/978- 1-4842-1842-6

Riley, C. (2014). How to Keep CALMS and Release More! Rapid7 Blog. 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 adop- tion benefits and challenges in practice: A case study. In Proceedings of International Conference on Product-Focused Software Process Improvement (PROFES 2016). https://doi.org/10.1007/978-3-319- 49094-6_44

Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps capabilities, practices, and challenges: Insights from a case study. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 (EASE'18). https://doi.org/ 10.1145/3210459.3210465

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(March), 3909–3943. https:// doi.org/10.1109/ACCESS.2017.2685629

Shao, Z., Feng, Y., & Hu, Q. (2016). Effectiveness of top management support in enterprise systems success: A contingency perspective of fit between leadership style and system lifecycle. European Journal of Information Systems, 25(2), 131–153. https://doi.org/ 10.1057/ejis.2015.6

Shropshire, J., & Sweeney, B. (2017). On devops and workforce morale. Proceedings of the Southern Association for Information Systems Conference, St. Simons Island. http://aisel.aisnet.org/sais2017/ 12%0Ahttps://aisel.aisnet.org/cgi/viewcontent.cgi? article=1011&context=sais2017

Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: A definition and perceived adoption impediments. In C. Lassenius, T. Dingsøyr, & M. Paasivaara Eds., Agile processes in software engineering and extreme programming (Vol. 212, pp. 166–177). Springer International Publishing. https://doi.org/10.1007/978-3-319-18612-2

Toh, M. Z., Sahibuddin, S., & Mahrin, M. N. (2019). Adoption issues in DevOps from the perspective of continuous delivery pipeline. Proceedings of the 2019 8th International Conference on Software and Computer Applications, 173–177.

Trigo, A., Varajão, J., Soto-Acosta, P., Barroso, J., Molina- Castillo, F. J., & Gonzalvez-Gallego, N. (2010). IT Professionals: An Iberian Snapshot. International Journal of Human Capital and Information Technology Professionals, 1(1), 61–75. https://doi.org/ 10.4018/jhcitp.2010091105

Tripp, J. F., & Armstrong, D. J. (2018). Agile methodolo- gies: Organizational adoption motives, tailoring, and performance. Journal of Computer Information Systems, 58(2), 170–179. https://doi.org/10.1080/08874417.2016.1220240

Van Belzen, M., & Kusters, R. (2019). Critical success factors of continuous practices in a DevOps context. Proceedings of Information Systems Development: Information Systems Beyond 2020 (ISD2019 Proceedings). Toulon, France: ISEN Yncréa Méditerranée.

Varajão, J. (2018). The many facets of information sys- tems (+projects) success. International Journal of Information Systems and Project Management, 6(4), 5–13. https://doi.org/10.12821/ijispm060401

Walls, M. (2013). Building a DevOps culture. O´Reilly. Wills, J. (2010). What devops means to me. CHEF BLOG. https://blog.chef.io/2010/07/16/what-devops-means- to-me/

Witte, R. S., & Witte, J. S. (2017). Statistics. Wiley.

Yin, R. K. (1989). Case study research. Sage publications. Yin, R. K. (2013). Case study research: Design and methods. Sage publications.

Zhang, Y., & Wildemuth, B. M. (2009). Qualitative ana- lysis of content. In B. M. Wildemuth (Ed.), Applications of social research methods to ques- tions in information and library science (pp. 1–12). Libraries Unlimited.