DevOps Adoption: Challenges & Barriers

Mike Krey, Ahmad Kabbout, Lavdrim Osmani, Armend Saliji
School of Management and Law
Zurich University of Applied Sciences
Winterthur, Switzerland
Mike.Krey@zhaw.ch, Kabboahm@students.zhaw.ch, Osmanlav@students.zhaw.ch, Salijarm@students.zhaw.ch

Abstract

As a modern software engineering paradigm, DevOps has recently gained increasing acceptance in the industry as a set of practices and cultural values to address daily dynamic software demands. While the rising trend of DevOps and its characteristics and challenges have often been characterized by practitioner communities and academic research circles, there is still a lack of a thorough understanding of how to tackle DevOps adoptions. This paper aims to help fill this gap by identifying, discussing, and summarizing current academic and practitioner DevOps adoption & implementation research. Our findings provide a basis for theoretical, empirical, or design-oriented research for IS scholars, that has the potential to be of practical importance. Our goal is to improve understanding of DevOps adoption by uncovering ambiguities in terms, conceptual conflations, and ideas underlying different uses of the concept as well as providing methods to deal with common challenges in the adoption process.

1. Introduction

In recent years, companies have shifted their software development to an agile and lean approach, wherein software can be released early and with higher frequency [11]. Companies wish to increase their software quality by using this approach. The introduction of agile methods in software development have improved the performance of development teams. This has been achieved by introducing cross-functional teams and providing closer collaboration with customers [23].

DevOps has been introduced as an approach to reach and expand these goals and to bring the software development and operations team together [11]. It is an organizational approach, which aims to create empathy and cross-functional collaboration. Overall, the goal is to reduce the time between development and operations of software without having a negative effect on quality. However, there is no census on which concepts are covered by DevOps, nor how DevOps is defined. Chapter 3 expands on this point and seeks to provide a comprehensive definition of DevOps.

DevOps has become an important phenomenon in software engineering for the private sector as well the scientific community [20]. This paradigm shift towards continuous deployment of software has brought opportunities as well as challenges [23]. Especially the adoption of DevOps has proven to be challenging [30]. It requires the organization to introduce processes, specially trained personnel, technological changes, and innovations. Most importantly, the adoption process is unique to the company. This task has proven to generate challenges when adopting DevOps.

In general, the challenges can be perceived through three perspectives: Engineers can benefit from both, qualifying for a DevOps position and possessing a vast amount of knowledge. However, they need to know how to redesign their systems to **incorporate continuous delivery (1)**. Managers desire to know how to introduce DevOps into the organization and how to assess the **performance and quality of the adoption (2)**. Both engineers and managers (also known as practitioners) share the need of choosing the adequate automation toolset. Lastly, researchers face the challenge of determining the state of DevOps in practice and educating the new generation of software engineers on **DevOps principles and practices (3**) [20].

To facilitate the introduction of DevOps, various frameworks, methods, and checklists have been proposed by different practitioners to successfully introduce DevOps and minimize risks [30]. Although large companies like Netflix and Flickr have adopted these approaches and achieved success, they are still controversial in terms of generalizability and feasibility. For example, the implementation of DevOps in SMEs (small and medium-sized enterprises) has not been researched comprehensively in the literature. Various scholars point to this scarcity on the topic, noting that implementation in practice creates a level of uncertainty [24].

In this paper, we seek to tackle the knowledge gap in the adaption and implementation process in companies. First, we lay out the theoretical framework and the state of the art in DevOps implementation.

URI: https://hdl.handle.net/10125/80219
978-0-9981331-5-7

Therein, we show challenges and impediments in the process. Furthermore, we explore different frameworks used in practice and aim to derive methods and approaches to overcoming these challenges and barriers.

Based on the challenges discussed above, the following research questions have been derived:

- **RQ1:** What challenges/barriers exist in SMEs when introducing DevOps?
- **RQ2:** How can companies overcome these challenges?

In accordance with the research questions posed, we aim to achieve the following goals for this paper:

1. Map the status quo of the connection between agile software development and IT operations and derive practice-oriented approaches for the implementation of DevOps.
2. Derive common attributes that relate to commonly used DevOps implementation practices and frameworks.
3. Deriving actionable organizational measures to mitigate resistance to embedding the DevOps mindset.

2. Methodology

This paper follows the tradition of business informatics, which sees itself as an applied science and draws on instruments from real, formal, and engineering science perspectives. On this basis, theory-practice integrating research approaches are used, with the aim of presenting practice-relevant results and recommendations, but also creating transparency in order to evaluate the scientific contribution of the paper.

In the upcoming section a literature review provides the most recent state of DevOps research, leading to a concluding discussion on challenges and barriers to DevOps for organizations. A literature review was conducted investigating contributions ranging from 2014 until 2021. In order to classify existing approaches and identify their relevance for the identified problem, the taxonomy framework by Cooper [9] is applied. This taxonomy addresses six characteristics: (1) focus, (2) goal, (3) perspective, (4) coverage, (5) organization, and (6) audience (p. 109). The following configuration of characteristics is applied for the literature review (see Table 1).

Table 1. Configuration of the literature review based on Cooper (p. 109) [9]

Characteristic	Configuration	Description
(1) Focus	Research outcome	The review focusses on the findings of present contributions 2014 until 2021
(2) Goal	Identification of central issues	The primary goal of the literature review is to explore the field of DevOps thoroughly and identify central issues.
(3) Perspective	Neutral	Given the focus of the review, the perspective is neutral when presenting findings as facts.
(4) Coverage	Exhaustive review with selective citation	A manageable number of 608 articles are reviewed focussing on contributions in English
(5) Organisation	Historical	The review is organized chronologically, emphasising the progression of contributions in the field over time.
(6) Audience	Specialised Scholars	The audience of this work are in the intersection of computer science and business, coming from both, science and practice.

Following the approach by Cooper [9], who proposed a stage model for conducting literature reviews, the following databases were searched, using "DevOps", "Development" AND "Operations", "DevOps Adoption" as search terms in titles and abstracts:

(1) ACM Digital Library, (2) IEEE Xplore Digital Library to explore sources with a technology perspective such as the development and implementation of DevOps systems, (3) AIS Electronic Library (AISeL) to locate core DevOps topics such as the use of DevOps in organizations or in management.

In addition to these management and technical related databases, (4) the SprinerLink database is queried, which covers recent journals, books, and book series to the analysis.

The ACM Digital Library yielded 42 and the IEEE Xplore 175 results. Although relatively new, 197

DevOps contributions were found by AISeL. Finally, the SpingerLink database yielded 194 results. This makes a total number of 608 contributions that were analyzed. After excluding inappropriate entries, as well as double counts (i.e. the same article listed in several databases, total of twenty-five contributions are analyzed in more detail. This includes the reading of each paper's abstract, introduction, and conclusion section. Of these studies, twenty-three were attributed to academic research. Unfortunately, no contribution was found dealing with DevOps in conjunction with SME. Tables 2- 4 provide an overview of the analyzed articles summarizing the objectives, placement of the DevOps aspects, and utilized research strategies according to Saunders, Lewis and Thornhill [28, p. 160).

The remainder of this paper is organized as follows: The first part provides the basic understanding, which includes definitions as well as challenges and already established approaches in practice (Section 3). This part is therefore essential, as the later sections build on it. Initial ideas and possible recommendations to practitioners may nevertheless emerge in this part, which is why they are also recorded.

In a second part, we present the results of our systematic literature review and show what has been researched so far in the DevOps adoption area. Based on this, we derive the aspects that have research potential. These are finally discussed and critically examined. A conclusion section with future work forms the final part of this paper.

3. DevOps

3.1. Definition

DevOps is often described as a software engineering culture and philosophy, which employs cross-functional teams in order to build test and release software faster and more reliably [24]. This is mainly achieved through automation. DevOps aims to improve collaboration between development and operation teams in software development.

In the scientific community, many views and stances on DevOps have been developed. Macarthy & Bass [24] show two conflicting views: DevOps is a cultural movement used to facilitate rapid software development (1) and the view that DevOps is rather a job description, which requires both development and IT operation skills (2).

Reflecting on these views of DevOps as well as its practical use, it becomes clear that automation only does not mean DevOps. Automation is an important part of DevOps, but there are additional aspects that need to be considered [31]: culture, processes, organization, automation, and continuous improvement.

Since much of the research done in DevOps relies on the cultural movement view, this will also be the view we represent in this paper. Furthermore, we believe that this view is essential for the perceived benefits of DevOps.

3.2. Driving factors of DevOps

Taking the development of application in the consideration, the importance of DevOps becomes clear. The high pressure on IT is mainly caused by three factors [31]: external environment (1), internal organization (2) and technical development (3).

The external environment (1) is adapting quickly and is changing the requirements for service deployments and applications, as well as the nature of collaboration within enterprises [31].

The internal organization (2) of the IT departments is also changing. An agile software development approach such as DevOps enables the organization to have quicker IT solutions. The implementation period can be shortened with an agile solution [18].

From an application maintenance point of view, the requirements for stability, governance and availability are becoming higher. Moreover, well established controls and processes need to be in place in order to be productive [31].

In addition to that, the technical development (3) needs to be considered [18]. Efficient open-source systems such as Jenkins, Ansible, Docker, Nagios, Git and/or OpenStack provide an option for the automation of DevOps. Simplified monitoring enables these tools to provide better evaluations of application behavior [31].

DevOps popularity is also driven by the benefits companies seek to obtain by adopting it. The key benefits are faster turnaround, shorter time to market, increased customer satisfaction, quality improvement, highly collaborative and motivated teams as well as improved productivity and efficiency [15].

3.3. The rise of DevOps

As explained previously, DevOps has become more popular in the last decade. This rise has its origin with some well-known companies such as Google, Apple, Amazon etc. that have been able to successfully implement and reap profits from it [3, 18].

Many companies of different industries are inspired by these promising developments and strive to achieve similar results by adopting such concepts and restructuring their IT based on common frameworks

(e.g. ITIL). However, due to the complexity of their architecture or their previous production cycles, they often encounter - as mentioned above - various hurdles that require the necessary attention, resources, and effort. These aspects are typically organizational, socio-technical, and technical in nature [7].

Through the work of Capizzi et al. [8], which is based on a longitudinal ethno-methodological approach, they have followed companies in their DevOps journey and identified eight different pitfalls that every company should consider before implementing DevOps. The results presented, however, are aimed at a fine-grained layer, either based on a specific company, or so vague that not every company can refer to it. Comparative results can be found in Bordeleau et al. [5], which have worked with practice partners and have surveyed their problems and tried to solve them.

These research papers deal mostly with individual challenges that organizations have. A deeper and more generalized view to answer the questions posed in a holistic yet comprehensible approach that organizations can use as a handbook and evaluate themselves is sparse in the literature.

Table 2 presents the state of the art in DevOps literature.

3.4. Implementation & adoption

Today, customers and users expect software applications to cater to their frequently changing needs [15]. In order to accommodate changing customer needs, it is imperative for companies to make frequents releases and deployments. This requires an environment that is disciplined; otherwise it may lead to a number of failures, customer dissatisfaction and losses. DevOps aims to fill this gap and the guiding principles can be summed up as follows: culture, measurement, collaboration, and automation [15].

Gupta et al. [15] identified four major (latent) variables, which influence the DevOps implementation process: source control (1), automation (2), cohesive teams (3) and continuous delivery (4).

In total eighteen attributes represent these factors. With the help of experts in the field, they were able to confirm that these factors are independent from each other. Based on these factors, 10 major sub-factors were identified, that influence implementation [15]. These are branching patterns, branching changes, feature toggle, automated code review, automated testing, branching depth, automated tools for monitoring and infrastructure as a Code. With the help of this framework, practitioners are able to conduct a detailed assessment of the maturity level of the DevOps implementation. This method uses statistical analysis [15].

Other popular frameworks include culture, automation, measurement, sharing and services as the main dimensions of DevOps [29]. As mentioned earlier some researchers classify DevOps as a cultural movement that enables rapid development [17, 22, 25].

The cultural movement has four defining characteristics: open communication, incentive and responsibility alignment, respect, and trust. However, some researchers argue that cultural aspects themselves cannot be the defining characteristics of DevOps but rather enablers to support software engineering process capabilities [29].

Another point of view some researchers hold is that DevOps can be characterized by several defining enablers. Three core aspects include: capability enablers, cultural enablers, and technological enablers [29]. These enablers can be seen as potential requirements to be met for implementing DevOps.

Capability enablers focus on basic activities in software development (i.e. planning, development, testing and deployment). These activities are carried out continuously and require that the support of technical practices such as test automation and effective collaboration between the teams is given [29]. The technological enablers are needed to support the DevOps capabilities by automating tasks. By automating tasks, continuous delivery and deployment is facilitated. Cultural enablers are related to behaviors that DevOps teams must exhibit to support DevOps capabilities. This requires extensive collaboration, low effort communication as well as shared goals and collective ownership [29].

3.5 Common attributes associated with DevOps

Lwakatare et al. [23] conducted a literature review on the DevOps issue and found these four elements to be key for DevOps implementation: collaboration (1) automation (2), measurement (3) and monitoring (4).

In the academic literature, there seems to be a consensus among practitioners, that DevOps at its core encompasses a culture of collaboration between software development and operations teams [23]. Testers and quality assurance teams are also important stakeholders. The collaboration aspect is enforced by sharing information, skillset broadening and shifting responsibilities between the two teams.

Moreover, a general sense of shared responsibility is instilled. A proper implementation of these approaches requires a change in the mind-set of the teams as well as changes in the culture of the organization [23]. Furthermore, collaboration has an impact

on the team structure and leads to changes in the required skillsets of software development and operations personnel.

Practitioners and scholars agree that automation in operations processes and testing is an essential part of DevOps and necessary in software development [23]. Operations processes need to be able to keep up with the pace of agile software development and continuous integration (CI). Therefore, these processes need to be repeatable, flexible, and fast [23]. This is done by eliminating manual processes. Especially in complex environments manually deploying functionality and managing configurations repeatedly and quickly is time consuming. Moreover, quality of deployed functionality is improved by test automation and CI [23]. A common approach used to address manual processes is known as Infrastructure as a Code (IaC). This concept describes the idea that almost all actions performed to the infrastructure can be automated [23].

Another important element according to Lawakatare et al. [23] is the ability to measure the development process. This is done by incorporating metrics, which help increase the efficiency. Furthermore, measurement should not only cover quality assurance efforts, but data should be used to seek insights about quality and usefulness of software functionality [23].

Lastly, monitoring tools are used to obtain information regarding a systems overall health [23]. However, locating the problem in largely voluminous logs is time-consuming. Furthermore, continuous deployment poses a challenge for monitoring activities since they require them to be effective and fast [23]. In DevOps, these challenges are addressed by putting am emphasis on collaboration such that the systems are design to expose relevant information. This information is provided to developers, product management and is used for product improvement and customization [23].

Table 3 gives an overview over commonly identified attributes, which are essential to a successful DevOps implementation.

3.6. Challenges & barriers in the implementation process

Many companies face challenges in the implementation process of DevOps. The six major challenges that small and medium-sized enterprises (SMEs) face today are [16]:

- **Costs**: should remain calculable.
- **Risks**: should remain low and assessable.
- **Scope**: should be flexible (benefit dependent)
- **Quality**: should be appropriate and measurable by criteria.
- **Business Value**: is the focus of all development activities as a business-customer- and stakeholder benefit.
- **Time:** it is important to be able to react adequately to market changes and to be able to quickly verify or falsify hypotheses about possible customer needs.

Due to a number of factors, DevOps implementation can prove to be unsuccessful. Insufficient communication is a key impediment for adopting DevOps successfully. Operations teams do not always pass or monitor all the performance and further metrics that can be useful to developers, which can also cause problems [25]. In suboptimal cases, developers and operations engineers care about possibly conflicting and different, metrics: developers worry about the release frequency whereas operations personnel are concerned with the uptime of servers [25].

Tools can create another problem of Dev and Ops teams, having completely different toolsets and metrics [14]. The communication between the two teams could be lacking if it occurs only through electronic systems, causing delays in reaction times to issues [25]. DevOps relies on variety of tools to construct a process. In-person communication is hard to replace with electronic tools. Furthermore, the integration of those tools can prove difficult and problematic to maintain and integrate at times. Thus, resistance to change occur [14]. The necessity of maintaining legacy systems is one of the biggest issues the developers face [17].

The adoption of DevOps also highlights cultural matters. A deep-rooted company can be a challenge since profound changes to the cultural mindset are required. There are a number of changes, responsibilities shift, people have to rethink their established roles and roles merge [25]. The move to DevOps can seem scary to many key stakeholders and team members [14].

Developers may have to take on new responsibilities for the operating environment, such as being on-call for system failures, and may have to take on responsibilities they are not used to. At the same time, operations staff may be wary because they are overwhelmed with handling more frequent releases, or developers may take away their domain. However, there is no consensus on how to effectively empower collaboration among departments, and there are only a few tools for tackling this issue [20].

Access to production system can be contractually or legally restricted so the company's feasibility and constraints in different domains need to be considered when applying DevOps. In some cases, the environments like data systems or databases used in production systems can be complicated enough to make replicating the environments for testing and verification

complex. As a result, automated testing becomes less trustworthy meaning that heterogeneous environments provide a barrier for successful DevOps adoption [25].

An overview of common challenges can be seen in Table 3.

3.7. Approaches to overcoming challenges

One solution for clear communication within a company after the introduction of DevOps is job crafting. Job crafting is the design of physical or cognitive task boundaries and/or relational boundaries. It implies that duties are at the core of the employee-employer relationship. Three specific types of job crafting may be essential in DevOps adoption [17]: task, relational, cognitive.

Task crafting refers to job crafting where work is completed in a timely manner and where the meaning of the work may change, making the employee the guardian or mover of a project. Relationship job crafting occurs when employees see their work as an important part of an integrated whole. In cognitive job crafting, employees change their perception of the work, and the focus is not exclusively on the quality of the outcome. Job crafting is a phenomenon that is often studied using a longitudinal approach [17].

A difficult question that has not been fully answered in the literature is how - or perhaps even if - an organization should be restructured to adopt DevOps. Leite et al. [19] suggest three different solutions that could help circumvent this problem: ***(1) Collaborating departments:*** Development and operations departments work closely together, with overlapping responsibilities between developers and operators. The use of collaborating departments is used when automation is not prioritized in operations [20]. ***(2) DevOps teams:*** Assigning a "DevOps team" as a bridge between developers and operators has become a trend. A DevOps team can be introduced if the project is time-limited and focused on passing development practices to operations staff and to developers [20]. Introducing a DevOps team has proven to be better accepted as a temporary cultural transformation strategy. *(3)* ***Cross-functional teams:*** Product team is responsible for deployment and operations. Cross-functional teams seem to be gaining traction in the literature. In this model, at least one team member must have operational skills. From this perspective, the so-called "DevOps Engineer" is also called "Full Stack Engineer". This method is also recommended by Amazon and Facebook [20].

Table 2. State of the art & research studies in DevOps

Author	Objective	Business case (B) and / or Research Imperative (R)	Focus	Approach and perspective	Empirical domain / participants	Key findings
Lwakatare et al. (2015) [23]	Investigate the elements that characterize the DevOps phenomenon	B: Lack of common understanding of what DevOps constitutes in academia and in the practitioners' communities R: Need for research that investigates the DevOps phenomenon and examines how it impacts software development and operations	Dimensions of DevOps	Literature review, survey & interviews	4 practitioners working actively involved in the DevOps movement	▪ Definition of the main elements that characterize the DevOps phenomenon: collaboration, automation, measurement, and monitoring ▪ An initial conceptual framework that describes the phenomenon ▪ Still a need for empirical research to validate and enhance the presented conceptual framework
De França et al. (2016) [10]	Characterizing DevOps in multiple perspectives in order to reduce the conceptual gap between the academic research and professional practices on this topic	B + R: DevOps definition and scope remains unclear in the scientific literature or among software practitioners	DevOps Characterization	Multivocal literature review, Grounded Theory	N/A	▪ Grounded definition for DevOps based on reviewing 43 different sources ▪ Identified DevOps recurrent principles, practices, required skills, potential benefits, challenges and what motivates the organizations to adopt it ▪ Further scientific investigations concerning the potential benefits and drawbacks needed
Erich et al. (2017) [11]	Determine whether DevOps has a positive effect on software organizations and their processes by reviewing the results of a systematic literature review and evaluation involving organizations in different industries	B: lack of evidence on the effectiveness of DevOps R: no consensus of what concepts DevOps covers, nor how DevOps is defined	DevOps effectiveness	Systematic literature review & exploratory interview-based study	Senior employees at a diverse set of organizations (6 in total)	▪ Comprehensive summary about DevOps in academic literature (40 papers) ▪ All involved organizations are positive about their experiences and only minor problems were encountered while adopting DevOps
Luz et al. (2018) [21]	Present a theory about DevOps adoption, highlighting the main related concepts that contribute to its adoption in industry	B: Increase the maturity level of DevOps adoption at a Brazilian Government institution R: Generating an adequate understanding of DevOps (practitioner's perspective)	Collaborative DevOps culture	Grounded Theory	15 companies across five countries that successfully adopted DevOps	▪ A theory on DevOps adoption and how to adopt it ▪ Core category of DevOps adoption is the collaborative culture ▪ Continuous measurement and quality assurance are both foundations
Agrawal & Rawat (2019) [2]	Understand how DevOps and Cloud work together to help businesses achieve their transformation goals	B: Need to automate DevOps processes using cloud and non-cloud DevOps automation tools to deliver services and applications at high velocity	DevOps & Cloud development	Prescriptive	N/A	▪ DevOps & Cloud Computing can accelerate innovation, improve collaboration, increase efficiency, reduce failure, and enhance job satisfaction ▪ Adopting new paths for development and operations at the same time is difficult to implement ▪ Further research/guidance needed to handle DevOps as a service to cloud applications
Bezemer et al. (2019) [4]	In-depth overview of how performance engineering is applied in DevOps	B: Get insights into how performance is addressed in industrial DevOps settings R: Many unanswered questions and research challenges in the area of performance evaluation in DevOps	DevOps performance management & evaluation	Industrial survey	134 professionals; Software development area	▪ Two third of participants do not conduct performance evaluation on a regular basis ▪ Low usage of performance models in practice ▪ Future research should focus on converting performance engineering practices into DevOps pipelines
Luz et al. (2019) [20]	Develop a model on DevOps adoption and investigate the relevance in a real scenario	B: Need for detailed guidance to support newcomers interested in adopting DevOps R: Process of DevOps adoption not adequately researched/explained	DevOps adoption	Grounded Theory, Focus Groups	15 practitioners of software companies from Brazil, Ireland, Portugal, Spain, and United States	▪ DevOps adoption involves seven categories: agility, automation, collaborative culture, continuous measurement, quality assurance, resilience, sharing and transparency ▪ Further research explorations / instantiations in other companies needed

Table 3. DevOps frameworks and its identified attributes

General Framework Attributes				Framework Design	
Name	**Author**	**DevOps Area**	**Purpose/ Goal**	**Key Concepts/ Elements**	**Description/Guidance on Use**
Report: DevOps Literature Review	Erich, Amrit & Daneva (2014) [11]	DevOps in general	Gain an understanding of DevOps in general and examine the research done in DevOps	DevOps is supported by a culture of collaboration, automation, measurement, information sharing and web service usage. DevOps has a positive effect on performance and quality assurance performance.	More research is needed, papers often low quality and little research has been done overall in the area of DevOps.
A Framework for Managing Mission Needs, Compliance and Trust in the DevOps Environment	Farroha & Farroha (2014) [13]	DevOps Management	Ensuring the continuity of strategic posturing while allowing maximum flexibility to tactical enhancements to meet emerging demands.	Important factors when implementing DevOps include prioritization, compliance and security, tools, and services as well a as policy.	Framework to be used by managers wishing to implement DevOps. Enables better collaboration
An Agile Framework for ITS Management in Organizations: A Case Study Based on DevOps	Abdelkebir, Maleh & Belaissaoui (2017) [1]	DevOps framework for ITSM	holistic and practical strategic framework to improve ITSM service management processes	Improving ITSM Service management processes by adding two drivers: 1 Agility management based on DevOps and 2. an agility process maturity framework (APMF)	Important aspects for a practical agile framework for ITSM efficiency has been proposed. Quality and efficiency improvements possible with this framework.
Enhancing Lean Software Development by using DevOps Practices	Farid, Helmy & Bahloul (2017) [12]	DevOps adoption and implementation	Improve the performance of lean software development production and introduce a new framework that merge lean and DevOps process	Enhancement of Lean Software Development process was done through determining the causes of the lean software development waste and how using DevOps practices in improving and addressing this waste	The paper proposes different methods to deal with waste in software development
DevOps: Concepts, Practices, Tools, Benefits and Challenges	Ghantous & Gill (2017) [6]	DevOps adoption and implementation	Identify Concepts, practices, tools, and challenges in DevOps	Communication and Collaboration, continuous deployment, delivery, feedback and planning automated pipeline, quality assurance and roll back code.	A catalogue has been developed, which provides a collective knowledge base of DevOps that can be used by researchers and practitioners to enhance their understanding and enable effective adoption
Modeling & measuring attributes influencing DevOps adoption in an enterprise using structural equation modeling	Gupta, Kapur & Kumar (2017) [15]	DevOps Implementation	Identify attributes influencing DevOps implementation	Four latent variables influencing DevOps → Source Control, Automation, Cohesive Teams, Continuous Delivery, resulting in various DevOps attributes	Use variables and attributes to assess and measure maturity level of DevOps implementation
Improve software quality through practicing DevOps	Perera, Perera & Silva (2017) [32]	DevOps in general	How DevOps practice has impacted to software quality	CAMS Framework (Culture, Automation, Measurement and Sharing) for improving software quality	Quality is important for customer satisfaction. Automation is the most critical success factor to improving software quality. Best practices have been proposed (TDD, BDD, ATDD)
Continuous Scrum: A Framework to Enhance Scrum with DevOps	Samarawickrama & Perera (2017) [27]	Capabilities and limitations of DevOps in practice	Overcome common challenges in software development	A framework that standardizes adoption of continuous integration. The expanded Scrum framework for DevOps performs better	A framework for the combination of Scrum and DevOps
DevOps Capabilities, Practices, and Challenges: Insights from a Case Study	Senapathi, Buchan & Osman (2018) [29]	DevOps Implementation	Identify attributes influencing DevOps implementation	Benefits arising from employing DevOps include improved user experience & higher team productivity, which in turn realizes a competitive advantage. DevOps is dependent on technological, team and capability enablers.	Overview over the benefits, drivers, and challenges in employing DevOps. Consult for a deeper understanding of issues as well as reasons for implementing DevOps
DevOps in practice: A multiple case study of five companies	Lwakatare et al. (2019) [22]	DevOps adoption and implementation	Determine how DevOps is implemented in practice	Attaining ownership and responsibility in software development as well as consistent toolchain usage leads to better results.	Deployment script validation practices should be used, as complex infrastructures with manual steps can cause reliability issues. Balance between speed and internal code quality is important.
Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach	Rafi, Yu & Akbar (2020) [33]	DevOps adoption and implementation	Use a Grounded Theory approach for DevOps Adoption	Raises security concerns in DevOps Adoption: 15 factors identified.	To successfully implement DevOps, (non-)functional security concerns must be addressed. From the survey of 13 practitioners, 15 security concerns were flagged and categorized.
SKI: A New Agile Framework that Supports DevOps, Continuous Delivery, and Lean Hypothesis Testing	Saltz & Sutherland (2020) [26]	Framework for effective DevOps support	Develop a framework to effectively support DevOps and Continuous Delivery teams	DevOps framework based on Kanban philosophy, enhanced by including a structured iteration process. Uses capability-based iterations. Establishes a well-defined process to be used consistently across groups.	Can be used in environments where a given project: 1. has the ability to rapidly realize iterations, 2. faces a significant degree of uncertainty 3. can dedicate a significant amount of their effort to new product development.

Table 4. Challenges and barriers

Author	Description	Main Challenges [16]					
		Cost	Risks	Scope	Quality	Business Value	Time
Jones et al. (2016) [17]	DevOps is an interdisciplinary topic which would greatly benefit from further management and potentially psychology-oriented research attention	Not mentioned	Having different agendas for their career progression and beliefs as to what their roles encompass	Having different agendas for their career progression and beliefs as to what their roles encompass	The necessity of maintaining legacy systems is one of the biggest issues the developers face	It follows, therefore, that these tasks are at the core of the employee-employer relationship, and job crafting	Not mentioned
Rigungu-Kalliosaari et al. (2016) [25]	A qualitative multi-case study. Interviews of representatives of three software development organizations in Finland.	Not mentioned	Difficult to determine what practices should be taken into consideration for DevOps	There are several changes, responsibilities shift, people must rethink their established roles and roles merge	Operations teams do not always pass or monitor all the performance and further metrics that can be useful to developers	A key impediment for successfully adopting DevOps is insufficient communication	developers worry about the release frequency whereas operations personnel are concerned with the uptime of servers
Ganthous & Gill (2017) [14]	DevOps originated in the context of agile software development, it seems to be an appropriate approach to enable the continuous delivery and deployment of working software in small releases.	Not mentioned	Not mentioned	Dev and Ops teams having separate toolsets and metrics	Establish and simulate an effective Agile-DevOps technology environment	Large scale enterprise agility depends on the most important human capability such as people competency and experience	Not mentioned
Gupta et al. (2019) [15]	This paper presents the experiences in a project of a software engineering team spread across three countries that successfully established continuous delivery	Ops automation in a cloud environment helps to control the ops cost	Automation helped to predict the risks and added more value to version increments	Release strategies compared to legacy fixed scope- based releases; value stream-based	Anticipating the quality and frequency of releases	Deliver higher business value with version increments	Delay in any milestone is too costly and impacts the timely release of the version increment
Leite et al. (2019) [19]	DevOps concepts and challenges presented in the literature. By associating these concepts with tools, they contributed to supporting practitioners in choosing a proper toolset.	DevOps aims to achieve some business outcomes, such as reducing risk and cost	DevOps aims to achieve some business outcomes, such as reducing risk and cost	Not mentioned	Metrics such as productivity were aggregated in a "total quality index".	Services interact through the network and are built around business capabilities	Significant deployment times, and the need for frequent and reliable releases led to inefficient execution of agile processes
Halstenberg et al. (2020) [16]	The meaning of DevOps has shifted in the past and other tools for DevOps keep evolving so one of the barriers is that the meaning of DevOps is unclear but also evolving.	Costs should remain calculable	Risks should remain low and assessable	The scope should be flexible based on the benefits	The quality level should be appropriate and measurable by criteria	The business value is the focus of all development activities as a business-customer- and stakeholder benefit	It is important to be able to react adequately to market changes

4. Discussion

This paper explores DevOps implementation by reviewing the relevant literature. Two main goals were identified: To uncover the challenges and barriers in the implementation process and to derive methods and approaches to overcoming these challenges.

The systematic literature review provides a broad overview of DevOps and its practices. Recently, the scientific community has conducted an increased number of research papers into DevOps. Today, the demand for customized IT solutions has increased significantly. Furthermore, in this digital world, customers demand fast results in order to accommodate the changing needs. With the fast-paced changes, traditional software development approaches have failed and need to be overhauled because of their lack of adaptation and non-agile proceeding.

Additionally, in order to meet today's customers' needs and satisfy their demand, a continuous delivery of digital products is a must. This goal can be achieved by implementing DevOps as a way to merge operational and development teams. This combination allows organizations to communicate more efficiently and makes delivery of software faster and more convenient to the customer's needs. The changing nature of software development is reflected in the definition process of DevOps. No single homogenous definition for DevOps has been defined by research. However, many definitions include common aspects outlined in this paper. Furthermore, the conflicting views (DevOps as a cultural movement vs. DevOps as a job description) make it evident, that there is a divide regarding the definition [24].

Common models include the CAMM or CAMS framework, which include a few key aspects, which define DevOps. The four dimensions are Culture, Automation, Measurement and Monitoring or Sharing. The first dimension encompasses aspects related to teamwork and collaboration. Open communication is considered to be a crucial factor when adopting DevOps especially in large teams and organizations. Smaller teams are often not as separated and experience more intensive communication. The automation aspect is based on a technical level [23].

The tools that are used to collaborate, measure, monitor and automate are equally important. However, it is important to note, that there are no universal solutions. The selection of the right tools in the appropriate environment with a specialized team is key to adapting and implementing DevOps.

RQ1: What challenges/barriers exist in SMEs when introducing DevOps?

DevOps is often adopted to benefit from higher productivity, lower delivery time, automated processes, and greater customer satisfaction.

However, implementing DevOps can be quite **challenging**. Overcoming these challenges is key to the success of DevOps. As shown in earlier chapters, there are technical as well as interpersonal **barriers**. Common challenges include lack of a clear definition (1), deep-seated company culture (2), insufficient communication (3), geographical distribution (4) and organisation structure (5) [23].

Apart from the challenge to adopt DevOps in an enterprise, the operation in a DevOps environment is also challenging. The literature does not provide recommendations or a model how to deal with the different views of development and operation teams. Nevertheless, a consensus on the importance of DevOps in an organization should be fostered, as this can help mitigate communication issues.

RQ2: How can companies overcome these challenges?

In order to overcome the challenges of the implementation of DevOps, companies must work towards a common goal. This can be achieved with a supportive management and a team that is willing to change the corporate culture that is aligned with the DevOps ideal. Recognizing and dealing with these issues appropriately is important. Combined with the increasing demand of "state-of-the-art IT-solutions" in the market and unpredictability of upcoming innovations, enterprises need to focus on leadership aspects to adopt DevOps successfully. Furthermore, since organizations and their needs are ever-changing and unique, no universal framework or solution to adopt DevOps in an existing infrastructure, can be defined.

The major cultural shift is probably the biggest obstacle to moving forward, bigger than process or current technical competencies. That is the point where leadership can play an important role. Management must be supportive and change the corporate culture to match the ideal DevOps setup.

5. Conclusion and directions for further research

This paper's aim is to identify challenges and barriers that come up in DevOps implementations and discussing possible causes and methods to overcome them. This was done through a literature review, evaluation, and discussion of the results. The main inten-

tion was to create a comprehensible and deeper understanding that would provide a basis and motivate future researchers for broader investigations in this area.

From the literature perspective, it is clear that DevOps is a very interdisciplinary field that has not yet fully exhausted the research needs and where many different research potentials still exist.

While the DevOps literature continues to grow, there is a diversity of meanings, aspirations and technologies that poses confusion and ambiguity for future research. Different publications highlight this experience and show that a common understanding and definition is lacking in academia but also among practitioner communities.

While much of the previous research has examined usage, characteristics and benefits of DevOps, there is a lack of research of practical significance that companies can adopt and use in practice to drive a successful DevOps implementation.

Based on the literature and the earlier discussion, it can be summarized that DevOps implementations consist of the following categories that are interrelated and have an interaction: Agility, Collaboration, Automation, Measurement, Monitoring and Transparency.

Implementing these aspects is crucial for a successful DevOps adoption. Therefore, it requires interaction between development and operations on a team and departmental but also individual level. If these aspects are not given the necessary attention early on, challenges arise that become more complicated to overcome later.

Therefore, early adopters should develop appropriate communication strategies and define and embody an organization-wide DevOps culture. This also requires embedding policies and practices (e.g. DevOps principles, Agile Manifesto) into the organizational structure. These should be accompanied by competent and experienced leadership that supports on an individual and mental level.

Given the complexities in the field of DevOps, a one-size-fits-all approach is questionable and calls for a novel approach to the adoption of DevOps considering given domain-specific conditions, particularities, and constraints e.g. in SME.

Based on the principles of method engineering, additional effort could be spent on the development of a method for the adoption of DevOps. It has to be noted that the proposed method should not directly aim at changing the implied styles or patterns of behavior discussed in the previous sections, but rather should take these issues for granted and should seek to improve an organization's ability in dealing with the management of DevOps.

To address the increased need for research with practical contextual rich results, we will focus on small and medium-sized Swiss enterprises (SMEs) in a second part of our research. The investigation of different use cases of DevOps implementations in different organizations offers the opportunity to validate the theoretical insights gained from this paper and to test them for practical relevance.

Based on this, we seek to better understand the quantifiable but also cognitive and social aspects of DevOps adoption within our practical work. Furthermore, the limitations that only become clear when time, cost and human resources are included in the research will become evident.

With this goal in mind, iterative qualitative research will be conducted, including interviews and focus groups. The goal is to explore the possibilities, provide assessments of the feasibility, usefulness, and subsequent measurement of success of a smooth DevOps implementation, and recommend when and in what context the introduction of DevOps is justified.

This will be particularly helpful for smaller organizations that are considering a DevOps implementation and are uncertain about whether or how to start.

6. Limitations

Although this literature review was conducted in a disciplined manner, potential limitations must be acknowledged.

There is a potential risk of missing relevant literature since all papers written in other languages than English were excluded. In addition, considering the previous section, it becomes obvious that a structured approach to the adoption of DevOps, especially for SME, could be a reasonable area of future research.

In this context, the paper at hand explored literature sources with a strong academic reference, which excludes contributions from practice such as software companies or consultancies.

Another possible limitation is selection bias. To reduce this threat, the review was conducted and validated by three independent researchers.

Another area of concern is that this paper covers publications that were published before the end of 2021. As a conclusion, the results of this review can quickly become outdated since the number of new DevOps contributions dealing with challenges and barriers is increasing constantly.

Finally, it has to be noted that the paper at hand is an ongoing research, which deals with the theoretical aspect in the first part. Therefore, this work is still in the early stages and should not be considered fully comprehensive and conclusive. Nevertheless, it offers first insights and illuminates the DevOps phenomenon from the literature viewpoint. It contributes to DevOps

research by summarising what has been written in the academic literature on DevOps adoption. This can serve as a basis for researchers to get an overview of the academic literature and to further explore potential gaps.

7. References

[1] Abdelkebir, S. A. H. I. D., Maleh, Y., & Belaissaoui, M. (2017). An agile framework for ITS management In organizations: A case study based on DevOps. In Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems. 1-8.

[2] Agrawal, P., & Rawat, N. (2019). DevOps, a new approach to cloud development & testing. In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). (1). 1-4.

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

[4] Bezemer, C. P., Eismann, S., Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., ... & Willnecker, F. (2019, April). How is performance addressed in DevOps?. In Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. 45-50.

[5] Bordeleau, F., Cabot, J., Dingel, J., Rabil, B. S., & Renaud, P. (2019). Towards modeling framework for DevOps: requirements derived from industry use case. In International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. 139-151.

[6] Bou Ghantous, G., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *PACIS2017*.

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

[8] Capizzi, G., Sciuto, G. L., Napoli, C., Połap, D., & Woźniak, M. (2019). Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning. *IEEE Transactions on Fuzzy Systems*, *28*(6), 1178-1189.

[9] Cooper, Harris M. "The Structure of Knowledge Synthesis." Knowledge in Society 1, no. 1 (1988): pp. 104–126.

[10] De França, B. B. N., Jeronimo, H., & Travassos, G. H. (2016). Characterizing DevOps by hearing multiple voices. In Proceedings of the 30th Brazilian symposium on software engineering. 53-62.

[11] Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, *29*(6).

[12] Farid, A. B., Helmy, Y. M., & Bahloul, M. M. (2017). Enhancing lean software development by using DevOps practices. *International Journal of Advanced Computer Science and Applications*, *8*(7), 267-277.

[13] Farroha, B. S., & Farroha, D. L. (2014). A framework for managing mission needs, compliance, and trust in the DevOps environment. In *2014 IEEE Military Communications Conference*. 288-293.

[14] Ganthous, G., & Gill, A. (2017). DevOps: Concepts, Practices, Tools, Benefits and Challenges. Association for Information Systems, AIS Electronic Library (AISeL). Sydney: Pacific Asia Conference on Information Systems (PACIS).

[15] Gupta, V., Kapur, P. K., & Kumar, D. (2017). Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling. *Information and software technology*, *92*, 75-91.

[16] Halstenberg, J., Pfitzinger, B., & Jestädt, T. (2020). DevOps - Ein Überblick. Wiesbaden: Springer Vieweg.

[17] Jones, S., Noppen, J., & Lettice, F. (2016). Management Challenges for DevOps Adoption within UK SMEs. Saarbrücken: QUDOS.

[18] Kloos, U., Martinez, N., & Tullius, G. (2018). Informatics Inside connect (IT). Reutlingen: INF Informatik.

[19] Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys (CSUR)*, *52*(6), 1-35.

[20] Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, *157*, 110384.

[21] Luz, W. P., Pinto, G., & Bonifácio, R. (2018). Building a collaborative culture: a grounded theory of well succeeded devops adoption in practice. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 1-10.

[22] Lwakatare, L. E, et al. (2019). DevOps in practice: A multiple case study of five companies. Information and Software Technology, Volume 114, 217-230

[23] Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of devops. In *International conference on agile software development*. 212-217.

[24] Macarthy, R. W., & Bass, J. M. (2020). An empirical taxonomy of DevOps in practice. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 221-228.

[25] Rigungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L., Tiihonen, J., & Männistö, T. (2016). DevOps Adoption Benefits and Challanges in Practice: A Case Study. Department of Computer Science, Department of Information Processing Science. Helsinki: Springer International Publishing AG.

[26] Saltz, J., & Sutherland, A. (2020). SKI: A new agile framework that supports DevOps, continuous delivery, and lean hypothesis testing. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

[27] Samarawickrama, S. S., & Perera, I. (2017). Continuous scrum: A framework to enhance scrum with DevOps. In *2017 Seventeenth international conference on advances in ICT for emerging regions (ICTer)*. 1-7.

[28] Saunders, Mark, Lewis, Philip, and Thornhill, Adrian. Research Methods for Business Students. 6th ed. Harlow, UK: Pearson, 2012.

[29] Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps capabilities, practices, and challenges: insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. 57-67.

[30] Smeds, J., Nybom, K., & Porres, I. (2015). DevOps: a definition and perceived adoption impediments. In *International conference on agile software development*. 166-177.

[31] Söllner, D. (2017). DevOps in der Praxis - Handlungsfelder für eine erfolgreiche Zusammenarbeit von Entwicklung und Betrieb. Wiesenbaden: CrossMark.

[32] Perera, P., Silva, R., & Perera, I. (2017). Improve software quality through practicing DevOps. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. 1-6.

[33] Rafi, S., Yu, W., & Akbar, M. A. (2020). Towards a hypothetical framework to secure DevOps adoption: Grounded theory approach. In *Proceedings of the Evaluation and Assessment in Software Engineering*. 457-462.