Integrating DevOps within IT Organizations– Key Pattern of a Case Study

Anna Wiedemann[1], Manuel Wiesche[2], Heiko Gewald[3] and Helmut Krcmar[4]

Abstract: Team-oriented, service-centric approaches for software delivery processes are becoming more and more popular. One approach that has appeared on the scene in the last decade is called DevOps. To date, little is known about how cross-functional product-oriented teams can be integrated within traditional companies. Hence, we decided to contact organizations that have already started implementing DevOps-specific processes. We talked to 34 people from 10 organizations and derived new insights into the area of DevOps. We show that there are different patterns by which DevOps can be integrated within companies—e.g. with the help of platform-oriented teams. Further, we find that teams are organized in different setups and use extended workbenches for collaboration. Our case study highlights that for successful DevOps implementation, it is important to convince every stakeholder and to work with agile coaches who can foster awareness for the necessity of DevOps.

Keywords: DevOps, Key Pattern, Characteristics, Case Study

1 Introduction

Digital transformation has led to new challenges for IT functions. A lot of companies are searching for new insights to manage their IT departments for meeting new customer requirements, because their working mode is rather reactive. Many incumbent companies are at an early stage of digital transformation and are under pressure to adapt their capabilities to develop digital strategies [PWK18; SR17].

IT departments have to work with new technologies, foster their business understanding, and focus on the delivery of new software features to customers [PWKa18]. Therefore, in recent times, service-centric IT team approaches are appearing and companies have started to implement the so-called DevOps IT teams. DevOps is a portmanteau of the words development and operations [LK16]. Literature highlights that there is no common definition of DevOps [FS17]. We define DevOps as a concept for integrating the tasks, knowledge, and skills pertaining to the planning, building, and running of activities in a single cross-functional team that is responsible for the combined development and operational tasks of one or more software service products. To quickly deliver new software features and innovations, ensure the quick handling of problems, and integrate maintenance activities, IT departments should integrate cross-functional teams rather than silo-organized IT units

[1] Neu-Ulm University of Applied Sciences, Center for Research on Service Sciences, Wileystr. 1, 89231 Neu-Ulm, anna.wiedemann@hs-neu-ulm.de

[2] Technical University of Munich, Chair for Information Systems, Boltzmannstr. 3, 85748 Garching, wiesche@in.tum.de

[3] Neu-Ulm University of Applied Sciences, Center for Research on Service Sciences, Wileystr. 1, 89231 Neu-Ulm, heiko.gewald@hs-neu-ulm.de

[4] Technical University of Munich, Chair for Information Systems, Boltzmannstr. 3, 85748 Garching, krcmar@in.tum.de

[FS17]. Aligning development and operational tasks within one team can lead to higher IT and organizational performance. High-performing DevOps IT functions are able to deploy 46 times more frequently than low-performing ones [BF16]. DevOps is a rising trend and publications predict that it will become a competitive necessity. More and more IT organizations are deciding to implement the DevOps concept [FK18]. According to a Gartner survey from 2016, about 25% of 2,000 global IT companies will work with the DevOps concept and tools in the DevOps toolchain in the future [RM15]. Hence, DevOps is being recognized as an important topic in the area of software development processes and process models. DevOps is a rising trend [WW18]. Hence, we have decided to conduct a case study with companies that are already working with DevOps principles. Taking into account the fact that the companies implement DevOps in different ways, we investigate the following research question: What are the key patterns of DevOps teams?

2 The DevOps Concept

Patrick Debois is considered as the person who branded DevOps. He held the first DevOpsDays conference in Ghent, Belgium in October 2009 [Re14]. As mentioned above, DevOps combines two words—development and operations. Key activities of DevOps are automated development, deployment, and monitoring of infrastructure within one cross-functional team. Organizations start using service-centric teams, integrating different roles and responsibilities, and breaking down historically grown silo departments with a high level of specialist knowledge. With the DevOps movement, a cultural shift toward better collaboration between developers, operations people, and quality assurance is necessary. The DevOps concept is helpful for delivering faster value to customers through timely provision of new software components, reducing problems through miscommunication, and enhancement of problem resolutions [EG16].

Prominent examples like Amazon and Google have already adopted principles of the DevOps concept and now have cycle times of new software components in seconds [SR17]. The DevOps approach enables the scaling of agility to the entire IT organization. The goal of DevOps is to enhance collaboration, automation, virtualization, and tools to bridge the activities of software development and operation [LK16]. Through DevOps, solutions are delivered to avoid interruptions between different stages of the software delivery process [FS14]. The software development lifecycle consists of planning, development, and operation tasks. DevOps helps companies to integrate the necessary speed and flexibility to deliver constant and rapid development and implementation of digital innovation. Hence, risks linked with software releases can be reduced, and feedback of new software features is received faster. In addition, agile methods can be used to manage the software development part of the team [LK16]. In agile development environments, there are different "continuous" activities, the best-known of which is continuous integration (CI). CI is defined as a process that is provoked automatically and encompasses interconnected stages like acceptance test, code validation, compliance checks, and release package development [FS14]. CI disables interruptions between the development and deployment stages of software delivery. It is very significant for the DevOps phenomenon because good collaboration between development and operation is needed [FS14]. The benefits of CI are improvements in communication, higher frequency in releases, and an enhancement in the productivity of developers [FS14; SB14]. CI can lead to other continuous activities like continuous deployment (CD) and delivery [FS17]. DevOps could help to enable these processes.

DevOps complements and extends CI and releases agile software development processes by bridging new software features quickly into production and value delivery to the customer. DevOps teams can use agile methods to manage collaboration and work within their deployment environments. Companies use DevOps for better collaboration and monitoring in order to enhance the continuous delivery of new software features and consequently foster innovation [LK16].

3 An Investigation of DevOps Teams

As mentioned above, there are some prominent examples of companies already working with DevOps approaches. However, little is known about how the DevOps concept is adopted in existing companies. Hence, we decided to contact organizations that have already adapted the DevOps approach. Therefore, we searched through the internet and social business networks (e.g. Xing), and contacted people in various companies responsible for DevOps. Thus, we were able to find 10 participants for our case study. We held two or more interviews per case. After conducting the interviews, we identified different characteristics that were adopted by each case, which we grouped to four different categories:

- **Organization of IT function:** This describes how the IT function is organized and how the DevOps approach implementation begins.
- **Implementation core and product:** This describes the DevOps team setting and service.
- **CI/CD:** This describes the degree of CI/CD pipeline implementation and the deployment rates.
- **Extended workbench:** This describes how the DevOps team is organized (i.e. with greater development or operations background) and the degree of work task organization within the DevOps team.

We present the characteristics of the DevOps setup in the similar cases in Table 1.

4 Data Collection

To answer our research question, an exploratory multiple-case-study approach is adopted for a number of reasons. The case study approach is defined as “an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context” [Yi09, p. 18]. DevOps is quite unexplored and case studies are helpful to examine new phenomena from various perspectives [Yi09]. The advantage of case studies is that they can zoom in on real-life situations and test or develop theoretical views in relation to phenomena, as they unfold in practice [Fl06].

Case	Organization of IT function	Implementation of core and product
Case 1 (Media)	Newly founded company with an IT function with high DevOps orientation.	The team is responsible for a data service for website delivery (internal service).
Case 2 (Service)	Transformation toward DevOps for several years.	The team is responsible for a product of their website (e.g. insurance service).
Case 3 (Home)	Ad hoc decisions to reorganize the IT function with DevOps teams.	The team is responsible for an activity of the online shop (e.g. shopping basket).
Case 4 (Pet Industry)	Strategic decision to reorganize the IT functions with the help of DevOps teams.	The team is responsible for an activity of the online shop (e.g. product rating).
Case 5 (Specialized Store)	Strategic decision to integrate DevOps teams for managing their online shop.	The team is responsible for an activity of the online shop (e.g. checkout).
Case 6 (Retail 1)	New company with the idea to integrate high DevOps orientation.	The team is responsible for an app development and support (e.g. delivery service).
Case 7 (Insurance)	Integration of the DevOps approach within an existing IT function.	The team is responsible for an internal delivery platform and offers self-service to other teams (internal service).
Case 8 (Bank)	Integration of the DevOps approach within an existing IT function.	The team is responsible for running and facilitating a securities management system (internal service).
Case 9 (Retail-non-food)	New company with the idea to integrate high DevOps orientation.	The team is responsible for building and running an internal platform, which is the substructure of the online shop of the company (internal service).
Case 10 (Retail 2)	Integration of the DevOps approach within an existing IT function.	The team is responsible for building and running an internal platform, which is the substructure of the online shop of the company (internal service).

Tab. 1: Findings of the Case Study Research (1/2)

Case	CI/CD	Extended workbench
Case 1 (Media)	CD is integrated. The team is able to deploy around every four days.	The team consist of 12 software engineers located in Germany and with a nearshoring partner in East Europe and shares development and operations work.
Case 2 (Service)	Frequent releases but no optimal CI/CD.	The team consists of eight software developers and is located in Germany. The team lead is responsible for the operations parts.
Case 3 (Home)	CD is integrated. The team is able to deploy several times a day.	The team consists of five software engineers. It is located in Germany and shares development and operations work.
Case 4 (Pet Industry)	No optimal CI/CD. The team is able to deploy every two weeks.	The team consist of four developers and is supported by a quality assurance engineer. The team is located in Germany and shares development and operations work.
Case 5 (Specialized Store)	No optimal CI/CD. The team is able to deploy every two weeks.	The team consists of five software developers and operations people. It is located in Germany and shares development and operations work.
Case 6 (Retail 1)	CD is integrated. The team is able to deploy every two weeks.	The team consists of four software developers and is located in Germany. The infrastructure team is a separated unit and is responsible for operations during the day.
Case 7 (Insurance)	CD is integrated. The team is able to deploy several times a day.	The team consists of 15 software engineers allocated in Germany and Asia and shares development and operations work.
Case 8 (Bank)	No optimal CI/CD. The team is able to deploy once a week.	The team consists of 15 software operations people and is located in Germany. The developers are a separate unit.
Case 9 (Retail-non-food)	CD is integrated. The team is able to deploy several times a day.	The team consists of six software operations people and is located in Germany. The developers are in a separated unit.
Case 10 (Retail 2)	No optimal CI/CD. The team is able to deploy every two weeks.	The team consist of three software developers and is located in Germany and with a nearshoring partner in Bulgaria. The infrastructure is managed by another subsidiary company of the group.

Tab. 1: Findings of the Case Study Research (2/2)

5 Data Analysis

We applied a multiple-case-study design that enables cross-case pattern search. This method helps to examine processes and patterns over several cases to understand how similar or contrasting the results are [MH94; Yi09]. The data were analyzed with the help of "within-case analysis" as well as "cross-case-analysis". In the within-case analysis, each team was seen as a stand-alone entity and analyzed [Yi09]. The analysis process was conducted through the lens of the key pattern. The focus was on cross-case analysis, to compare the cases and identify the patterns obtained for each case. In the present research, the coding approach emphasized by Miles and Huberman (1994) has been applied. Afterwards, in vivo coding was applied to examine each case and emergent topics or explanations. Additionally, it helps to achieve more familiarity with each case and fosters cross-case analysis [MH94; Yi09].

6 Key Findings and Discussion of the Case Study

In the following section, we present several insights achieved through our case study, as well as the key pattern identified in our research.

6.1 Insights from the DevOps Teams

To summarize, we talked with 34 persons from the aforementioned companies. All of them offer different insights into the DevOps setting. During our investigation, we recognized that there are different patterns across the several cases. Nevertheless, we found some similarities as well. Table 2 presents an illustrative example of the key findings that we identified in the different IT functions. The table presents two patterns generated with the help of our case study.

Cases 5, 6, 7, 9, and 10 have integrated a so-called platform team, which they have organized with the help of the DevOps approach. That means that an internal platform is used as the basic infrastructure—as the foundation for the online shop of the organizations. The infrastructure teams are often organized with the help of operations people who have high interest in development tasks as well. For example, they develop their own software program and write playbooks, mostly using Linux as the operating system. Hence, for them, it is a key requirement that new team members should have knowledge in Linux.

DevOps platform teams usually have operations background and undertake on-call duty outside normal business hours or developers with experience in system administration. The development teams are responsible for their services during the normal business hours. For a couple of reasons such as costs, companies may decide not to integrate on-call duty in the development teams. Very often, a lot of deployments are made during the day.

Companies start implementing some rules and regulations for deployments. For example, one case mentions that new software components are not allowed to be deployed into the production environment on Fridays, because they want that the system to be running stably during the weekend. Another case mentions that if a developer deploys new software components after the regular deployment times, the person has to be available for the DevOps platform team in the case of failure messages. The overall aim of the platform DevOps teams is to integrate self-services for the development teams. This means that developers are able to help themselves with the particular services they need (e.g. databases or firewalls). The DevOps platform team serves these services to the development teams with the help of application program interfaces (APIs).

Pattern 1	**Pattern 2**
DevOps teams have an internal platform orientation with self-service opportunity for development teams.	The IT department mainly consists of DevOps teams that are working with e.g. micro-service architecture.
Most team members have operations background and have started working with development tasks.	Most team members have software development tasks and have to learn operations tasks.
The team is mainly responsible for providing infrastructure to the development teams and supporting them.	The team is mainly responsible for service and infrastructure. If superordinate problems appear, they are supported by a support unit.

Tab. 1: Pattern of DevOps Teams

The other cases that we investigate present insights into another DevOps setup. Cases 1, 3, 4, and (partly) 2 claim that they have integrated the development and operations tasks within one team. The CTO of Case 3 mentions that it is important for the development team to be fully responsible for its service. Most of the cases are responsible for one service of an online shop (e.g. shopping basket, check-out processes). If problems appear, they have to manage it as well. The DevOps team has a high decision-making freedom regarding new technology. Furthermore, it has to guarantee that skills and knowledge necessary for managing the service are integrated within the team. For example, Case 2 mentions that it is not possible for everyone to know everything. The team should be organized so as to ensure that the service is working, even if team members are on holiday or unwell. Hence, team members should be able to stand in for each other.

IT organizations and companies organize meetings, tech-talks, and other presentations to share knowledge within and outside the company. Sharing knowledge is a key factor in the DevOps movement [BC13]. These knowledge-sharing activities are helpful because the teams learn from each other. Additionally, the teams are supported by agile coaches or disciplinarians. These people manage the teams and help them with decisions —for example, deciding whether new technologies are really necessary if the technology already exists within the company with the help of another tool. Figure 1 visualizes two examples of the different DevOps setups observed in several cases.

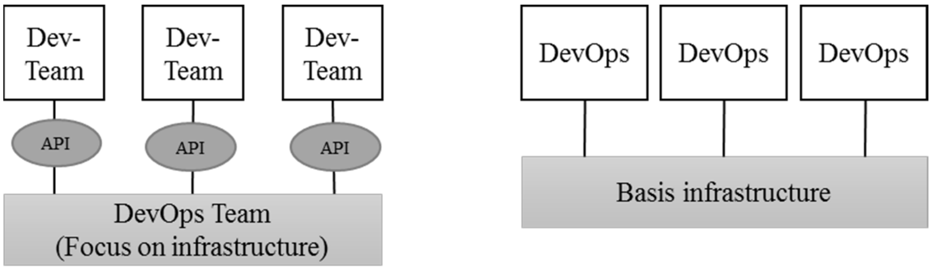

Fig. 1: Organizational Structure of DevOps

Furthermore, our findings show that the integration of DevOps presents challenges for the existing IT function and organizational structure. We recognize that the IT functions—whether completely reorganized with DevOps teams or organized as bimodal IT functions—are traditional silo units and DevOps teams coexisting.

6.2 Starting Points for DevOps

Another finding of our case study is that a lot of tradition-oriented companies are searching for ways to implement DevOps within their company. The study participants mention different starting point for the implementation of DevOps within new or existing IT functions. We summarize the following four different starting points for DevOps implementations:

- Spin-offs/subsidiary companies
- Greenfield projects
- Slow accession process
- Ad-hoc adjustment

Our findings show that different starting points for a successful implementation are possible. These opportunities should be considered by organizations trying to implemented DevOps.

6.3 Movement Towards Product Orientation

A major finding of our study regarding the organization of DevOps teams is that they do not usually work in a project setup. They are organized in the so-called product setups. IS projects usually have pre-defined project aims such as project delivery on-time and within-budget [Ki04]. Our findings suggest that there is a shift toward product-orientation—this means that there are no regular starting or ending dates for the product team. Furthermore, budget and on-time milestones were not controlled by the team in our investigation. A redesign of processes and end-to-end service responsibility is important for cross-functional DevOps teams [BRC11].

7 Conclusion for DevOps Implementation

Our findings present insights into the DevOps implementation. We depict different characteristics and types of DevOps patterns across our case-study participants. For the implementation of the DevOps concept, it is important to convince every stakeholder. We present an overview of two key patterns that we identified with the help of our case study. The first pattern consists of operations team members with a highly internal platform orientation with the aim to integrate self-services for development teams. The second pattern presents evidence that DevOps teams have high general knowledge among team members with developer backgrounds. The teams are responsible for one or more product-oriented services.

The stakeholders should be integrated into strategic decision-making processes. The integration of DevOps teams is accompanied by several challenges. The DevOps teams need more freedom for decision-making processes and higher self-responsibility for their service. Thus, traditional hierarchies should be broken down. Organizations should only give a minimum number of rules and regulation, e.g. for technology choices. Traditional silo-organized IT departments with highly specialized knowledge need to be reorganized. Cross-functional (service-centric) teams with general knowledge about the service should be integrated within the IT department.

8 Acknowledgment

This research has received funding from the German Federal Ministry of Education and Research (BMBF) under grant code 03FH005PX4. The responsibility for the content lies with the authors.

References

[BRC11] Balaji, S., Ranganathan, C., Coleman, T.: IT-Led Process Reengineering: How Sloan Value Redesigned Its New Product Development Process. MIS Quarterly Executive 10/2, S. 81–92, 2011.

[BC13] Bang, S.K., Chung, S., Choh, Y., Dupuis, M.: A Grounded Theory Analysis of Modern Web Applications: Knowledge, Skills, and Abilities for DevOps. In: Proceedings of the 2nd Annual Conference on Research in Information Technology. Orlando, S. 61–62, 2013.

[BF16] Brown, A., Forsgren, N., Humble, J., Kersten, N., Kim, G.: State of DevOps Report. Puppet+DORA, 2016.

[EG16] Ebert, C., Gallardo, G., Hernantes, J., Serrano, N.: DevOps. IEEE Software 33/3, S. 94–100, 2016.

[FS14] Fitzgerald, B., Stol, K.-J.: Continuous Software Engineering and Beyond: Trends and Challenges. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering. Hyderabad, 2014.

[FS17] Fitzgerald, B., Stol, K.-J.: Continuous Software Engineering: A Roadmap and Agenda. Journal of Systems and Software 123, S. 176–189, 2017.

[Fl06] Flyvbjerg, B.: Five Misunderstandings About Case-Study Research. Qualitative Inquiry 12/2, S. 219–245, 2006.

[FK18] Forsgren, N., Kersten, M.: DevOps Metrics. Communications of the ACM 61/4, S. 44–48, 2018.

[Ki04] Kirsch, L.J. Deploying Common Systems Globally: The Dynamics of Control. Information Systems Research 15/4, S. 374–395, 2004.

[LK16] Lwakatare, L., Karvonen, T., Sauvola, T., Kuvaja, P., Holmström Olsson, H., Bosch, J., Oivo. M.: Towards DevOps in the Embedded Systems Domain: Why Is It So Hard?. In: Proceedings of the Hawaii International Conference on System Sciences, Kauai, 2016.

[MH94] Miles, M. B., Huberman, A. M.: Qualitative Data Analysis: An Expanded Sourcebook. Thousand Oaks, Sage Publications (Hrsg.), 1994.

[PWK18] Pflügler, C., Wiesche, M., Krcmar, H.: Subgroups in Agile and Traditional IT Project Teams, In: Proceedings of the Hawaii International Conference on System Sciences, Waikoloa Village, 2018.

[PWKa18] Przybilla, L., Wiesche, M., Krcmar, H.: The Influence of Agile Practices on Performance in Software Engineering Teams: A Subgroup Perspective. In: Proceedings of the ACM SIGMIS-CPR, Buffalo, 2018.

[Re14] Reed, J.P.: DevOps in Practice. Sebastopol, O'Reilly Media, Inc. (Hrsg.), 2014.

[RM15] Rivera, J., van der Meulen, R.: Gartner Says by 2016, DevOps Will Evolve from a Niche to a Mainstream Strategy Employed by 25 Percent of Global 2000 Organizations. Stamford, Gartner (Hrsg.), 2015.

[SR17] Sebastian, I.M., Ross, J.W., Beath, C., Mocker, M., Moloney, K.G., Fonstad, N.O.: How Big Old Companies Navigate Digital Transformation. MISQ Executive 16/3, S. 197–213, 2017.

[SB14] Ståhl, D., Bosch, J.: Modeling Continuous Integration Practice Differences in Industry Software Development. Journal of Systems and Software 87, S. 48–59. 2014.

[WW18] Wiedemann, A., Wiesche, M..: Are You Ready for DevOps? Required Skill Set for DevOps Teams. In: Proceedings of the European Conference on Information Systems. Portsmouth, 2018.

[Yi09] Yin, R.K.: Case Study Research: Design and Methods. Thousand Oaks, Sage Publications Inc. (Hrsg.), 2009.