UNIVERSITY OF HELSINKI

https://helda.helsinki.fi

# DevOps Adoption Benefits and Challenges in Practice: A Case Study

**Riungu-Kalliosaari, Leah; Mäkinen, Simo; Lwakatare, Lucy Ellen; Tiihonen, Juha; Männistö, Tomi**

**Abrahamsson, Pekka; Jedlitschka, Andreas; Nguyen Duc, Anh; Felderer, Michael; Amasaki, Sousuke ...**

**2016-11**

http://hdl.handle.net/10138/178485

Riungu-Kalliosaari, L, Mäkinen, S, Lwakatare, L E, Tiihonen, J & Männistö, T 2016, DevOps Adoption Benefits and Challenges in Practice: A Case Study. in P Abrahamsson, A Jedlitschka, A Nguyen Duc, M Felderer, S Amasaki & T Mikkonen (eds), Product-Focused Software Process Improvement : 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10027, Springer International Publishing , pp. 590-597, Product-Focused Software Process Improvement, International Conference, Trondheim, Norway, 22/11 /2016. https://doi.org/10.1007/978-3-319-49094-6_44

Downloaded from Helda, University of Helsinki institutional repository. https://helda.helsinki.fi
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.
Please cite the original version.

Author's post-print. Cite as:
Riungu-Kalliosaari, Leah; Mäkinen, Simo; Lwakatare, Lucy Ellen; Tiihonen, Juha; Männistö, Tomi.
DevOps Adoption Benefits and Challenges in Practice: A Case Study.
Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway,
November 22-24, 2016, Proceedings, LNCS 10027, pp. 590-597. Springer International Publishing, 2016.
The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-49094-6_44

# DevOps Adoption Benefits and Challenges in Practice: A Case Study

Leah Riungu-Kalliosaari[1(✉)], Simo Mäkinen[1], Lucy Ellen Lwakatare[2], Juha Tiihonen[1], and Tomi Männistö[1]

[1] University of Helsinki
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 UNIVERSITY OF HELSINKI, Finland
[2] University of Oulu
Department of Information Processing Science
P. O. Box 3000
FI-90014 UNIVERSITY OF OULU, Finland

**Abstract.** DevOps is an approach in which traditional software engineering roles are merged and communication is enhanced to improve the production release frequency and maintain software quality. There seem to be benefits in adopting DevOps but practical industry experiences have seldom been reported. We conducted a qualitative multiple-case study and interviewed the representatives of three software development organizations in Finland. The responses indicate that with DevOps, practitioners can increase the frequency of releases and improve test automation practices. DevOps was seen to encourage collaboration between departments which boosts communication and employee welfare. Continuous releases enable a more experimental approach and rapid feedback collection. The challenges include communication structures that hinder cross-department collaboration and having to address the cultural shift. Dissimilar development and production environments were mentioned as some of the technical barriers. DevOps might not also be suitable for all industries. Ambiguity in the definition of DevOps makes adoption difficult since organizations might not know which practices they should implement for DevOps.

## 1 Introduction

Software and product development in the modern era of interconnected, highly available systems requires close collaboration between members of the development team. Software distribution has changed and recurring software updates can happen to the point that software releases are served at the same time as when software is being used [12]. Because of the nature and frequency of the releases, possible issues in the production systems need to be monitored closely in order to provide optimum user experience [12]. Software organizations need to adapt their practices to various changes brought about by new concepts such as DevOps.

Our working definition of DevOps is: "a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality" [3]. The DevOps phenomenon has two core principles: (1) emphasis on collaboration between development and operations; (2) the use of agile principles and automation to configure and manage deployment environments [8]. DevOps extends collaboration between development and operations teams which eases the handling of changes in the production environment.

Industry reports on the DevOps practices have been rare in the past [9], although more studies have surfaced lately. What potential do practitioners see in DevOps, and what kind of challenges are there in adopting the DevOps? To answer these questions, we performed a multiple-case study of three software companies in Finland. We conducted semi-structured interviews with representatives from the companies and report the results of our analysis in this article.

The article is structured as follows. Section 2 describes previously reported DevOps experiences, benefits and challenges. Section 3 presents the applied research method. Section 4 focuses on the results whereas Section 5 discusses the results and considers the validity threats. Section 6 concludes the work.

## 2 Related Work

Several recent studies have recognized the importance of DevOps. Developer and operations teams can streamline development processes in order to fine-tune the performance of services and increase scalability with virtualization [6]. Monitoring the production systems at real-time enables developers to react whenever anomalies are detected [2, 6, 11]. On-demand infrastructures and timely feedback from monitoring support continuous software delivery and deployment. Release cycles can shorten to hours instead of weeks and months [11], which is seen as a definite advantage of DevOps [4, 7]. Monolithic architectures restrict the release frequency [13] but microservice architectures help to break down components into smaller pieces suitable for frequent releases [2, 13].

Combining the expertise and knowledge of software experts from different functions can be challenging. The lack of cooperation between developers and operations personnel results in uncoordinated activities [9, 14]. This causes serious problems which include: (1) IT operations not being involved in the requirements specification, (2) poor communication and information flow, (3) unsatisfactory test environments, (4) lack of knowledge transfer, (5) immature systems, and (6) operational routines not being established prior to deployment [9].

Poor communication between the development and operations functions produces undesirable results. Non-functional requirements e.g. performance or availability might be overlooked as the responsibility of running the product is shifted to the operations team, letting developers off the hook [14]. Without proper access to production systems and error logs, developers become frustrated [14, 13]. Such challenges are only exacerbated if the development teams are distributed [13].

**Table 1.** Characteristics of the organizations involved in the case study

| Org. | Domain | Personnel | Unit Size | Interviewee Position |
|---|---|---|---|---|
| A | Software development and big data analytics | 90 | 20 | Senior Consultant 1 & 2 |
| B | Digital service development | 100 | 20 | Senior Developer |
| C | Software development and research | >900 | 10 | Lead Architect for Cloud |

## 3 Research Method

We employed an explorative, qualitative research approach in our case study. We wanted to find out how the industry sees DevOps along with its advantages and limitations. Our research questions are:

- RQ1: How do industry practitioners perceive the benefits of DevOps practices in their organization?
- RQ2: How do industry practitioners perceive the adoption challenges related to DevOps?

We targeted software development organizations with sufficient DevOps experience. Three Finnish development organizations were selected for the study. Two of them were participants in a joint industry-academia program called Need for Speed [1]. The sampling for the study can be considered as convenience sampling. Table 1 introduces the organizations.

Organization A is a consultancy company with a specific DevOps unit. Its customers' operations environments ranged from private servers and clouds to the application of public cloud services. In Organization B, developers work as consultants in customer projects, bringing their expertise as required. Organization C is an international organization with a technological expertise services unit dedicated to cloud platform services. It sees DevOps as feasible for areas with direct control of the operations environment.

Semi-structured interviews were the data collection method. The themes related to views on the DevOps phenomenon, along with its advantages and limitations. Two interviews were conducted on-site and one remotely. The interviews lasted from one to two hours. The interviews were recorded and transcribed.

Thematic analysis and synthesis [5] was used to support data analysis. The method can be used to find patterns in the data by assigning codes to segments of text, translating the codes into higher-order themes and finally creating a model of the themes [5]. The interview transcripts were coded and a set of themes created as a result which allowed to construct a model of the benefits and challenges of DevOps.

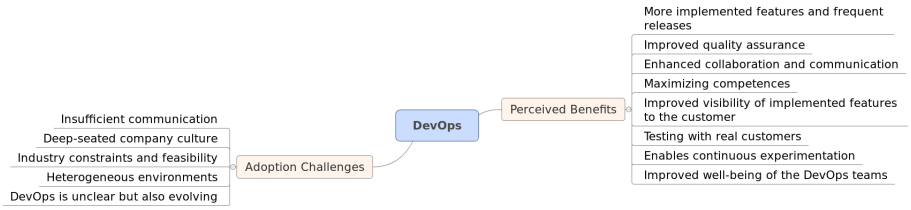



**Fig. 1.** Thematic map for the perceived benefits and adoption challenges of DevOps

## 4 Results

After analyzing the interview transcripts, we identified several themes. The higher-level themes considered were the benefits and challenges of adopting DevOps. Figure 1 illustrates the thematic map and the themes covered in this section.

### 4.1 Perceived Benefits of DevOps

The respondents thought that DevOps practices lead to a number of benefits. Broadly speaking, DevOps was seen to positively affect the pace at which software products could be delivered and the quality of products for instance by intensifying feedback cycles. The internal mechanics of engineering products in the organization including aspects such as communication were perceived to be affected, too.

As a definite advantage, the respondents highlighted *more implemented features and frequent releases*. Powered by automated build, testing and deployment processes, one of the main advantages of DevOps is that organizations are able to channel more features into the production and delivery pipelines as noted by an interviewed Lead Architect. Automation also reduces the effort required for setting up releases, making it possible for organizations to churn out the releases as frequently as required.

"There are multiple factors, relating to this but yes, definitely one of the key drivers of DevOps is actually that you can get more commits of code into each day basically." (Lead Architect, Org C)

Subsequently, 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. 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.

One of the major impacts of DevOps is that it forces the development and operations teams to interact with each other more than before, which was seen to lead to *enhanced collaboration and communication*. The walls between the traditional development versus operations silos are slowly broken down, and as a result encouraging a unified way of working. Increased collaboration accelerates

the exchange of knowledge and experiences between the teams. Multi-functional teams with a variety of capabilities that boost productivity result in *maximizing competences* since a broader set of the skills is utilized.

The respondents felt that when applying DevOps practices, the possibility to release frequently has implications to the whole development process. Shorter development cycles were seen as beneficial for the customers who can enjoy newly developed features faster than before. Smaller releases made more often promote *improved visibility of implemented features to the customer*. In addition, due to frequent releases, the development and operations teams are able to receive early feedback from the end users and able to do *testing with real customers*, which helps in improving the end products.

Working with real customers, companies are better equipped with knowledge on customer preferences and are ultimately able to tailor their products to meet the market demands. According to the perceptions of the interviewees, DevOps can help companies to test different ideas quickly and make decisions accordingly. The constant testing of hypotheses in order to determine the value gains for both the customer and the organization is referred to as continuous experimentation [10]. An efficient DevOps process supports and *enables continuous experimentation* which requires well set up experimental infrastructure and environments.

DevOps can impact occupational welfare, too. A senior developer noted that frequent releases help to reduce the stress levels because the anxiety related to handling huge releases is minimized. So, DevOps processes not only bring benefits to the organization, but they also improve the way of working, hence positively contributing to the *improved well-being of the DevOps teams*.

### 4.2 DevOps Adoption Challenges

Due to many factors, DevOps might not always be successful. The respondents raised challenges related to communication patterns, organization cultures which are not malleable, different constraints stemming from the domain and environments, and the obscurity of the meaning of DevOps.

A key impediment for successfully adopting DevOps is *insufficient communication*. For instance, it was mentioned that operations teams do not always monitor or pass all the performance and other metrics that can be of use to developers, which can cause problems. In sub-optimal cases, operations engineers and developers care about different, and possibly conflicting, metrics: operations personnel worry about the uptime of servers whereas developers are concerned with the release frequency. It was noted that communication between the two groups may also be lacking if it occurs only through electronic systems, causing delays in reaction times to issues. In person communication is hard to replace with electronic tools.

DevOps adoption also highlights cultural matters. Profound changes to the cultural mindset are required and the *deep-seated company culture* can be a challenge. As mentioned in the interviews, roles merge, responsibilities shift, and people have to rethink their established roles. Developers have to take on tasks they are not used to and might have reservations on accepting new responsibilities

for the operations environment like being on call for system failures. At the same time, operations people may be wary of the developers taking over their turf or overly taxed with handling more frequent releases. Changing people's behavior can be difficult, especially if they have had long careers. Smaller organizations might be in a better position to change their practices, though. As mentioned by one of the interviewees, people need to be receptive towards changing the company culture to fit the DevOps ideal since it is hard to push such initiatives through if, for instance, the management is not supportive. A Lead Architect stated the importance of culture and hardships in DevOps adoption.

"I think that's a big cultural shift that we are also seeing, difficult to address and what is probably the biggest blocker in moving ahead with this, bigger than the actual technical competences or processes. For me, I see it as two traditional roles that suddenly need to merge and we just need to find ways to, work towards one common goal." (Lead Architect, Org C)

DevOps practices might not be suitable in all circumstances. Access to production systems can be legally or contractually restricted so the *industry constraints and feasibility* in different domains need to be considered when applying DevOps. In specific cases, the environments such as databases used in production systems can be complex enough to make replicating the environments for verification and testing difficult as mentioned by a respondent. As a consequence, automated testing becomes less trustworthy meaning that *heterogeneous environments* provide a challenge for successful DevOps adoption.

While there is agreement about certain characteristics of DevOps, its true essence is still somewhat vague. Since there is not a standard set of fixed practices related to DevOps, practitioners find it hard to say what practices they should take into use for DevOps. It was seen that the meaning of DevOps has shifted in the previous years and new tools for DevOps keep coming up so one of the challenges is that *DevOps is unclear but also evolving.*

## 5 Discussion

The interviewees saw that adopting DevOps had some benefits. They saw DevOps as a means to increase the implemented features and generate more releases. The idea of rapid delivery aligns with the DevOps notion of reducing the time it takes for a software release to reach the production environment [3]. DevOps encourages automation, which was seen to help in improving the quality of releases.

DevOps helped to bridge the communication gap between developers and operations engineers. This fosters collaboration towards improving the development process and end product. Furthermore, the existing different skills can be readily utilized hence increasing the team's reactivity to problems.

The respondents thought that DevOps practices support real-time monitoring which helps to foster fast feedback loops and an experimental culture that engages more interaction with the end users. Real-time monitoring has also been previously highlighted by developers as a factor which helps to create fault-aware systems [2, 6] and DevOps has been seen to encourage experimental culture [11].

Shortcomings in communication and the prevailing company culture were some of the challenges that we identified. The lack of knowledge and information sharing can result in obscuring vital facts. Guidelines suggesting how to share information can help but changing the culture of a company can be a challenge. The cultural aspects are significant, as has been previously stated [7, 13]. The size of the company or having company-wide support for the change might matter. Smaller companies are in a better position to react faster to changes.

Constraints in the environment can prove challenging, too. Working habits emphasizing e.g. security can prevent some companies from using DevOps practices. Technical environments that are difficult to replicate add up to the challenges. Both challenges in the environment have been highlighted earlier [13].

As DevOps evolves, its definition, practices and tools are expected to change – a challenge that is expected to remain in the long run. As organizations continue to adjust to changes, Bass et al. [3] advise that DevOps should not be tied to any specific must-have tools or communication practices, but it should be aligned to the higher-level goals an organization wishes to achieve.

As an exploratory case study, there are no strong claims for causality and threats to internal validity are not central. The interviewees' responses are their own opinions on DevOps. Hence, the reported benefits and challenges are not universal. External validity could be threatened by the selection of the cases as two of the companies were consultancies. Consultants may see the situation differently because they can draw experiences from multiple clients. Still, considering external validity, it is possible that a company adopting DevOps might observe similar benefits and challenges as presented in the study.

The most notable threats to validity are related to construct validity and the operationalization of the concept of DevOps. Lacking a clear definition, respondents might have understood DevOps differently. Factors contributing to the benefits and challenges could have been left out if a respondent had a narrow understanding of DevOps which could affect the interpretation of its implications.

## 6 Conclusions

The study indicates some benefits and challenges involved in adopting DevOps. The benefits include more frequent releases, improved test automation, better communication and enhanced occupational welfare. DevOps can also support an experimentation culture in software development.

The factors inhibiting DevOps adoption were along human aspects e.g. lack of communication and resistance to change; and technical aspects e.g. the complexity of development and production environments. Suitability of DevOps might be questioned for certain domains and industry sectors, at least for now. The fuzzy definition of DevOps also prevents companies from having clear targets.

It would be interesting to learn about further implications of DevOps. Frequent releases are an advantage, but what are the effects of short release cycles and other DevOps practices? How does DevOps affect the end users, or is it just an

internal matter for the development organizations? Understanding the effects in a larger scale could help in assessing the real value of DevOps.

## Acknowledgments

This article was supported by TEKES as part of the N4S Program of DIMECC (Digital, Internet, Materials & Engineering Co-Creation).

## References

1. Digile N4S (2016), http://www.n4s.fi/en, retrieved: September 2016
2. Balalaie, A., Heydarnoori, A., Jamshidi, P.: Microservices architecture enables DevOps: Migration to a cloud-native architecture. IEEE Software 33(3), 42–52 (May 2016)
3. Bass, L., Weber, I., Zhu, L.: DevOps: A Software Architect's Perspective. Addison-Wesley (2015)
4. Callanan, M., Spillane, A.: DevOps: Making it easy to do the right thing. IEEE Software 33(3), 53–59 (May 2016)
5. Cruzes, D., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: International Symposium on Empirical Software Engineering and Measurement (ESEM). pp. 275–284 (September 2011)
6. Cukier, D.: DevOps patterns to scale web applications using cloud services. In: Proc. of the 2013 Companion Publication for Conference on Systems, Programming, & Applications: Software for Humanity. pp. 143–152. SPLASH '13, ACM, New York, NY, USA (2013)
7. Ebert, C., Gallardo, G., Hernantes, J., Serrano, N.: Devops. IEEE Software 33(3), 94–100 (May 2016)
8. Humble, J., Farley, D.: Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley Professional (2010)
9. Iden, J., Tessem, B., Päivärinta, T.: Problems in the interplay of development and IT operations in system development projects: A Delphi study of Norwegian IT experts. Information and Software Technology 53(4), 394–406 (2011)
10. Lindgren, E., Münch, J.: Agile Processes, in Software Engineering, and Extreme Programming: 16th Int. Conf., XP 2015, Helsinki, Finland, May 25-29, Proceedings, chap. Software Development as an Experiment System: A Qualitative Survey on the State of the Practice, pp. 117–128. Springer International Publishing (2015)
11. Neely, S., Stolt, S.: Continuous delivery? Easy! Just change everything (well, maybe it is not that easy). In: Proceedings of the 2013 Agile Conference. pp. 121–128. AGILE '13, IEEE Computer Society, Washington, DC, USA (2013)
12. Roche, J.: Adopting DevOps practices in quality assurance. Communications of the ACM 56(11), 38–43 (Nov 2013)
13. Smeds, J., Nybom, K., Porres, I.: Agile Processes, in Software Engineering, and Extreme Programming: 16th Int. Conf., XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings, chap. DevOps: A Definition and Perceived Adoption Impediments, pp. 166–177. Springer International Publishing (2015)
14. Tessem, B., Iden, J.: Cooperation between developers and operations in software engineering projects. In: Proceedings of the 2008 Int. Workshop on Cooperative and Human Aspects of Software Engineering. pp. 105–108. CHASE '08, ACM, New York, NY, USA (2008)