ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2023-04-03

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Faustino, J, Adriano, D., Amaro, R. M. D., Pereira, R. & Silva, M. M. (2022). DevOps benefits: A systematic literature review . Software: Practice and Experience. 52 (9), 1905-1926

Further information on publisher's website:
10.1002/spe.3096

Publisher's copyright statement:
This is the peer reviewed version of the following article: Faustino, J, Adriano, D., Amaro, R. M. D., Pereira, R. & Silva, M. M. (2022). DevOps benefits: A systematic literature review . Software: Practice and Experience. 52 (9), 1905-1926, which has been published in final form at https://dx.doi.org/10.1002/spe.3096. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

1. Introduction

With recent technological advances, Information Technology (IT) departments have taken an increasingly strategic role in organizations [1] given the importance of IT in helping the accomplishment of business objectives [2]. Several disciplines like IT Governance and IT Service Management (ITSM) have built mechanisms and processes so that both IT and business can be aligned in terms of aims and expectations, helping organizations satisfy their objectives [3–5].

Just as organizations' strategic view has evolved, software development lifecycles (SDLC) have also matured to satisfy current demands. To face the great changes observed in the modern-day markets, businesses need to have greater speed and flexibility. This translates to challenges for IT departments worldwide [6]. As stated, the SDLC has been evolving, no longer strictly focusing on the performance of its own processes, as seen on traditional software development methodologies like Waterfall [7,8], but on the iterations and relationships between the intervenient on the SDLC process and the value that the software can bring to the business [9]. These kinds of software development methodologies are considered agile methodologies and follow the "Agile Manifesto" [10]. Even though Business and IT Development are brought closer, a gap is still observed within the IT Department's Development and Operations teams [11]. The major issue between these two teams are the different objectives for each team, IT Development team is focused on delivering new features or products, while the IT operators are focused on the stability [12]. It's believed by introducing changes on systems would lead to instability [13]. however due to the constant market changes, IT of the organisations need to evolve into new functionalities [14]. A DevOps culture has emerged to address this gap. The DevOps word itself comes from the junction of two other words: Development and Operations [15].

DevOps has been adopted across the globe and new research articles flourish. Several studies have reported practices, benefits, and challenges however not always in a structured, clear, and concise way [16–18]. In literature, one can find studies that synthesize DevOps practices, as for example, Deployment Automation [19–21]; however, it presently lacks research that specifically synthesize its benefits and challenges, guiding professionals in what they may expect during and after DevOps practice implementation [22]. Lack of willingness to share can be a challenge to DevOps implementation [23].

Being a contemporary topic, with both theoretical and empirical studies found in literature, this research aims to synthesize the benefits organizations may expect with DevOps implementation and how to achieve them. Being said, by synthetizing the DevOps benefits, this research also provides which problems organisations faced before the DevOps adoption and what was the benefit achieved after that. This will help organisations to know what problems could be fixed by the DevOps adoption. The adopted research methodology will include two Systematic Literature Reviews (SLR).

2. Research Background

The term DevOps started to be researched after Patrick Debois introduced it at a conference entitled "Agile Infrastructure and Operations" in 2008 [24]. A DevOps culture aims to bridge the gap between IT Development and IT Operations, who support applications after they are delivered to production [25]. The focus of DevOps is on improving communication, collaboration, and synergy of IT teams [26,27], enabling the continuous development and enhancement of applications to meet both market changes and the dynamic needs of the business [28,29].

In order to achieve said objectives, DevOps builds a foundation in the following areas: Culture, Automation, Lean, Measurement and Sharing [6]. By looking at Wiedemann et all's [6] work, one may note that of the perspectives presented above, *people* play an important role for Culture and Sharing. Willingness to share is needed, allowing for colleagues and team members to learn and improve their knowledge and experiences. On automation and measurement, one can state that technological tools are the main factor; tools that are used to improve performance, automating what is being done manually, removing the element of human error, and be used to measure tasks and find improvements [30]. Process optimization is a focal point for Lean methodologies. They are used in DevOps to identify opportunities for process enhancement, leveraging feedback loops between a its main actors. In later studies [31], people, technology and processes are considered the cornerstones of DevOps.

Since 2001, organisations have adopted agile methodologies for its SDLC [32] where the most implemented methodologies are XP and SCRUM [33]. These methodologies are the foundation of DevOps and DevOps can be seen as its extension, since they are based on the same principles of introducing short release cycles and to develop forward the customer or user feedback [34]. However, DevOps includes the operations team on early stages of the software development, being able to develop the software already with the operations team input, thus developing software more stable including the business feedback [35]. Also, DevOps stands out due to the collection of techniques and tool to enable software continuous delivery, clearing the path of the software to production [36,37].

In conclusion, a DevOps culture seems to be very attractive to organizations worldwide, being based in a "*The faster you fail, the faster you* recover" philosophy [14(p1)], enabling a culture of experimentation to release new products, services and software, allowing the organization to grow and to satisfy their customers [38].

3. Research Methodology

To achieve this research goal, the authors have chosen the systematic literature review (SLR) methodology. It is seen as one of the most widely used research methods to collect and synthesize evidence [39]. SLR's are meant to have a well-defined process to identify, evaluate, and interpret all the evidence collected during research [40,41]. Thus, for this investigation, the authors have followed the framework proposed by Kitchenham [40] where the SLR is split into three stages (Figure 1). Moreover, to add rigour to this research, the authors have chosen to perform two SLR's: the first to find all the benefits described on existing literature, while the second one being used to find instances of those benefits being reported on case studies from DevOps implementation. This second SLR will confirm and evaluate the findings from the first SLR, where all the DevOps benefits were gathered from literature. The authors believe that searching for case studies is a reliable method of evaluation given that these are a research methodology known by providing evidence of a certain phenomenon [42]. The first SLR was conducted between May and September 2020, while the second SLR was carried out between August and October 2020.

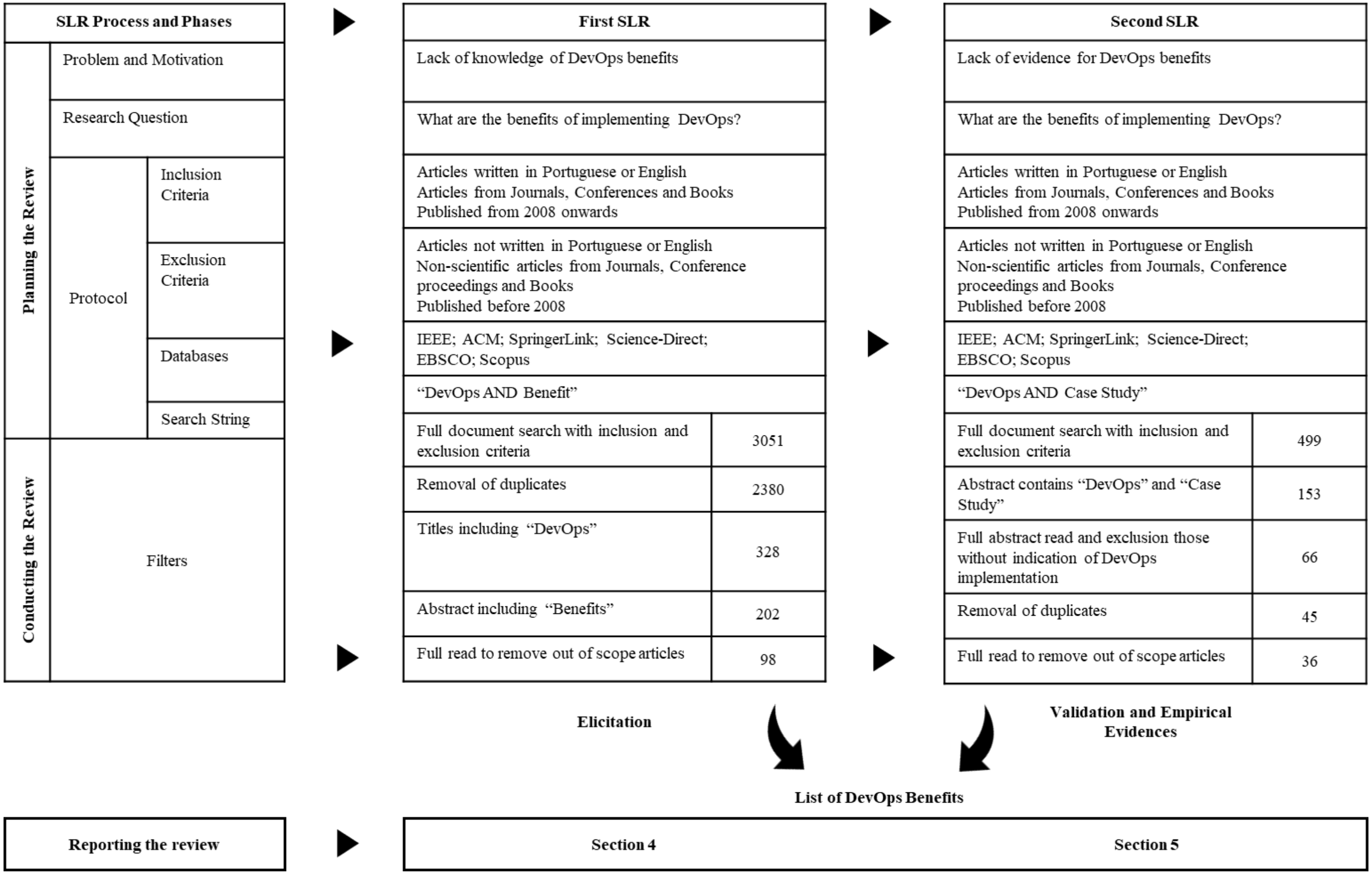
The process designed by Kitchenham [40] was followed by both SLR's. The authors have started by defining the Problem and Motivation for the review. For the first SLR where the expected result was to identify the reported benefits on the literature, the motivation was to acknowledge the DevOps benefits on the literature, while for the second SLR the motivation was to find evidence of the DevOps benefits. The next step of the process was to define the Research Question (RQ) for the review. In this case, the same RQ was identified for both SLRs "What are the benefits of implementing DevOps?".

After the RQ definition, the next step is to define a protocol where inclusion and exclusion criteria was defined, along with the search databases and the search string of each SLR. The inclusion and exclusion criteria were based on the language of the publications, scientific publications, and publication date. Regarding inclusion and exclusion criteria (IEC) a minimum date was set, considering that the DevOps concept was born after the aforementioned “Agile Infrastructure and Operations” conference, in 2008 [24]. For the databases, the authors have used some of the most known and used databases on the scientific community. All these criteria were the same for both SLRs except the search string. For the first SLR the search string was focused on DevOps benefits while for the second SLR the search string was focused on DevOps case studies.

After applying these criteria, some filters were added to the review to exclude publications that wouldn’t provide the necessary information for this research. One example of the used filters was the removal of duplicated. All this process definition can be seen with more detail in Figure 1 for both SLRs.

To evaluate the research subject’s trend, the researchers have analysed the date of publication of each relevant piece of literature from a chronological point of view. This is extremely helpful to prove that the research topic has a corresponding trend and is largely demanded by the market. The researchers used the concept-centric approach [43] to better synthesize and analyse the concepts elicited from the literature. This helps to understand the focus of the review, for a better understanding of the readers. Also, it helps the researchers to structure the review. The usage of the concept-centric approach can be seen in Section 4 where the concepts identified are the benefits found per reference, while in Section 5 one can see the case studies identified per reference.

Figure 1 – Steps performed in each of the performed SLRs



SLR Process and Phases			First SLR		Second SLR	
Planning the Review	Problem and Motivation		Lack of knowledge of DevOps benefits		Lack of evidence for DevOps benefits	
	Research Question		What are the benefits of implementing DevOps?		What are the benefits of implementing DevOps?	
	Protocol	Inclusion Criteria	Articles written in Portuguese or English Articles from Journals, Conferences and Books Published from 2008 onwards		Articles written in Portuguese or English Articles from Journals, Conferences and Books Published from 2008 onwards	
		Exclusion Criteria	Articles not written in Portuguese or English Non-scientific articles from Journals, Conference proceedings and Books Published before 2008		Articles not written in Portuguese or English Non-scientific articles from Journals, Conference proceedings and Books Published before 2008	
		Databases	IEEE; ACM; SpringerLink; Science-Direct; EBSCO; Scopus		IEEE; ACM; SpringerLink; Science-Direct; EBSCO; Scopus	
		Search String	“DevOps AND Benefit”		“DevOps AND Case Study”	
Conducting the Review	Filters		Full document search with inclusion and exclusion criteria	3051	Full document search with inclusion and exclusion criteria	499
			Removal of duplicates	2380	Abstract contains “DevOps” and “Case Study”	153
			Titles including “DevOps”	328	Full abstract read and exclusion those without indication of DevOps implementation	66
			Abstract including “Benefits”	202	Removal of duplicates	45
			Full read to remove out of scope articles	98	Full read to remove out of scope articles	36

Reporting the review	Section 4	Section 5

4. First SLR: List of Benefits

After performing the first SLR and analysing the articles, the list of DevOps benefits was elicited and can be seen in Table 1. The concept-centric approach taken by the researchers can be found in Appendix A, where it is possible to see the match between the concepts and the authors that have identified those concepts in literature.

In the next section, one can see a discussion and some conclusions about the benefits found on the literature, regarding Table 1. After the full read of the publications, the authors have identified the benefits described on the publications and grouped those benefits by the concepts, also seen on Table 1.

Several authors among literature claim to see an improvement on the communication and collaboration (as seen in Appendix A) between developers and operators [22,28,44,45], creating a synergetic environment where both teams desire to work together towards accomplishing overall objectives [11,46].

Table 1 – List of Benefits Identified in Literature

Concept ID	Benefits	# References
B01	Cross Team Collaboration and Communication	49
B02	Faster Time to Market	41
B03	Faster and Better Feedback Loops	38
B04	Increase of Code Quality	32
B05	Increase of Value	26
B06	Improvement of System Reliability	22
B07	Less Mean Time to Recover	17
B08	Increase of Team Performance	17
B09	Costs Reduction	13
B10	Processes and Tools Standardization	13
B11	Maximization of Competences	13
B12	Decrease of Manual Work	11
B13	Increase of Customer Satisfaction	11
B14	Less Failed Changes	11
B15	Increase of Employees Motivation	9
B16	More Innovation	8
B17	Better Deployment Management	5
B18	Less Security Issues	5
B19	Organizational Cultural Changes	2

Before DevOps, operators and developers may have had different mindsets when facing change. With the disappearing of the waterfall software development methodology and the emergence of the “Agile Manifesto” [10,47], the developer’s mindset shifted to deliver more features as fast as possible to production, while the operator’s continued to have the mindset of guaranteeing the stability of the systems it was solely responsible for [12]. These divergent views on change typically lead to finger-pointing, with operators blaming developers for the production impact of deployments when they might have been involved earlier in the development process to try to anticipate possible problems before they reached production [48].

Because of the resultant DevOps synergy, both operators and developers are more driven to collaborate across teams. They will feel that they are working towards a common and greater goal for everyone [49]. However, this can also be extended to the business. Just as Agile practices and principles brought business and developers together [10], DevOps introduces operators into the mix, emphasizing the significance of operations management in the organization [50].

As seen in Appendix A, faster time to market, related to continuous integration and continuous delivery capabilities, is one of the most reported benefits from DevOps. Organisations can design new, better features for their products as a result of faster feedback [51]. Through DevOps enabled automation they are then able to put said features into the market at a quicker rate than competition [52].

It is noteworthy to mention that various authors were able to identify the different sources that contribute for a better and faster feedback under DevOps: Customers and end users (business users) [14] as well as between the DevOps team itself [49]. Customers and end users are those who use the application; they are best to identify issues and potential improvements [53,54]. DevOps has a practice to shorten the Feedback Loops between Operators and Developers, which also leads to faster feedback when something is going wrong and requires further work [55].

Improved feedback does not only contribute for better development and application stability, but also leads to opportunities for DevOps teams to learn about its components (for example, Operators can learn about the development process, and developers learn about processes which guide operators work) as well as the business itself [56].

Code Quality can be increased as a result of implementing improved delivery pipelines under which code is built into packages and introduced to the respective repository after each check [57]. During the packaging of a new build, code can be submitted through quality gates, ensuring that best practices defined for that application are being adhered to [20]. Due to the continuous integration capability encouraged by DevOps culture, developers from several teams will be working in collaboration with other developers. There will be opportunities to find issues or needed improvements to other developers' code, improving the overall code quality of applications [58].

There is great consensus in literature about the increase of value when using DevOps. DevOps is a culture that uses Lean and Agile practices. DevOps phenomenon arose as an extension of agile software development inspired by lean concepts [59]. The first Agile Manifesto principle is about value: "*Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."* [10]. Due to the continuous delivery capability that DevOps employs, shorter development and release cycles [60] can be achieved, where business and customers will notice the ongoing improvement of software, realizing the continuous increase in value of their applications [29].

Automation brings an additional benefit in the ability to perform defined actions after an event is triggered by automated monitoring [61]. By automating infrastructure using infrastructure as code, the availability and reliability of applications will be improved. Such infrastructure can scale up and scale down according to its reported usage and demand [14].

Related to the Faster Time to Market benefit, feedback and automation is not only used to deliver new or improved products [62]. Due to the premises of DevOps in having a solid IT team, both developers and operations work together to guarantee that fixes are deployed in production instantly [63]. This contributes to the stability and availability of applications, so that defects do not cause greater impact [23].

The objective of implementing any framework, practice or methodology is to improve performance. However, operators and developers commonly have different objectives and use different metrics to measure their performance, as explained in Section 3.1.3.4. Thus, to improve a team's performance, an alignment is needed for the definition of clear and visible goals [64]. In case of operations, these would be aimed at the stability and reliability of an application, while for developers the focus should be on the features delivered for it [44,65]. Since DevOps is also targeted towards using Lean and Agile practices [66], it concentrates its aim on improving people, processes and technologies capabilities, specifically in the way the work in process is limited and done in small batches, therefore contributing to the well-being of their teams [29].

Cost reduction is amongst the top goals of every organization in the world. As discussed before, DevOps can help reduce costs by reducing bottlenecks in the SDLC, optimizing time to deploy changes in production and enabling better resource management [22]. This can help balance software quality with costs, helping organisations to have an increased return on investment [67,68].

To optimize the SDLC it is essential that operations can react quickly, helping developers have their environments stable, up and running. DevOps encourages operators to use the Infrastructure as a Code capability in order to help manage and configure environments more quickly and in a standardized way [69,70]. This allows developers to have development and pre-production environments, which aids in the discovery of issues early in the SDLC [71]. Likewise, the environment process configuration and tools used by each team should be standardized, avoiding common situations like "it was working on my machine" [72].

With the mixing of IT teams by making developers and operators work together, the competences of these resources will be increased [25]. Developers will be able to get abilities that most often regard to operations, while operators will get abilities on areas of development [65]. This also contributes for improved knowledge management [22], allowing a DevOps team to be more complete in terms of their joint competences.

On a Decrease of Manual Work a consensus can be concluded in literature. This is accomplished by using automation. There are three major areas where it can be applied: testing [73], the delivery pipeline [74] and configuration or provisioning [75]. Test scripts can be automated by using tools that will perform the actions of the testers, verifying if the final output is the desired one. Thus, this capability reduces the manual work of testers as well as the risk for human error. Moreover, automated tests enable continuous testing capability which helps find integration issues earlier in the development cycle, making defect resolution faster and with less impact on production environments [73]. This also frees up the tester to create other, exploratory testing activities.

Operations are usually not only responsible to guarantee the stability of production environments but also of the lower environments. If a development team requires several development environments, each requiring operators to configure manually, a blocking of development resources may occur. DevOps encourages the usage of infrastructure-as-code to allow the operators to manage their infrastructure and environment configurations by using code, replicating said configurations for several alternative environments, speeding up configuration [71]. Furthermore, it is possible to automatically provision environments with resources based on predetermined thresholds, guaranteeing their stability and availability [68].

DevOps encourages developers to continuously integrate their code so that issues can be found earlier [76]. However, this requires a lot of work if every time a developer checks-in his code, a manual package needs creation for other developers to review. Under DevOps, every time that a developer commits code to a code repository, a script is triggered that will automatically test and create a package or artifact, checking and giving immediate feedback if there is any error and, if successful, storing it properly [56]. From this point onwards, the developed package can be used for installation across all environments. With the package stored, one could also trigger a script that will deploy the package with new code in a test environment, making it available for testers; alternatively, once the deployment is completed, more complex automated testing can be triggered, like integration or end-to-end tests and developers informed if the new code failed in any test, speeding up the bug fixing and increasing the software's stability [22,77].

Customer satisfaction can be seen as a consequence, resulting from a variety of previously described benefits. Since DevOps will continuously improve the stability of applications while reaching for customer feedback, customer satisfaction will increase [22]. By reducing bottlenecks on the SDLC process, the customers' feedback is deployed on the application faster, further increasing satisfaction [78]. Also, looking from a perspective in which a customer is internal, DevOps can also contribute to cost reduction (please see section 4.1.13).

Less Failed Changes can be seen as a consequence from both the standardisation of processes and tools, as well as from other DevOps capabilities in general. With a standardisation of processes, like those used in a deployment, for example, issues on a deployment script can be found and fixed in other environment, before reaching a production deployment [22]. With all the automation (in testing areas, for example) and continuous integration that DevOps encourages, help is obtained towards identifying issues with development work earlier on the SDLC, helping to avoid failed changes when moving to production [79].

Employees of an organisation will feel more motivated by working on a more communicative environment, in which they feel that their team will back them up [12]. This will contribute to reduced blame-games between developers and operators [54]. Due to the sharing culture that DevOps promotes, developers will learn about operators’ tasks just as much as operators will learn about developers’ tasks. Thus, employees will be more capable to backup each other up on different types of work [15,80].

Due to the increased speed of development, and by enabling a faster time to market, DevOps allows organisations to experiment new solutions, features and products [29] without incurring in significant economic impacts. Start-up companies are known for creating new market segments due to the innovative solutions they create. DevOps brings a great opportunity for these organisations, which does not have much revenue, allowing a spirit of the “*Faster you fail, faster you recover*” [14(p1)].

The setup of IT Teams before DevOps were structured in a way that deployments were manually performed by single or multiple operation teams that had the responsibility for configuring and setting up production environments, database configuration, backups of software build and reversing bad builds on the new software [56]. This raises the possibility and concern of human errors, which can impact the entire service stack of an organization [70]. Automation is one of the most used capabilities in DevOps which can help on this matter. By building automatic deployment mechanisms it is possible to decrease the volume of potential outages from applications [65]. Moreover, DevOps gives the ability for developers to perform their own deployments under the motto “You build it, you run it” [81], which empowers developers to find bad builds before operators, resulting in improved deployment management.

DevOps promotes monitoring during the entire deployment pipeline, using tools to notify developers and operators in case of something going wrong, or the need for manual actions, like rolling back the software to a previous version [63], contributing also to a better deployment management.

DevOps is usually allied with cloud implementations which help deploy security integration and carry out penetration tests between applications [82,83]. Nowadays, cloud providers offer services that promote the usage of DevOps, in which a security model for their customers is ensured [84].

As discussed earlier, DevOps is not only focused on automating processes and improved performance, but also on cross team collaboration and interaction between people. For DevOps, or other agile software development methodologies, organisations need to have a culture that allows for these interactions. Lean, Agile, and DevOps appeared in various times to meet various requirements [85], but they all concentrate on organizational culture by forming interdisciplinary teams, cutting waste, concentrating on the customer, embracing change, and providing value on a continual basis. Under DevOps, sharing is the key for operators and developers to work together. As such, organizational culture needs to be adapted to promote this kind of involvement [77].

5. Second SLR: Empirical Evidences of DevOps Benefits

A second SLR was carried out to confirm and evaluate the findings from the first SLR, in which all DevOps benefits were gathered from literature. To do so, the authors captured and analysed a total of 36 DevOps implementation Case Studies. Each of the studies was read for data on the outcomes of introducing DevOps capabilities in a business environment. A list of these articles, their references, and basic vectors of analysis, are found on Table 2.

Due to the data provided in Table 2, it was possible to produce Figure 2 with a segregation of the case studies by Continent, Country, and Business Sector. It is possible to see that DevOps is more present in Europe or on Multinational organisations that work in several countries from multiple continents. Regarding the business sector, the IT business sector clearly stands out from the other sectors. Since DevOps is a culture that is focused on IT developers and operators, it makes sense that IT organisations implement this culture before other sectors. However, from professional experience from the authors, the DevOps culture have been expanding on the financial sector (Banking and Insurance).

Having identified and analysed the final list of DevOps implementation articles, we proceeded to map business benefit Concept IDs to Case Studies in which they are mentioned. Some of the documents included findings from more than one Case Study; for these, we relied on decimals to differentiate implementation results from each organization as much as we possibly could. However, it is important to note that some authors merged in a single body the observations and results of multiple, different DevOps case studies, making full differentiation impossible. In total, 69 Case Studies were identified and reviewed as part of our research. The results of this effort are presented in Table 3 (also refer to Appendix B). Moreover, one of the case studies didn’t presented any benefit, where the authors have identified to study the benefits as their own future work. The Benefit ID and Benefit Description columns are referring to the concepts previously presented in Table 1. Lastly, it is relevant to add that most of the Case Studies did not provide quantitative evidence of these benefits, but often referred to them in a qualitative manner.

Table 2 – List of DevOps Implementation Case Studies Analysed

ID	Reference	Country	Continent	Business Sector
CS.01	86	Sweden	Europe	Information Technology
CS.02	87	Spain	Europe	Human Resources
CS.03	88	Italy	Europe	Lighting Business
CS.04	56	Denmark	Europe	Information Technology
CS.05	28	Brazil	South America	Government Organization
CS.06	89	Morocco	Africa	Information Technology
CS.07	90	Montenegro	Europe	Banking Industry
CS.08	91	Germany	Europe	Information Technology
CS.09	55	USA	North America	Information Technology
CS.10	92	Multinational	Multinational	Healthcare
CS.11	93	USA	North America	University Project
CS.12	94	Finland	Europe	Information Technology
CS.13	95	New Zealand	Oceania	Finance & Insurance Industry
CS.14	96	USA	North America	Government Organization
CS.15	69	Spain	Europe	Information Technology
CS.16	53	Multinational	Multinational	Mixed
CS.17	15	Finland	Europe	Information Technology
CS.18	97	Australia	Oceania	Information Technology
CS.19	98	Finland	Europe	Information Technology
CS.20	99	USA	North America	Information Technology
CS.21	18	Multinational	Multinational	Information Technology
CS.22	100	N/A	Europe	Information Technology
CS.23	101	Multinational	Multinational	Mixed
CS.24	76	(Not Provided)	(Not Provided)	Finance & Insurance Industry
CS.25	102	(Not Provided)	(Not Provided)	Information Technology
CS.26	23	Multinational	Multinational	Mixed
CS.27	103	Multinational	Multinational	Information Technology

CS.28	[104]	UK	Europe	Information Technology
CS.29	[70]	Multinational	Multinational	Information Technology
CS.30	[105]	Spain	Europe	University Project
CS.31	[106]	USA	North America	Government Organization
CS.32	[107]	Multinational	Multinational	Information Technology
CS.33	[108]	Sweden	Europe	Information Technology
CS.34	[109]	Finland	Europe	Information Technology
CS.35	[110]	Germany	Europe	Information Technology
CS.36	[68]	(Not Provided)	(Not Provided)	Information Technology

An improvement in the rate by which new development is produced, deployed, and reaches the market was by a considerable margin the most widely and explicitly observed benefit of a DevOps adoption. The implementation of DevOps practices, particularly when it comes to establishing a bridge between development and operations teams [94], was commonly pointed out as an enabling factor towards faster delivery [111]. The added flexibility associated with DevOps practices allows for new software evolutions to be implemented faster, while sustaining a quality standard [76]. Shorter response times [53] and increased deployment speed [107] are likely to be observed in a successful DevOps integration. In Luz et al.'s [28] study it is stated that "after the DevOps adoption, it was possible to make 29 deployments on a single day" whereas before, due to rigid and conflicting policies at the operational level, deployment were only scheduled to occur once, weekly.

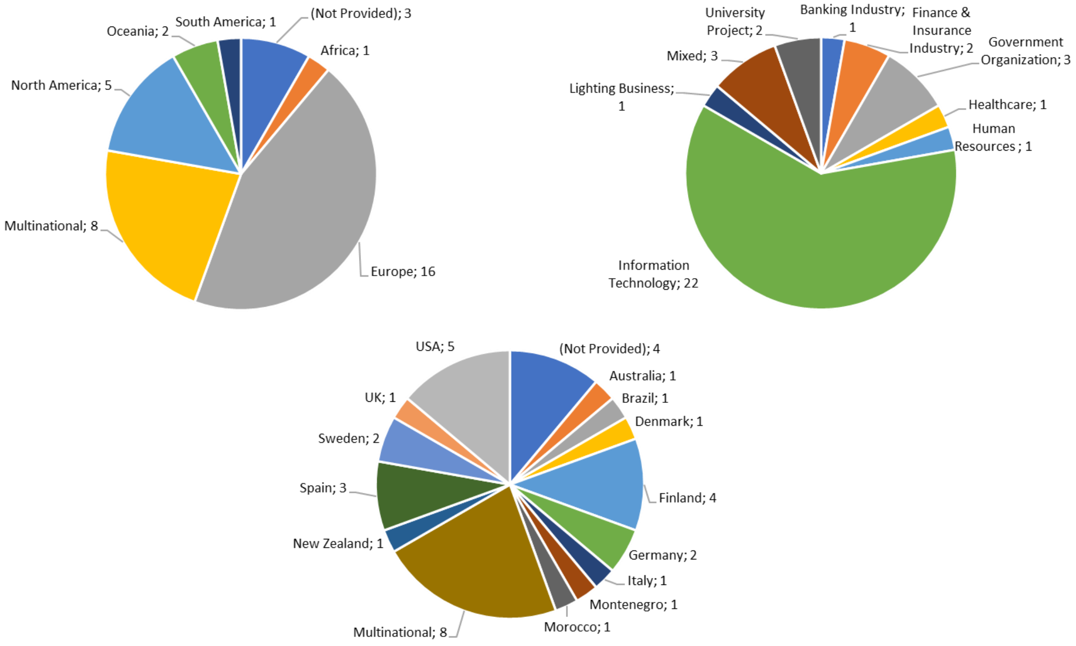


Figure 2 – Case Studies Segregation

As established, the development of synergies between teams is a foundational principle for applying DevOps practices. From our research, improved collaboration and communication between developers and operational staff was a frequently reported benefit resulting from DevOps implementation. Increased awareness of the overall software development processes, standard deployment practices and service management took place [70] as teams abandoned traditional "work silos" in favour of DevOps [94]. Furthermore, this "empowerment of teamwork" [99] between development and operations seems to heavily tie in with other business benefits ranging from improved reliability, quality, and security [15] to competence maximization, innovation and employee motivation. Referring to Shahin's [107] work, in interviews that were held with participants of a DevOps implementation study, the opportunity of learning about overall operational and architecture aspects was often commented as a deeply useful and "growing exercise".

Table 3 - DevOps Benefits Analysis

Benefit ID (from Table 1)	**Benefit Description (from Table 1)**	**Occurrences in case studies**	**Percentage of Case Studies**
B02	Faster Time to Market	49	71%
B01	Cross Team Collaboration and Communication	39	57%
B12	Decrease of Manual Work	38	55%
B08	Increase of Team Performance	30	43%
B04	Increase of Code Quality	27	39%
B17	Better Deployment Management	25	36%
B06	Improvement of System Reliability	23	33%
B03	Faster and Better Feedback	22	32%
B10	Processes and Tools Standardization	19	28%
B11	Maximization of Competences	20	29%
B13	Increase of Customer Satisfaction	19	28%
B15	Increase of Employees Motivation	18	26%
B09	Costs Reduction	12	17%
B07	Less Mean Time to Recover	10	14%
B19	Organizational Cultural Changes	9	13%
B16	More Innovation	6	9%
B05	Increase of Value	6	9%
B14	Less Failed Changes	5	7%
B18	Less Security Issues	2	3%

DevOps practices emphasize automation over manual work in the development, testing and deployment of software [23]. Over 50% of the reviewed Case Studies clearly mention a reduction in the volume of manual tasks. For example, in Laukkanen et al's [108] study, "manual test[ing] had been the bottleneck" for reducing feature freeze periods; with DevOps implemented, release tests for specific systems were automated, causing a reduction in the time necessary for completion. Luz et al. [28] also describe how before having DevOps implemented a vast majority of automatable tasks were done manually, often causing errors and need for rework. Similar to what we observed in our analysis of B01 (Cross Team Collaboration and Communication), the benefit of reducing manual work appears to tie in with faster delivery[14], less failed changes, improved code quality and even employee motivation, as was observed in R. K. Gupta et al.'s [92] case study. Here, teams focused on incremental automation, focusing on a single, critical workflow at a time; upon reviewing progress, it is stated that "such small successes motivated the team", encouraging them to pursue further automation.

Although Increase of Value (B05), Less Failed Changes (B14) and Less Security Issues (B18) were not commonly and explicitly discussed in the analysed Case Studies, there is room for further investigation towards better understanding how business benefits can relate to each other. Despite said links not being subject to investigation under the present research, it may not be unreasonable to consider that organizations who increase release rates and quicken their time to market (B02) are in a better position to deliver greater value to stakeholders (B05); or that those who significantly improve communication and collaboration between Developers and Operations (B01) may also observe a reduction in failed changes or release faults (B14).

6. Results and Discussion of DevOps Empirical Evidences

Table 4 presents examples for each of the 19 business benefits identified as part of our research. Where applicable, cells referring to the "Problem Solved" column are also filled in, indicating the motivation or reasoning that led to the implementation of DevOps, which led to the observed benefits. This section shows that DevOps can solve different problems on the organisations, indicating an empirical evidence of the benefits got after the DevOps implementation.

Table 4 - Case Study Analysis: DevOps Benefit and Problem Solved

Benefit	Example of Benefit	Problem Solved	Case Study
(B01) - Cross Team Collaboration and Communication	*“The inclusion of operation team members and operation topics help the operation team to know the development topics and plan their readiness accordingly. Additionally, they take building knowledge and feedback for risk assessment without additional effort.”*	*“We soon realized that with the current approach we would not be able to release the first couple of version increments. Team members in India and USA have experience in traditional software development and product management group in Germany has no experience in software development.”*	CS.10
(B02) - Faster Time to Market	*The organization achieved a one deployment per week frequency, with one hour / one day lead time for changes.*	*“The organization size, the diversity of its departments (development, operations, security, service, QA, architecture, etc.) as well as the interaction between them, and the complexity of its processes, hampered reducing time to market, and made this company less competitive”*	CS.15
(B03) - Faster and Better Feedback	*“The flexibility afforded by the DevOps approach allowed the development teams to recognize, characterize and accommodate- date changes to DART’s control algorithms for NEXT-C in real time. The team was able to update the test specifications and procedures in real time, and ultimately achieve the goal of demonstrating NEXT-C at Technology Readiness Level.”*	*“While NEXT-C was well characterized from its own development and test perspective, there were unknowns in the specifics of DART’s tailored use-case for the thruster.”*	CS.14
(B04) - Increase of Code Quality	*“Higher levels of automation were found to drive improved quality assurance. (...) The automated DevOps production pipeline helps to ensure that every change is verified before it is pushed forward for delivery.”*	*Description of a Problem / Motivation was not provided.*	CS.12
(B05) - Increase of Value	*“Increase in deployment frequency from about 30 releases a month to an average of 120 releases per month.”*	*“Need for a change by the business in order to remain agile and competitive. (...) Prior to DevOps, the company had been maintaining and developing its aging monolith application that was hosted in a traditional data centre.”*	CS.13

(B06) - Improvement of System Reliability	*"The time spent in the queue for the Basic approach is about 330 times that of the Containerized approach, and similarly the queue time using the Hosted agent is 1,110 times that of the Containerized approach, which translates to significant time saved. Since all of the infrastructure is managed without any new cost incurred, yet the throughput is high, our CI/CD pipeline is very lean."*	*"We recently decided to move towards a micro-services-based architecture (...) Consequently, the number of build and release definitions would increase significantly, and the infrastructure that was utilized may no longer be sufficient."*	CS.8
(B07) - Less Mean Time to Recover	*"This case study illustrates how rapid and simple its deployment was, in accordance with the DevOps principles, and therefore focusing on how self-service monitoring infrastructure for threats detection provided engineers—both developers and IT operators—fast and continuous feedback of the Library Energy-Efficiency System deployed into production. (...) it provides evidence of how this cybersecurity monitoring infrastructure enabled to detect threats, such as denial attacks, and helped to better anticipate spoofing problems."*	*"The development and deployment of such systems [IoT] into production as well as their operation and monitoring are highly complex due to the heterogeneity of delivery endpoints. (...) The Cluster of European Projects on Software Engineering for Services and Applications highlights the importance of ensuring Quality of Service (QoS) and correctness of IoT systems together with the complexity of such purpose as devices and software could change for various reasons such as bugs and failures, changing interfaces and implementations, and changing requirements."*	CS.30
(B08) - Increase of Team Performance	*"Considering the e-TCE system, after the DevOps adoption, it was possible to make 29 deployments on a single day. Before the DevOps adoption, and due to the rigid policies of the operations team, the deployments were schedule to occur once a week."*	*"Before DevOps, deployment activities were historically a controversial point at the TCU. Several conflicts occurred over time. Rigid procedures were created to try to avoid problems."*	CS.5
(B09) - Costs Reduction	*"Most companies confirm that DevOps brings shorter response time and more frequent deployments, higher productivity, better feedback from the client and lower IT cost."*	*Description of a Problem / Motivation was not provided.*	CS.16
(B10) - Processes and Tools Standardization	*"Although having simpler deployment pipeline for each component or service can bring a lot of benefits but the requirement of a dedicated pipeline needs extra effort to set up the dedicated pipelines for the first time. Some of the participants reported that they were employing automation technologies such as Docker to simplify the deployment process."*	*"Our analysis of the data revealed that it was challenging for a couple of practitioners to design applications for different operations environments, in which they may have had difficulty to make consistency in a set of heterogeneous operations environments"*	CS.32

(B11) - Maximization of Competences	*"The advantage is that the DevOps team teaches the student the necessary activities and attempt to integrate him\her into the team. There are no educational programs, for example, from the university that teach all necessary competencies that are required to work in a DevOps team. Hence, companies train their students or team members to get ready for the role."*	*"In the traditional silo organized IT department, there is a high level of specialist knowledge. However, in the DevOps setups, these departments are linked, and the human capitals move from highly specialized to more generalized knowledge."*	CS.26
(B12) - Decrease of Manual Work	*"Overall, developers are able to perform the defect validations much more quickly without having to wait to manually configure the hardware with latest software bundles having their fix in it. With this automation, developers have full control – to validate any defect they have to just pick and choose the config and within few clicks they will have a setup up and running on which, they can validate the defect in production like environment."*	*"No organizations can afford to live with manual, error prone and repeated activities in the software delivery lifecycle (...) the project teams identify this precise business need and adopt DevOps to optimize their processes, it is going to reap more fruits."*	CS.36
(B13) - Increase of Customer Satisfaction	*"The more and faster development team adds new features, more citizens visit the website or in the web application. (...) The deliverables may be released daily or at the end of the release cycle time. Subsequently, the development team gains faster feedback from end-users that would help in mitigating several risks"*	*Description of a Problem / Motivation was not provided.*	CS.23
(B14) - Less Failed Changes	*"Because every change in the code is checked at every stage of the development, and errors are discovered and resolved on the fly, the end products have fewer bugs, and the software can be readily released."*	*Description of a Problem / Motivation was not provided.*	CS.12
(B15) - Increase of Employees Motivation	*"The instantiation of the role rotation in the cross-functional DRR practice in our case enabled large-scale learning and KS since all team members were able to perform several roles and become more knowledgeable. (...) When team members rotate, they can take on responsibilities, develop skills, and acquire knowledge. This fosters the team's autonomy."*	*Cross-functional collaboration and team self-organization were described as major challenges.*	CS.25
(B16) - More Innovation	*"The single-case study presented in this research was helpful to answer the two research questions. First, DevOps may be considered an approach that contributes to implementing innovation for software-*	*"To develop its own consulting approach, T-Systems MMS initiated a DevOps program, which explicitly aims to*	CS.35

	defined business environments, (…) Second, the case shows that (IT) consulting companies need to transform themselves for DevOps."	*improve the company's offering in the area of innovative digital services."*	
(B17) - Better Deployment Management	*"This has reduced errors caused by builds with wrong dependencies, incorrect deployment documents, and human errors in general, since only automated processes would deploy in the environments. (…) Initially there will be the impression that some legacy systems and technologies will not be able to be automated or benefitted by the Continuous Delivery process, but in the case of the institution of the case study, even COBOL and Powerbuilder systems have benefited from process automation."*	*"It was identified that the deployment process executed until the beginning of this work required a lot of effort and there was a lot of bureaucracy."*	CS.24
(B18) - Less Security Issues	*"The success so far shows that organizations with large bureaucratic obstacles and stringent software security and accreditation requirements are able to use (Sec)DevOps processes and toolsets to produce software that meets security and accreditation requirements and ultimately satisfies their customers."*	*"Ensure that security became a continuous practice rather than being tacked on at the end."*	CS.31
(B19) - Organizational Cultural Changes	*"DevOps culture and mind-set, which were enriched with colocation, were observed in the wider dissemination of DevOps approach across the organisation."*	*"Prior to this improvement, the team spent huge efforts in merging code and resolving merge conflicts, which were causing broken builds often."*	CS.17

7. Challenges in DevOps Adoption: The other side of the coin

Even thought, this research is about the benefits of the DevOps culture adoption, the main objective is to show what to expect when adopting DevOps. Thus, for this article some DevOps challenges will also be presented, since some of the researchers that identified DevOps benefits, were also able to identify challenges to the DevOps implementation. In Table 5, one can see which challenges were identified by the researchers that also identified benefits.

Table 5 - DevOps Adoption Challenges

ID	Challenge	# Of References	References
C.01	Industry constraints	2	[29] [112]
C.02	Deep-seated company culture	2	[39] [29]
C.03	Insufficient communication	1	[29]
C.04	DevOps is unclear	1	[29]

As it can be seen, some of the challenges shown in Table 5, are more related with the culture, environment, and business industry where the DevOps culture is being implemented, rather than the technologic point of view of DevOps, such as “Insufficient communication”, “Deep-seated company culture” and “Industry constraints”. This shows that when an organisation is thinking to adopt DevOps, should self-assess if it is culturally ready for this change. Moreover, to help to mitigate this challenge, the top management of the organisation should be propelling for this change so it could be example for the rest of the organisation [25]. But there is a technologic challenge regarding the automation of the deployment scripts for several technologies. Organisations have multiple applications, where each of them can have different coding languages which needs its own deployment script. This requires a lot of different skills for DevOps to be able to automate these different deployment scripts.

DevOps has been evolving constantly, which could help regarding the challenge “DevOps is unclear but also evolving”. The amount of publications shows that the DevOps adoption has been growing over the time, showing that organisations have been able to understand how to implement DevOps.

Every new adoption for an organisation takes time to learn, and DevOps is not an exception for it. To adopt DevOps, it is important to give training to the organisations employees so they can understand how to implement DevOps.

8. Validity of the SLRs

The authors have submitted this research to validity tests where the validity is made in four different categories, construct validity, external validity, internal validity and conclusion validity [113]. Zhou et al. [113] have performed a research to synthetise the most common pitfalls when performing literature reviews by the different review phases. In Table 6 one can see some of these common pitfalls and how the authors have passed the test for this research.

Table 6 - Validity Tests

Pitfall Description	Review Test
Non-specification of SLR’s setting and sufficient details	These pitfalls are regarding the planning phase of the review. However, this research has a process and protocol correctly defined describing the decisions for the criteria, databases and search terms used. This shows a path that other researchers can follow to reproduce and replicate this research, adding more validity to this research.
Incorrect or incomplete search terms in automatic search	
Incomprehensive venues or databases	
Inappropriate inclusion & exclusion criteria	
Inadequate size and number of samples	For both SLRs on this study, it was possible to gather a significative amount of publications. From these samples, the authors were able to identify several benefits on the first SLR likewise, on the second review where was possible to identify several DevOps case studies.
Restricted time span	The only time restriction defined was the minimum date of research since DevOps was first presented in 2008.
Bias in Study Selection	To avoid the bias study selection, the authors have defined filters and criteria to select the studies on the same way for all of them.
Paper/database inaccessible	The databases used are some of the known databases by the Academic/Scientific and Software engineering communities, showing the reliability of these databases.
Primary study duplication	To avoid duplication, the authors have applied a filter to remove duplicated articles.
Bias in Data Extraction	The several authors of this research have reviewed the data extracted from each author to avoid that some researchers have not identified important data.

9. Conclusion

DevOps is a novel culture being adopted worldwide. The authors noticed a lack of synthetization for DevOps implementations benefits in present literature. Thus, the objective for this research was to consolidate the benefits of DevOps implementation so new practitioners know what to expect when adopting the methodology.

To accomplish this objective, the authors have chosen to perform an SLR on the benefits reported in literature. The SLR methodology is known for adding rigor to research due to the well-defined protocol that one must comply to when defining it. Additionally, a second SLR was carried out to find Case Studies of DevOps implementation. This second SLR was important for research, allowing for the mapping between issues that organisations faced, what were the achieved benefits, and what empirical evidence are there, respectively. Given the accomplishment of the study objective, it is possible to note that this study brings contributions to the theoretical body of knowledge by synthetising the DevOps implementations benefits.

Regarding the findings originated from this research it is possible to state that even though there was a small number of studies in common between both SLR's, all benefits listed from the first SLR were also found on the second SLR. This demonstrates that empirical evidence exists for said benefits. It was also interesting to note that the top five benefits with more references from the first SLR are not the same as the top five benefits with more occurrences in the second SLR. Of the top five from the first SLR one can find benefits B03 and B05, while on the second SLR one finds benefits B08 and B11. Comparing B05 with B08, the authors can understand that it is easier to measure an improvement in team performance rather than a measure of value increase. As such, it makes sense to find B08 with more occurrences with empirical evidence. Furthermore, when comparing B03 with B11, one can also suppose that all the automation that DevOps encourages makes it easier to record a decrease of manual work, as the effect should be immediate, while faster and better feedback often results from willingness by individuals themselves.

It is possible to see that the most reported benefits are common between the two SLR's. Those benefits are B01 and B02. This is aligned with the premises of DevOps, bridging the gap between developers and operators, working together in delivering software or products faster to their customers.

Regarding the least reported benefit it is possible to see B18 on the bottom of each SLR. It seems that this benefit is related with DevOps, but it is more specifically studied as an own discipline for security, called DevSecOps.

The fact that Case Study authors did not frequently provide quantitative evidence regarding the observed business benefits did increase the difficulty of establishing fully consolidated findings. This brings the opportunity of future researchers to expose metrics on how to measure the DevOps benefits, to compare how the organisations business units behave with these DevOps benefits. Another limitation to this study is due to the novelty of DevOps, the authors couldn't apply a quality filter on the SLR's for top conferences and top journals, otherwise, the total amount of articles for analysis would be low. As future work, the authors suggest performing a similar study for DevOps, but instead of benefits it could be directed at finding adoption challenges and how to overcome them. The authors believe that combining this research with a study where adoption challenges are tackled would help new DevOps practitioners clarify what is expected to be achieved with DevOps and how to go about its implementation. Moreover, this research would help organisation on the decision to implement DevOps, since this research shows the trade-off between challenges and benefits. Furthermore, there may be value in studying to what extent do identify DevOps business benefits can relate to each other, building a potential series of linked, expected improvements for business.

10. References

1. Ajayi BA, Hussin H. Influence of ITG on organisation performance: The mediating effect of absorptive capacity. *Proceedings - 6th International Conference on Information and Communication Technology for the Muslim World, ICT4M 2016*. Published online 2017:1-6. doi:10.1109/ICT4M.2016.13
2. Setyadi R, Fattah A, Waseso B. Trust Effect on Business-IT Governance Alignment in Society Culture (A Case Study in Indonesia). *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*. Published online 2019:1-5. doi:10.1109/CITSM47753.2019.8965411
3. Bartolini C, Salle M, Trastour D. IT service management driven by business objectives An application to incident management. In: *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*. ; 2006:45-55. doi:10.1109/NOMS.2006.1687537
4. Cavalcante V, Bianchi S, Braz A, Amorin F, Nauata N. Investigating business needs fluctuations on IT delivery operations. *Annual SRII Global Conference, SRII*. Published online 2014:19-26. doi:10.1109/SRII.2014.13
5. Setyadi R. Assessing Trust Variable Impact on the Information Technology Governance Using Business-IT Alignment Models: A Model Development Study. *ICSECC 2019 - International Conference on Sustainable Engineering and Creative Computing: New Idea, New Innovation, Proceedings*. Published online 2019:218-222. doi:10.1109/ICSECC.2019.8907224
6. Wiedemann A, Forsgren N, Wiesche M. Research the Practice: The DevOps Phenomenon. *Communications of the Acm*. 2019;68.
7. Trivedi P, Sharma A. A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation. *Proceedings of*

the 2013 2nd International Conference on Information Management in the Knowledge Economy, IMKE 2013. Published online 2013:188-194.

8. Michener JR, Clager AT. Mitigating an Oxymoron: Compliance in a DevOps Environments. *Proceedings - International Computer Software and Applications Conference*. 2016;1:396-398. doi:10.1109/COMPSAC.2016.155
9. Chen L. Continuous Delivery: Overcoming adoption challenges. *Journal of Systems and Software*. 2017;128:72-86. doi:10.1016/j.jss.2017.02.013
10. Beck K, Beedle M, Bennekum A van, et al. Agile Manifesto. Published 2001. Accessed January 9, 2020. https://agilemanifesto.org/
11. Katal A, Bajoria V, Dahiya S. DevOps: Bridging the gap between development and operations. In: *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*. IEEE; 2019:1-7. doi:10.1109/ICCMC.2019.8819631
12. Hussaini SW. A Systemic Approach to Re-inforce Development and Operations Functions in Delivering an Organizational Program. In: *Complex Adaptive Systems*. Vol 61. Elsevier Masson SAS; 2015:261-266. doi:10.1016/j.procs.2015.09.209
13. Rance S. *ITIL 2011 Service Transition*. 1st Editio. (Hanna A, ed.). The Stationery Office; 2011.
14. Soni M. End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. In: *Proceedings - 2015 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2015*. ; 2016:85-89. doi:10.1109/CCEM.2015.29
15. Lwakatare LE, Kilamo T, Karvonen T, et al. DevOps in practice: A multiple case study of five companies. *Information and Software Technology*. 2019;114(March 2017):217-230. doi:10.1016/j.infsof.2019.06.010
16. Sharma S, Coyne B. *DevOps For Dummies*. 2nd ed. (IBM, ed.). John Wiley & Sons, Inc; 2014.
17. Ebert C, Gallardo G, Hernantes J, Serrano N. DevOps. *IEEE Software*. 2016;33(3):94-100. doi:10.1109/MS.2016.68
18. Caprarelli A, Nitto E, Tamburri D. Fallacies and Pitfalls on the Road to DevOps: A Longitudinal Industrial Study. In: ; 2020:200-210. doi:10.1007/978-3-030-39306-9_15
19. Jabbari R, bin Ali N, Petersen K, Tanveer B. What is DevOps? A Systematic Mapping Study on Definitions and Practices. In: *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops*. ; 2016:1-11. doi:10.1145/2962695.2962707
20. Ivanov V. Implementation of DevOps pipeline for Serverless Applications. *Springer*. Published online 2018. doi:10.1007/978-3-030-03673-7_4
21. Teixeira D, Pereira R, Henriques T, da Silva MM, Faustino J, Silva M. A Maturity Model for DevOps. *International Journal of Agile Systems and Management*. 2020;13(4):464-511. doi:10.1504/IJASM.2020.112343
22. Jabbari R, bin Ali N, Petersen K, Tanveer B. Towards a benefits dependency network for DevOps based on a systematic literature review. *Journal of Software: Evolution and Process*. 2018;30(11):1-26. doi:10.1002/smr.1957
23. Wiedemann A, Wiesche M, Krcmar H. Integrating development and operations in cross-functional teams — Toward a DevOps competency model. *SIGMIS-CPR 2019 - Proceedings of the 2019 Computers and People Research Conference*. Published online 2019:14-19. doi:10.1145/3322385.3322400
24. Lwakature LE. Devops Adoption and Implementation in Software Development Practice: Concept,

practices, benefits and challenges. In: ; 2017:99. doi:ISBN 978-952-62-1711-6

25. Leite L, Rocha C, Kon F, Milojicic D, Meirelles P. A survey of DevOps concepts and challenges. *ACM Computing Surveys*. 2019;52(6). doi:10.1145/3359981

26. Punjabi R, Bajaj R. User stories to user reality: A devops approach for the cloud. In: *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings*. ; 2017:658-662. doi:10.1109/RTEICT.2016.7807905

27. Silva MA, Faustino J, Pereira R, Silva MM. Productivity Gains of DevOps Adoption in an IT Team: A Case Study. In: Andersson B, Johansson B, S. Carlsson C, Barry ML, Linger H, Schneider C, eds. *Conference: 27TH INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS DEVELOPMENT*. ; 2018:13.

28. Luz WP, Pinto G, Bonifácio R. Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*. 2019;157:1-16. doi:10.1016/j.jss.2019.07.083

29. Riungu-Kalliosaari L, Mäkinen S, Lwakatare LE, Tiihonen J, Männistö T. DevOps adoption benefits and challenges in practice: A case study. In: *17th Internacional Conference, PROFES 2016, Throndheim, Norway*. Vol 10027 LNCS. ; 2016:590-597. doi:10.1007/978-3-319-49094-6_44

30. Faustino J, Pereira R, Alturas B, da Silva MM. Agile information technology service management with DevOps: an incident management case study. *International Journal of Agile Systems and Management*. 2020;13(4):339-389. doi:10.1504/IJASM.2020.112331

31. Teixeira D, Pereira R, Henriques TA, Silva M, Faustino J. A Systematic Literature Review on DevOps Capabilities and Areas. *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*. Published online 2020:22. doi:10.4018/IJHCITP.2020070101

32. Silva CC, Goldman A. Agile methods adoption on software development: A pilot review. In: *Proceedings - 2014 Agile Conference, AGILE 2014*. Institute of Electrical and Electronics Engineers Inc.; 2014:64-65. doi:10.1109/AGILE.2014.14

33. Rauf A, Alghafees M. Gap Analysis between State of Practice and State of Art Practices in Agile Software Development. In: *Proceedings - 2015 Agile Conference, Agile 2015*. Institute of Electrical and Electronics Engineers Inc.; 2015:102-106. doi:10.1109/Agile.2015.21

34. Govil N, Sayrakhia M, Agnihotri P, Shukla S, Agarwal S. Analyzing the Behaviour of Applying Agile Methodologies & DevOps Culture in e-Commerce Web Application. In: *4th International Conference on Trends in Electronics and Informatics*. IEEE; 2020. doi:10.1109/ICOEI48184.2020.9142895

35. Snyder B, Curtis B. Using Analytics to Guide Improvement during an Agile-DevOps Transformation. *IEEE Software*. 2017;35(1):78-83. doi:10.1109/MS.2017.4541032

36. Dornenburg E. The Path to DevOps. *IEEE Software*. 2018;35(5):71-75. doi:10.1109/MS.2018.290110337

37. Silva MA, Faustino J, Pereira R, Mira M. Productivity Gains of DevOps Adoption in an IT Team : A Case Study. *27th International Conference on Information Systems Development*. Published online 2018. https://repositorio.iscte-iul.pt/handle/10071/16388

38. Furfaro A, Gallo T, Garro A, Saccà D, Tundis A. ResDevOps: A Software Engineering Framework for Achieving Long-Lasting Complex Systems. *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*. Published online 2016:246-255. doi:10.1109/RE.2016.15

39. Shahin M. Architecting for DevOps and Continuous Deployment. *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference on - ASWEC ' 15 Vol II*. Published online

2015:147-148. doi:10.1145/2811681.2824996

40. Kitchenham B. Procedures for Performing Systematic Reviews. Published online 2004:33.

41. Okoli C. A guide to conducting a standalone systematic literature review. *Communications of the Association for Information Systems*. 2015;37(1):879-910. doi:10.17705/1cais.03743

42. Thomas G. *How to Do Your Case Study*. 2nd ed. (Seaman J, Piper J, eds.). Sage Publications Asia-Pacific PTE LTD; 2016.

43. Webster J, Watson RT. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*. 2002;26(2):xiii-xxiii. doi:10.1.1.104.6570

44. Veres O, Kunanets N, Pasichnyk V, Veretennikova N, Korz R, Leheza A. Development and Operations - The Modern Paradigm of the Work of IT Project Teams. *IEEE 2019 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2019 - Proceedings*. 2019;3:103-106. doi:10.1109/STC-CSIT.2019.8929861

45. Sousa L, Trigo A, Varajão J. DevOps – Foundations and perspectives [DevOps – Fundamentos e perspetivas]. *Atas da Conferencia da Associacao Portuguesa de Sistemas de Informacao*. 2019;(October).

46. Céspedes D, Angeleri P, Melendez K, Dávila A. Software Product Quality in DevOps Contexts: A Systematic Literature Review. In: Mejia J, Muñoz M, Rocha Á, A. Calvo-Manzano J, eds. Advances in Intelligent Systems and Computing. Springer International Publishing; 2020:51-64. doi:10.1007/978-3-030-33547-2_5

47. Ottosson S. Agile Principles and Innovation Development Stig. *Developing and Managing Innovation in a Fast Changing and Complex World: Benefiting from Dynamic Principles*. 2018;(March):1-281. doi:10.1007/978-3-319-94045-8

48. Wahaballa A, Wahballa O, Abdellatief M, Xiong H, Qin Z. Toward Unified DevOps Model. In: *6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. ; 2015:1-4. doi:10.1109/ICSESS.2015.7339039

49. Beigi-Mohammadi N, Litoiu M, Emami-Taba M, et al. A DevOps Framework for Quality-Driven Self-Protection in Web Software Systems A DevOps Framework for Quality-Driven Self-Protection in Web Software. *Systems*. 2018;(October):270-274. doi:10.1145/nnnnnnn.nnnnnnn

50. Roche J. Adopting DevOps practices in quality assurance. *Communications of the ACM*. Published online 2013:38-43. doi:10.1145/2524713.2524721

51. Wettinger J, Andrikopoulos V, Leymann F. Enabling DevOps Collaboration and Continuous Delivery Using Diverse Application. In: *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*. Vol 9415. ; 2015:348-358. doi:10.1007/978-3-319-26148-5

52. Schaefer A, Reichenbach M, Fey D. Continuous Integration and Automation for DevOps. *IAENG Transactions on Engineering Technologies*. Published online 2012. doi:10.1007/978-94-007-4786-9

53. Díaz J, Almaraz R, Pérez J, Garbajosa J. DevOps in practice - An exploratory case study. *ACM International Conference Proceeding Series*. Published online 2018:18-20. doi:10.1145/3234152.3234199

54. Sánchez-Gordón M, Colomo-Palacios R. Characterizing DevOps Culture: A Systematic Literature Review. In: Stamelos I, O'Connor R V, Rout T, Dorling A, eds. Communications in Computer and Information Science. Springer International Publishing; 2018:3-15. doi:10.1007/978-3-030-00623-5_1

55. Debroy V, Miller S, Brimble L. Building lean continuous integration and delivery pipelines by applying devops principles: A case study at varidesk. *ESEC/FSE 2018 - Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the*

Foundations of Software Engineering. Published online 2018:851-856. doi:10.1145/3236024.3275528

56. Kuusinen K, Balakumar V, Jepsen SC, et al. A Large Agile Organization on Its Journey Towards DevOps. In: *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE; 2018:60-63. doi:10.1109/SEAA.2018.00019
57. Machiraju S, Gaurav S. *DevOps for Azure Applications*. 1st ed. (Srivastava S, Moodie M, Modi D, eds.). Apress; 2018. doi:10.1007/978-1-4842-3643-7
58. Atwal H. DevOps for DataOps. In: *Practical DataOps*. Apress; 2019:161-189. doi:https://doi.org/10.1007/978-1-4842-5104-1_7
59. Forsgren N, Humble J, Kim G. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press; 2018.
60. Bruneo D, Fritz T, Keidar-Barner S, et al. CloudWave: Where adaptive cloud management meets DevOps. In: *2014 IEEE Symposium on Computers and Communications (ISCC)*. Vol Workshops. ; 2014:1-6. doi:10.1109/ISCC.2014.6912638
61. Perez-Palacin D, Ridene Y, Merseguer J. Quality Assessment in DevOps. Published online 2017:133-138. doi:10.1145/3053600.3053632
62. Riti P. Cloud and DevOps. In: *Practical Scala DSLs*. 1st ed. Apress; 2018:209-220. doi:10.1007/978-1-4842-3036-7_11
63. Elberzhager F, Arif T, Naab M, Süß I, Koban S. From agile development to devops: Going towards faster releases at high quality - Experiences from an industrial context. *Lecture Notes in Business Information Processing*. 2017;269:33-44. doi:10.1007/978-3-319-49421-0_3
64. Sahid A, Maleh Y, Belaissaoui M. An agile framework for ITS management in organizations. A case study based on DevOps. In: *ICCWCS'17 : Proceeding of the 2nd Edition of the International Conference on Computing and Wireless Communication Systems*. ; 2017:8. doi:10.1145/3167486.3167556
65. de França BBN, Jeronimo H, Travassos GH. Characterizing DevOps by hearing multiple voices. *ACM International Conference Proceeding Series*. Published online 2016:53-62. doi:10.1145/2973839.2973845
66. Forsgren N, Humble J, Kim G. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press; 2018.
67. Ravichandran A, Taylor K, Waterhouse P. DevOps and Real World ROI. In: Ravichandran A, Taylor K, Waterhouse P, eds. *DevOps for Digital Leaders: Reignite Business with a Modern DevOps-Enabled Software Factory*. Apress; 2016:139-150.
68. Virmani M. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *5th International Conference on Innovative Computing Technology, INTECH 2015*. 2015;(Intech):78-82. doi:10.1109/INTECH.2015.7173368
69. Díaz J, Perez JE, Yague A, Villegas A, de Antona A. Devops in practice - A preliminary analysis of two multinational companies. *arXiv*. Published online 2019:1-8. doi:10.1007/978-3-030-35333-9_23
70. Nybom K, Smeds J, Porres I. On the Impact of Mixing Responsibilities Between Devs and Ops. In: Sharp H, Hall T, eds. Lecture Notes in Business Information Processing. Springer International Publishing; 2016:131-143. doi:10.1007/978-3-319-33515-5_11
71. Siebra C, Lacerda R, Cerqueira I, et al. Empowering Continuous Delivery in Software Development: The DevOps Strategy. In: van Sinderen M, Maciaszek LA, eds. *Software Technologies*. Communications in Computer and Information Science. Springer International Publishing; 2019:247-265. doi:10.1007/978-3-030-29157-0_11

72. Benguria G, Alonso J, Etxaniz I, Orue-Echevarria L, Escalante M. Agile Development and Operation of Complex Systems in Multi-technology and Multi-company Environments: Following a DevOps Approach. *Communications in Computer and Information Science*. 2018;896:15-27. doi:10.1007/978-3-319-97925-0_2

73. Angara J, Gutta S, Prasad S. DevOps with Continuous Testing Architecture and Its Metrics Model. In: Sa PK, Bakshi S, Hatzilygeroudis IK, Sahoo MN, eds. Advances in Intelligent Systems and Computing. Springer; 2018:271-281. doi:10.1007/978-981-10-8633-5_28

74. Kaiser AK. *Reinventing ITIL® in the Age of DevOps*. Apress; 2018. doi:10.1007/978-1-4842-3976-6

75. Cuppett MS. DevOps, DBAs, and DBaaS. In: *DevOps, DBAs, and DBaaS*. ; 2016:73-85. doi:10.1007/978-1-4842-2208-9

76. Albuquerque AB, Cruz VL. Implementing DevOps in Legacy Systems. In: Silhavy R, Silhavy P, Prokopova Z, eds. Advances in Intelligent Systems and Computing. Springer International Publishing; 2019:143-161. doi:10.1007/978-3-030-00184-1_14

77. Baudry B, Harrand N, Schulte E, et al. A spoonful of DevOps helps the GI go down. *Proceedings - International Conference on Software Engineering*. Published online 2018:35-36. doi:10.1145/3194810.3194818

78. Lai ST, Leu FY. *A Micro Services Quality Measurement Model for Improving the Efficiency and Quality of DevOps*. Vol 773. Springer International Publishing; 2019. doi:10.1007/978-3-319-93554-6_55

79. Mala DJ, Reynold, Pradeep A. Towards Green Software Testing in Agile and DevOps Using Cloud Virtualization for Environmental Protection. In: *Software Engineering in the Era of Cloud Computing*. Springer, Cham; 2020:277-297. doi:https://doi.org/10.1007/978-3-030-33624-0_11

80. Masombuka T, Mnkandla E. A DevOps collaboration culture acceptance model. *ACM International Conference Proceeding Series*. Published online 2018:279-285. doi:10.1145/3278681.3278714

81. Kaiser AK. *Reinventing ITIL® in the Age of DevOps*. Apress; 2018. doi:10.1007/978-1-4842-3976-6

82. Jaatun MG, Cruzes DS, Luna J. DevOps for better software security in the cloud. *ACM International Conference Proceeding Series*. 2017;Part F1305. doi:10.1145/3098954.3103172

83. Jaatun MG. Software security activities that support incident management in secure DevOps. *ACM International Conference Proceeding Series*. Published online 2018. doi:10.1145/3230833.3233275

84. Rahman AAU, Williams L. Software security in DevOps: Synthesizing practitioners' perceptions and practices. *Proceedings - International Workshop on Continuous Software Evolution and Delivery, CSED 2016*. Published online 2016:70-76. doi:10.1145/2896941.2896946

85. Kim G, Debois P, Willis J, Humble J. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press; 2016.

86. Šmite D, Gonzalez-Huerta J, Moe NB. "When in Rome, Do as the Romans Do": Cultural Barriers to Being Agile in Distributed Teams. In: Stray V, Hoda R, Paasivaara M, Kruchten P, eds. *Agile Processes in Software Engineering and Extreme Programming*. Lecture {Notes} in {Business} {Information} {Processing}. Springer International Publishing; 2020:145-161. doi:10.1007/978-3-030-49392-9_10

87. Colomo-Palacios R, Fernandes E, Soto-Acosta P, Larrucea X. A case analysis of enabling continuous software deployment through knowledge management. *International Journal of Information Management*. 2018;40:186-189. doi:10.1016/j.ijinfomgt.2017.11.005

88. De Sanctis M, Bucchiarone A, Trubiani C. A DevOps Perspective for QoS-Aware Adaptive Applications. In: Bruel JM, Mazzara M, Meyer B, eds. *Software Engineering Aspects of Continuous*

Development and New Paradigms of Software Production and Deployment. Lecture {Notes} in {Computer} {Science}. Springer International Publishing; 2020:95-111. doi:10.1007/978-3-030-39306-9_7

89. Sahid A, Maleh Y, Belaissaoui M. An agile framework for ITS management in organizations. A case study based on DevOps. In: *ICCWCS'17 : Proceeding of the 2nd Edition of the International Conference on Computing and Wireless Communication Systems*. ; 2017:8. doi:10.1145/3167486.3167556

90. Šćekić M, Gazivoda M, Šćepanović S, Nikolić J. Application of DevOps approach in developing business intelligence system in bank. In: *2018 7th Mediterranean Conference} on Embedded Computing} (MECO)*. ; 2018:1-4. doi:10.1109/MECO.2018.8406047

91. Mohan V, Ben Othmane L, Kres A. BP: Security Concerns and Best Practices for Automation of Software Deployment Processes: An Industrial Case Study. In: ; 2018:21-28. doi:10.1109/SecDev.2018.00011

92. Gupta RK, Venkatachalapathy M, Jeberla FK. Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization. In: *Proceedings - 2019 ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE 2019*. IEEE; 2019:30-34. doi:10.1109/ICGSE.2019.00020

93. Sampedro Z, Holt A, Hauser T. Continuous Integration and Delivery for HPC: Using Singularity and Jenkins. In: *Proceedings of the Practice and Experience on Advanced Research Computing*. {PEARC} '18. Association for Computing Machinery; 2018:1-6. doi:10.1145/3219104.3219147

94. Riungu-Kalliosaari L, Mäkinen S, Lwakatare LE, Tiihonen J, Männistö T. DevOps adoption benefits and challenges in practice: A case study. In: *17th Internacional Conference, PROFES 2016, Throndheim, Norway*. Vol 10027 LNCS. ; 2016:590-597. doi:10.1007/978-3-319-49094-6_44

95. Senapathi M, Buchan J, Osman H. DevOps capabilities, practices, and challenges: Insights from a case study. *ACM International Conference Proceeding Series*. 2018;Part F1377. doi:10.1145/3210459.3210465

96. Heistand C, Thomas J, Tzeng N, et al. DevOps for Spacecraft Flight Software. *IEEE Aerospace Conference Proceedings*. 2019;2019-March:1-16. doi:10.1109/AERO.2019.8742143

97. Ghantous GB, Gill AQ. DevOps Reference Architecture for Multi-cloud IOT Applications. In: *2018 IEEE 20th Conference on Business Informatics (CBI)*. Vol 01. ; 2018:158-167. doi:10.1109/CBI.2018.00026

98. Smeds J, Nybom K, Porres I. DevOps: A Definition and Perceived Adoption Impediments. In: Lassenius C, Dingsøyr T, Paasivaara M, eds. *Agile Processes in Software Engineering and Extreme Programming*. Lecture {Notes} in {Business} {Information} {Processing}. Springer International Publishing; 2015:166-177. doi:10.1007/978-3-319-18612-2_14

99. Jiménez M, Rivera LF, Villegas NM, Tamura G, Müller HA, Gallego P. DevOps' Shift-Left in Practice: An Industrial Case of Application. In: Bruel JM, Mazzara M, Meyer B, eds. Lecture Notes in Computer Science. Springer International Publishing; 2019:205-220. doi:10.1007/978-3-030-06019-0_16

100. Hemon A, Lyonnet B, Rowe F, Fitzgerald B. From Agile to DevOps: Smart Skills and Collaborations. *Inf Syst Front*. 2020;22(4):927-945. doi:10.1007/s10796-019-09905-1

101. AL-Zahran S, Fakieh B. How DevOps Practices Support Digital Transformation. *International Journal of Advanced Trends in Computer Science and Engineering*. 2020;9:2780-2788. doi:10.30534/ijatcse/2020/46932020

102. Hemon-Hildgen A, Fitzgerald B, Lyonnet B, Rowe F. Innovative Practices for Knowledge Sharing in Large-Scale DevOps. *IEEE Software*. 2019;PP. doi:10.1109/MS.2019.2958900

103. Gall M, Pigni F. Leveraging DevOps for mission critical software. In: *AMCIS*. ; 2018. Accessed February 7, 2022. https://aisel.aisnet.org/amcis2018/ITProjMgmt/Presentations/2

104. Jones S, Noppen J, Lettice F. Management challenges for devops adoption within UK SMEs. *QUDOS 2016 - Proceedings of the 2nd International Workshop on Quality-Aware DevOps, co-located with ISSTA 2016*. Published online 2016:7-11. doi:10.1145/2945408.2945410

105. Díaz J, Pérez JE, Lopez-Peña MA, Mena GA, Yagüe A. Self-Service Cybersecurity Monitoring as Enabler for DevSecOps. *IEEE Access*. 2019;7:100283-100295. doi:10.1109/ACCESS.2019.2930000

106. Bruza M. An Analysis of Multi-domain Command and Control and the Development of Software Solutions through DevOps Toolsets and Practices. Published online 2018.

107. Shahin M, Babar MA, Zhu L. The Intersection of Continuous Deployment and Architecting Process: Practitioners' Perspectives. In: *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. {ESEM} '16. Association for Computing Machinery; 2016:1-10. doi:10.1145/2961111.2962587

108. Laukkanen E, Paasivaara M, Itkonen J, Lassenius C, Arvonen T. Towards Continuous Delivery by Reducing the Feature Freeze Period: A Case Study. In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. ; 2017:23-32. doi:10.1109/ICSE-SEIP.2017.21

109. Lwakatare LE, Kuvaja P, Oivo M. An Exploratory Study of DevOps: Extending the Dimensions of DevOps with Practices. In: *The Eleventh International Conference on Software Engineering Advances (ICSEA)*. ; 2016.

110. Alt R, Auth G, Kögler C. Transformation of Consulting for Software-Defined Businesses: Lessons from a DevOps Case Study in a German IT Company. In: *Advances in Consulting Research*. Springer, Cham; 2018:385-403. doi:https://doi.org/10.1007/978-3-319-95999-3_19

111. Düllmann TF, Paule C, Hoorn A v. Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines. In: *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*. ; 2018:27-30.

112. Laukkarinen T, Kuusinen K, Mikkonen T. DevOps in regulated software development: Case medical devices. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Results Track, ICSE-NIER 2017*. Published online 2017:15-18. doi:10.1109/ICSE-NIER.2017.20

113. Zhou X, Jin Y, Zhang H, Li S, Huang X. A map of threats to validity of systematic literature reviews in software engineering. In: *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*. Vol 0. ; 2016:153-160. doi:10.1109/APSEC.2016.031

114. Erich FMA, Amrit C, Daneva M. A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*. 2017;29(6):1-20. doi:10.1002/smr.1885

115. Metzger S, Durden D, Sturtevant C, et al. eddy4R 0.2.0: a DevOps model for community-extensible processing and analysis of eddy-covariance data based on R, Git, Docker, and HDF5. *Geoscientific Model Development*. 2017;10(9):3189-3206. doi:10.5194/gmd-10-3189-2017

116. Gupta V, Kapur PK, Kumar D. Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling. *Information and Software Technology*. 2017;92:75-91. doi:10.1016/j.infsof.2017.07.010

117. Chen B. Improving the software logging practices in devops. In: *Proceedings - 2019 IEEE/ACM 41st*

International Conference on Software Engineering: Companion, ICSE-Companion 2019. IEEE; 2019:194-197. doi:10.1109/ICSE-Companion.2019.00080

118. Barna C, Khazaei H, Fokaefs M, Litoiu M. Delivering Elastic Containerized Cloud Applications to Enable DevOps. *Proceedings - 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2017*. Published online 2017:65-75. doi:10.1109/SEAMS.2017.12
119. Olszewska M, Waldén M. DevOps meets formal modelling in high-criticality complex systems. *1st International Workshop on Quality-Aware DevOps, QUDOS 2015 - Proceedings*. Published online 2015:7-12. doi:10.1145/2804371.2804373
120. Muñoz M, Negrete M. Reinforcing DevOps Generic Process with a Guidance Based on the Basic Profile of ISO/IEC 29110. In: Mejia J, Muñoz M, Rocha Á, A. Calvo-Manzano J, eds. Advances in Intelligent Systems and Computing. Springer International Publishing; 2020:65-79. doi:10.1007/978-3-030-33547-2_6
121. Cruzes DS, Melsnes K, Marczak S. Testing in a DevOps Era: Perceptions of Testers in Norwegian Organisations. In: Misra S, Gervasi O, Murgante B, et al., eds. Lecture Notes in Computer Science. Springer International Publishing; 2019:442-455. doi:10.1007/978-3-030-24305-0_33
122. Daneva M, Bolscher R. What We Know About Software Architecture Styles in Continuous Delivery and DevOps? In: van Sinderen M, Maciaszek LA, eds. Communications in Computer and Information Science. Springer International Publishing; 2020:26-39. doi:10.1007/978-3-030-52991-8_2
123. Scheaffer J, Ravichandran A, Martins A. *The Kitty Hawk Venture*. 1st ed. (McDermott S, Berendson L, Fernando R, eds.). Apress; 2018. doi:10.1007/978-1-4842-3661-1
124. Hüttermann M. *DevOps for Developers*. 1st ed. Apress; 2012. doi:10.1007/978-1-4302-4570-4
125. Davis A. DevOps. In: *Mastering Salesforce DevOps*. Apress; 2019. doi:https://doi.org/10.1007/978-1-4842-5473-8_3
126. Kuruba M. DevOps for IT Service Reliability and Availability. In: Karanki D, Vinod G, Ajit S, eds. *Advances in RAMS Engineering*. 1st ed. Apress; 2019:149-183. doi:https://doi.org/10.1007/978-3-030-36518-9_6
127. Familiar B, Barnes J. DevOps Using PowerShell, ARM, and VSTS. In: *Business in Real-Time Using Azure IoT and Cortana Intelligence Suite*. 1st ed. Apress; 2017:21-93. doi:https://doi.org/10.1007/978-1-4842-2650-6_2
128. Muñoz M, Díaz O. DevOps: Foundations and Its Utilization in Data Center. In: J. MG, M. M, M. R, W. N, R O, eds. *Engineering and Management of Data Centers*. Springer, Cham; 2017:205-225. doi:https://doi.org/10.1007/978-3-319-65082-1_10
129. Nagarajan AD, Overbeek SJ. A DevOps Implementation Framework for Large Agile-Based Financial Organizations. In: H. P, C. D, H. P, C. A, D. R, R. M, eds. *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*. Springer, Cham; 2018:1-17. doi:https://doi.org/10.1007/978-3-030-02610-3_10
130. Wang W, Casale G, Iuhasz G. Closing the Loop Between Ops and Dev. In: Di Nitto E, ed. *Model-Driven Development and Operationof Multi-Cloud Applications*. Springer, Cham; 2017:95-105. doi:https://doi.org/10.1007/978-3-319-46031-4_10
131. de Feijter R, Overbeek S, van Vliet R, Jagroep E, Brinkkemper S. DevOps Competences and Maturity for Software Producing Organizations. In: Gulden J, Reinhartz-Berger I, Schmidt R, Guerreiro S, Guédria W, Bera P, eds. Lecture Notes in Business Information Processing. Springer International Publishing; 2018:244-259. doi:10.1007/978-3-319-91704-7_16

132. Cito J, Wettinger J, Lwakatare LE, Borg M, Li F. Feedback from Operations to Software Development—A DevOps Perspective on Runtime Metrics and Logs. In: Bruel JM, Mazzara M, Meyer B, eds. Lecture Notes in Computer Science. Springer International Publishing; 2019:184-195. doi:10.1007/978-3-030-06019-0_14

133. Alban D, Eynaud P, Malaurent J, Richet JL, Vitari C. *Information Systems Management: Governance, Urbanization and Alignment*. ISTE Ltd and John Wiley & Sons, Inc; 2019. doi:10.1016/B978-012226570-9/50107-1

134. Beulen E. Implementing and Contracting Agile and DevOps: A Survey in the Netherlands. In: Kotlarsky J, Oshri I, Willcocks L, eds. Lecture Notes in Business Information Processing. Springer International Publishing; 2019:124-146. doi:10.1007/978-3-030-15850-7_7

135. Colavita F. DevOps Movement of Enterprise Agile Breakdown Silos, Create Collaboration, Increase Quality, and Application Speed. In: Ciancarini P, Sillitti A, Succi G, Messina A, eds. Advances in Intelligent Systems and Computing. Springer International Publishing; 2016:203-213. doi:10.1007/978-3-319-27896-4_17

136. Fishman N, Stryker C. *Smarter Data Science*. First. (Minatel J, ed.). John Wiley & Sons, Inc; 2020. doi:https://doi.org/10.1002/9781119697985

137. Hüttermann M. *DevOps for Developers*. 1st ed. Apress; 2012. doi:10.1007/978-1-4302-4570-4

138. Johng H, Kalia AK, Xiao J, Vuković M, Chung L. Harmonia: A Continuous Service Monitoring Framework Using DevOps and Service Mesh in a Complementary Manner. In: Yangui S, Bouassida Rodriguez I, Drira K, Tari Z, eds. *Service-Oriented Computing*. Lecture Notes in Computer Science. Springer International Publishing; 2019:151-168. doi:10.1007/978-3-030-33702-5_12

139. Jones S, Noppen J, Lettice F. Management challenges for DevOps adoption within UK SMEs. In: ; 2016:7-11. doi:10.1145/2945408.2945410

140. Kang H, Le M, Tao S. Container and Microservice Driven Design for Cloud Infrastructure DevOps. In: *2016 IEEE International Conference on Cloud Engineering (IC2E)*. ; 2016:202-211. doi:10.1109/IC2E.2016.26

141. Lewerentz M, Bluhm T, Daher R, et al. Implementing DevOps practices at the control and data acquisition system of an experimental fusion device. *Fusion Engineering and Design*. 2018;(October):0-1. doi:10.1016/j.fusengdes.2018.11.022

142. Lewis B, Smith I, Fowler M, Licato J. Enabling DevOps for Containerized Data-Intensive Applications: An Exploratory Study. In: *28th Modern Artificial Intelligence and Cognitive Science Conference, MAICS 2017*. ; 2017:189-190. doi:10.1145/1235

143. Palermo J. *..NET DevOps for Azure*. 1st ed. (Murray J, Berendson L, Balzano J, eds.). Apress; 2019. doi:10.1007/978-1-4842-5343-4

144. Ravichandran A, Taylor K, Waterhouse P. DevOps for Digital Leaders. *DevOps for Digital Leaders*. Published online 2016. doi:10.1007/978-1-4842-1842-6

145. Rong G, Zhang H, Shao D. CMMI guided process improvement for DevOps projects. *Proceedings of the International Workshop on Software and Systems Process - ICSSP '16*. Published online 2016:76-85. doi:10.1145/2904354.2904372

146. Senapathi M, Buchan J, Osman H. DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. *EASE*. Published online 2018. doi:10.1145/3210459.3210465

147. Tomlinson T. Drupal 8 DevOps. In: *Enterprise Drupal 8 Development*. Apress; 2017:271-279. doi:https://doi.org/10.1007/978-1-4842-0253-1_11

148. Wettinger J, Breitenbücher U, Falkenthal M, Leymann F. Collaborative gathering and continuous

delivery of DevOps solutions through repositories. *Computer Science - Research and Development*. 2017;32(3-4):281-290. doi:10.1007/s00450-016-0338-z

149. Wiedemann A, Wiesche M, Krcmar H. Integrating Development and Operations in Cross-Functional Teams - Toward a DevOps Competency Model. In: *Proceedings of the 2019 on Computers and People Research Conference*. {SIGMIS}-{CPR} '19. Association for Computing Machinery; 2019:14-19. doi:10.1145/3322385.3322400

150. Ali N, Daneth H, Hong JE. A hybrid DevOps process supporting software reuse: A pilot project. *Journal of Software: Evolution and Process*. 2020;32(7):1-23. doi:10.1002/smr.2248

151. Gotimer G, Stiehm T. DevOps advantages for testing: Increasing quality through continuous delivery. *CrossTalk*. 2016;29(3):13-18.

152. Morales JA, Yasar H, Volkman A. Implementing DevOps practices in highly regulated environments. *ACM International Conference Proceeding Series*. 2018;Part F1477. doi:10.1145/3234152.3234188

153. Pérez JF, Wang W, Casale G. Towards a DevOps approach for software quality engineering. *WOSP-C 2015 - Proceedings of the 2015 ACM/SPEC Workshop on Challenges in Performance Methods for Software Development, in Conjunction with ICPE 2015*. Published online 2015:5-10. doi:10.1145/2693561.2693564

154. de Kort W. *DevOps on the Microsoft Stack*. 1st ed. (DeWolf J, Pundick D, eds.). Apress; 2016. doi:10.1007/978-1-4842-1446-6

155. Rittgen P, Cronholm S, Göbel H. Towards a Model for Assessing Collaboration Capability Between Development and Operations. In: Walker A, O'Connor R V, Messnarz R, eds. Communications in Computer and Information Science. Springer International Publishing; 2019:111-122. doi:10.1007/978-3-030-28005-5_9

156. Murphy GC, Kersten M. Towards Bridging the Value Gap in DevOps. In: Bruel JM, Mazzara M, Meyer B, eds. Lecture Notes in Computer Science. Springer International Publishing; 2020:181-190. doi:10.1007/978-3-030-39306-9_13

157. Arulkumar V, Lathamanju R. Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method. *Procedia Computer Science*. 2019;165(2019):399-405. doi:10.1016/j.procs.2020.01.032

158. Medina O, Schumann E. *DevOps for SharePoint*. 1st ed. (Murray J, Berendson L, Balzano J, eds.). Apress; 2018. doi:10.1007/978-1-4842-3688-8

159. Koilada DK. Business model innovation using modern DevOps. *2019 IEEE Technology and Engineering Management Conference, TEMSCON 2019*. 2019;(Cd):1-6. doi:10.1109/TEMSCON.2019.8813557

160. Rong G, Jin Z, Zhang H, Zhang Y, Ye W, Shao D. DevDocOps: Enabling continuous documentation in alignment with DevOps. *Software - Practice and Experience*. 2019;50(3):210-226. doi:10.1002/spe.2770

161. Forsgren N, Kersten MIK. DevOps Metrics. *ACM Queue*. 2017;(december):1-16.

162. Cukier D. DevOps patterns to scale web applications using cloud services. *SPLASH 2013 - Proceedings of the 2013 Companion Publication for Conference on Systems, Programming, and Applications: Software for Humanity*. 2013;(Figure 2):143-152. doi:10.1145/2508075.2508432

163. Combemale B, Wimmer M. Towards a Model-Based DevOps for Cyber-Physical Systems. In: Bruel JM, Mazzara M, Meyer B, eds. Lecture Notes in Computer Science. Springer International Publishing; 2020:84-94. doi:10.1007/978-3-030-39306-9_6

Appendix A. Concept-Centric Approach for Benefits and Literature

This appendix provides a mapping between all the authors (references) that mention a certain DevOps benefit.

Concept ID	Reference
B01	6,12,15,17,22,25,28,29,44–46,49–51,56,57,62–65,67,69,70,73,76,80,84,92,95,110,114–132
B02	6,12,14,17,22,25,39,51–54,56,58,63,65,75,79,94,116,125,126,128,129,132–149
B03	6,14,15,22,23,39,44,45,49,51,53–56,58,60,63,67,73,75,76,96,112,114,120,121,124,125,128,136,143,150–156
B04	14,15,20,22,28,29,45,51,57,65,71,76–78,95,96,110,112,116,121–123,125,128,131,135,138,151,154,156–158
B05	6,17,22,28,29,45,50,56,58,60,65,69,92,95,104,121,123–125,128,136,138,143,159–161
B06	6,14,22,46,57,61,64,65,68,71,96,110,119,121,122,124–126,133,153,158,162
B07	6,14,23,45,46,56,62–64,67,69,83,125,126,140,147,151
B08	12,22,28,29,44,45,53,57,63–65,70,116,138,150,157,159
B09	22,28,51,65,67,68,75,110,128,132,133,148,151
B10	22,65,68–72,112,127,128,145,150,152
B11	15,22,23,25,29,54,65,67,70,95,120,121,126
B12	22,56,65,68,71,73,75,77,81,120,154
B13	22,29,51,55,56,58,64,65,78,116,136
B14	22,46,58,65,67,75,76,79,121,147,163
B15	12,15,54,63,70,80,95,116,124
B16	14,29,120,136,141,156,159,163
B17	56,63,65,70,81
B18	82–84,125,138
B19	15,77

Appendix B. Identified Business Benefits per DevOps Implementation Case Study

In this appendix there is the mapping between the case studies and the benefits identified on each of these case studies.

ID	Case Study Number	DevOps Benefit Concept ID
CS.1	1	B01
CS.2	2	B02; B04; B17
CS 3	3	B01; B06
CS.4	4	B11
CS.5	5	B01; B02; B08; B10; B17
CS.6	6	B01; B03; B13; B04; B06; B10; B12;
CS.7	7	B01; B02; B06; B10; B17
CS.8	8	B01; B02; B06; B10; B18; B17;
CS.9	9	B01; B04; B06; B16
CS.10	10	B01; B02; B03; B10; B17;
CS.11	11	B04; B06
CS.12	12.1	B01; B02; B03; B04; B05; B06; B07; B08; B10; B12; B14; B15; B17
	12.2	B01; B02; B03; B04; B05; B06; B07; B08; B10; B12; B14; B15; B17
	12.3	B01; B02; B03; B04; B05; B06; B07; B08; B10; B12; B14; B15; B17
CS.13	13	B01; B04; B05; B10; B13; B15; B16
CS.14	14	B06; B17
CS.15	15.1	B02; B06; B07; B17
	15.2	B02; B06; B07; B17
CS.16	16.1	B02; B03; B08; B09; B12; B13
	16.2	B02; B03; B08; B09; B12; B13
	16.3	B02; B03; B08; B09; B12; B13
	16.4	B02; B03; B08; B09; B12; B13
	16.5	B02; B03; B08; B09; B12; B13
	16.6	B02; B03; B08; B09; B12; B13
	16.7	B02; B03; B08; B09; B12; B13
	16.8	B02; B03; B08; B09; B12; B13
	16.9	B02; B03; B08; B09; B12; B13
	16.10	B02; B03; B08; B09; B12; B13
	16.11	B02; B03; B08; B09; B12; B13
CS.17	17.1	B4; B6; B7; B9; B10; B11; B17; B19
	17.2	B02; B04; B08; B15; B19
	17.3	B01; B02; B04; B06; B07; B08; B15; B19
	17.4	B02; B04; B06; B07; B08; B15; B19
	17.5	B02; B04; B08; B15; B19
CS.18	18	B01; B02; B04; B08; B12
CS.19	19	B01
CS.20	20	B04; B06
CS.21	21	No benefits identified
CS.22	22	B01; B03; B08; B11
CS.23	23.1	B02; B06; B11; B13; B15; B16; B17
	23.2	B02; B03; B06; B11; B13; B16; B15
CS.24	24	B01; B02; B03; B04; B06; B07; B12 B13; B17
CS.25	25	B01; B08; B11; B12; B15; B19
CS.26	26.1	B01; B08; B11; B12; B15

	26.2	B01; B08; B11; B12; B15
	26.3	B01; B08; B11; B12; B15
	26.4	B01; B08; B11; B12; B15
CS.27	27	B01; B02
CS.28	28	B01
CS.29	29	B01; B06; B11; B12; B14; B15; B19
CS.30	30	B01; B06; B07; B08; B12; B19
CS.31	31	B01; B02; B03; B04; B05; B06; B08; B13; B15; B16; B17; B18
CS.32	32.1	B01; B02; B04; B10; B11; B12; B17
	32.2	B01; B02; B04; B10; B11; B12; B17
	32.3	B01; B02; B04; B10; B11; B12; B17
	32.4	B01; B02; B04; B10; B11; B12; B17
	32.5	B01; B02; B04; B10; B11; B12; B17
	32.6	B01; B02; B04; B10; B11; B12; B17
	32.7	B01; B02; B04; B10; B11; B12; B17
	32.8	B01; B02; B04; B10; B11; B12; B17
	32.9	B01; B02; B04; B10; B11; B12; B17
	32.10	B01; B02; B04; B10; B11; B12; B17
CS.33	33	B02; B06; B14
CS.34	34.1	B02; B12
	34.2	B02; B12
	34.3	B02; B12
	34.4	B02; B12
CS.35	35	B01; B02; B03; B05; B13; B16
CS.36	36	B02; B03; B08; B09; B12; B13; B19