Thesis no: MSSE-2016-28

Continuous Integration, Deployment and Testing in DevOps Environment

Anand Srivatsav Amaradri

Swetha Bindu Nutalapati

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Anand Srivatsav Amaradri
E-mail: anam15@student.bth.se

Swetha Bindu Nutalapati
E-mail: swnu15@student.bth.se

University advisor:
Ahmad Nauman Ghazi
Department of Software Engineering

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Context. Owing to a multitude of factors like rapid changes in technology, market needs, and business competitiveness, software companies these days are facing pressure to deliver software rapidly and on a frequent basis. For frequent and faster delivery, companies should be lean and agile in all phases of the software development life cycle. An approach called DevOps, which is based on agile principles has come into play. DevOps bridges the gap between development and operations teams and facilitates faster product delivery. The DevOps phenomenon has gained a wide popularity in the past few years, and several companies are adopting DevOps to leverage its perceived benefits. However, the organizations may face several challenges while adopting DevOps. There is a need to obtain a clear understanding of how DevOps functions in an organization.
Objectives. The main aim of this study is to provide a clear understanding about how DevOps works in an organization to researchers and software practitioners. The objectives of the study are to identify the benefits of implementing DevOps in organizations where agile development is in practice, the challenges faced by organizations during DevOps adoption, to identify the solutions/ mitigation strategies to overcome the challenges, the DevOps practices, and the problems faced by DevOps teams during continuous integration, deployment and testing.
Methods. A mixed methods approach having both qualitative and quantitative research methods is used to accomplish the research objectives. A Systematic Literature Review is conducted to identify the benefits and challenges of DevOps adoption, and the DevOps practices. Interviews are conducted to further validate the SLR findings, and identify the solutions to overcome DevOps adoption challenges, and the DevOps practices. The SLR and interview results are mapped, and a survey questionnaire is designed. The survey is conducted to validate the qualitative data, and to identify the other benefits and challenges of DevOps adoption, solutions to overcome the challenges, DevOps practices, and the problems faced by DevOps teams during continuous integration, deployment and testing.

Results. 31 primary studies relevant to the research are identified for conducting the SLR. After analysing the primary studies, an initial list of the benefits and challenges of DevOps adoption, and the DevOps practices is obtained. Based on the SLR findings, a semi-structured interview questionnaire is designed, and interviews are conducted. The interview data is thematically coded, and a list of the benefits, challenges of DevOps adoption and solutions to overcome them, DevOps practices, and problems faced by DevOps teams is obtained. The survey responses are statistically analysed, and a final list of the benefits of adopting DevOps, the adoption challenges and solutions to overcome them, DevOps practices and problems faced by DevOps teams is obtained.
Conclusions. Using the mixed methods approach, a final list of the benefits of adopting DevOps, DevOps adoption challenges, solutions to overcome the challenges, practices of DevOps, and the problems faced by DevOps teams during continuous integration, deployment and testing is obtained. The list is clearly elucidated in the document. The final list can aid researchers and software practitioners in obtaining a better understanding regarding the functioning and adoption of DevOps. Also, it has been observed that there is a need for more empirical research in this domain.

Keywords: DevOps, software development, development, operations, benefits, challenges, principles, practices.

Acknowledgments

We would like to thank our supervisor Mr. Ahmad Nauman Ghazi for his invaluable support, patience and guidance throughout our thesis. Without his help and motivation, we would not have completed our thesis. We would like to thank our families for supporting us throughout our journey. Our heartfelt thanks to all the practitioners who have taken the time to help us out.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

A multitude of factors like increasing business competitiveness, rapid changes in technology and market needs have changed the environment in which software companies operate [14] [47]. Many enterprises as a result face a common challenge: faster and frequent software delivery [53]. Those times when a customer waited for months for a product to release, and then provided feedback have passed. Customers expect fast responses to their ever-changing requirements. Frequent releases are essential for the customer to provide continuous feedback [53]. Companies like facebook are deploying software to customers on a regular basis [29]. While continuously delivering software to customers offers a business advantage over other rival companies, it is not easy to do it because some challenges, technical and non-technical have to be overcome first [29]. The companies must be lean and agile throughout the entire software development lifecycle in order to overcome these challenges [47].

Agile software development is popular in several enterprises because of its plethora of benefits like reduced development time, increased project success rate, reduced development cost and increased customer satisfaction. Agile development increases the responsiveness and the stakeholders' needs will be met on time. Agile methods facilitate the collaboration among the development teams [34]. A few examples of agile methodologies include Scrum, XP, Kanban, DSDM, Lean, Crystal etc. [23]. The various agile methodologies are based on iterative and incremental development, and they increase communication between the development team and the stakeholders [34].

However, the main focus in agile methodologies has been only on the development side, ignoring the operations side [29]. The operations teams deploy, manage and support systems' performance at the customers' site [29]. Owing to this, the development teams deliver builds much faster and the operations teams cannot keep up with the pace, and they will lag behind. This may cause the entire delivery to be delayed [47]. As the development and IT operations are separate organizational units, there is a high possibility of misunderstandings and conflict [49]. According to some software practitioners, this gap between the development and operations teams has a negative effect [13].

To bridge this gap and increase the communication, collaboration and integration between the developers and operations personnel, a new approach called DevOps has

come into play [49]. “Dev”+”Ops”=”DevOps” [14]. While the agile methodologies improved the development teams’ performance through collaboration, DevOps extends the collaboration of development towards operations [29]. DevOps is based on agile principles, and it encourages faster development and deployment cycles [34]. DevOps is a loosely defined set of practices intended to make the developers and operations personnel to work together. The overall objective behind DevOps is to maximize the return of investment and increase the customer satisfaction by continuously providing features and increased service quality [14]. Several companies like Facebook, Yahoo, Netflix, Flickr and Fotopedia are implementing DevOps [29] [3] [13] [15].

So, how do we define DevOps? DevOps is a vague term which has different meanings in different contexts [37]. It is often defined as a combination of development and operations [3] [37]. Dyck et al. [10] have defined DevOps as an "organizational approach that stresses empathy and cross-functional collaboration within and between teams - especially development and IT operations - in software development organizations, in order to operate resilient systems and accelerate delivery of changes.” According to Roche [38], there is no standard definition for DevOps. There is much debate surrounding on what DevOps really means. He states that some blog posts have summarized DevOps as a specific job which needs the skills of both development and operations, while some others argue that there is no such thing. They believe that DevOps is a new criteria for development, testing, support etc. He points out that many companies take the former view, given that there are several job listings that seek for “DevOps Engineers” [38]. Erich et al. [13] have defined DevOps as a “conceptual framework for supporting development and operations.” From the literature, we have understood DevOps as a framework of principles and practices that organizations can include in their development process.

Peuraneimi [37] refers to DevOps as “agile on steroids.” DevOps includes several principles from the agile methodologies and also some Lean principles. It is a way of managing the software development lifecycle [37]. DevOps is not limited to just development and operations teams [10]. Various stakeholders including business analysts, developers, testers, quality assurance personnel and operations personnel (system, database and network administrators, webmasters and security officers) are involved in DevOps [3]. DevOps covers all the aspects that aid in the delivery of fast, optimized and high quality software [47]. There are four main aspects of DevOps: Culture, Automation, Measurement and Sharing [20]. The main focus on DevOps is on enabling communication and collaboration between teams, not on the standards or tools in an organization [19]. DevOps is facilitated by cloud computing [3]. Usually, it is involved in cloud based services like Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [37]. Tools which are used to support DevOps include Puppet, Chef, Bladelogic, Salt, Amazon WorkOps, Docker, Vagrant etc. [37] [20] [3].

Let us take a look at the main aspects of DevOps: Culture, Automation, Measurement and Sharing [20].

- Culture implies "people over processes and tools" [37]. It means that operations must take part in development and deployment [20]. Operations representatives must attend retrospectives and meetings of the development teams. Similarly, development representatives must have frequent meetings with the operations teams.
- Automating the build, deployment and testing processes is essential for obtaining rapid feedback and achieving low lead times. A deployment pipeline is used for automating the processes. If any changes are made in the system, each change will be validated by passing it through a series of automated tests in the pipeline. If the changes are successful, they will be ready for deployment to the testing, staging and production environments [20].
- Measurement involves monitoring of both high-level and low-level business metrics. High-level metrics include revenue, transactions per unit time. At low-level, people are measured. To ensure the behavior of people is not influenced by measurement, key performance indicators must be carefully chosen. The metrics should be made visible to everyone involved in the software delivery so that they can see how they are performing [20].
- Sharing includes sharing of knowledge, sharing of development tools and sharing of techniques for managing environments and infrastructure by the development and operations teams. A good practice is for the two teams to celebrate together when a release is successful [20].

Lwakatare et al. [29] have identified another aspect of DevOps: Monitoring. Operations teams use certain tools and logs to monitor the systems for any issues. Usually the logs can be very large and identifying the issues can be a tedious task for the development and operations teams. Also, no details about the issues will be visible to the teams because of the system design. DevOps ensures that the two teams collaborate to design systems in such a way that details will be displayed. Thus, monitoring will be more effective [29].

Peuraniemi [37] identified Infrastructure as Code (IaC) as the main principle of DevOps. IaC means automatically managing the configurations. Using IaC, environments can be configured faster and infrastructural changes can be made more easily. Subsequently, faster deliveries can be made to the customer. He also identified four principles of collaboration: Respect for one another, Commitment to shared goals, Collective ownership and Shared values [37]. All team members should respect the values of each other and interact well. Everyone should be committed to the shared goals and work towards the success of the project. Collective ownership can ensure that quality of the developed software is increased by using optimal solutions for development. Shared values guarantee customer satisfaction by giving more importance

to working system and rapid delivery, rather than formal plans [37]. Fietelson et al. state that a culture of personal responsibility is important and methodologies and tools alone are not adequate, as they can always be exploited [15].

Virmani [47] states that DevOps aims to address various problems related to software projects, such as adaptability to change, increasing quality, cost reduction and speed to market. Farroha et al. [14] claim that the main focus of DevOps is on: Continuous and high-quality software delivery, agility and simplicity in technology, processes and human factors, breaking down the barriers through collaboration between development and operations teams. The DevOps principles can be applied to any type of projects, including complex projects, small projects and mobile application projects [47]. Various factors like the type of project involved, organizational needs and capabilities and return on investment (ROI) will determine which principles an organization needs to adapt. The organization will decide how and using what tools and technologies it adopts a principle or a practice of DevOps [47].

The practices or the enabling techniques of DevOps are: Continuous planning, Continuous integration, Continuous deployment, Continuous testing, Continuous delivery, Continuous monitoring and Continuous feedback [42] [49] [47] [17]. These practices are known to facilitate rapid delivery of software to the customers [47]. Continuous planning makes it easier for the teams to adapt to the fast changes in business environments by having a prioritized product backlog in hand and a channel that enables continuous feedback from customers. Changes can be made to the backlog any time, based on the feedback obtained [47]. Continuous integration involves an automated process where code compilation and sanity tests will be carried out and deployment packages will be built, as soon as a developer makes a change to the code [47] [17]. Continuous deployment involves automated and continuous deployment of valid software builds to a repository, rather than to the users [17]. Continuous testing involves automation of all test cases in such a way that no user intervention is required, and errors in the builds will be detected faster [47] [17]. Continuous delivery is the practice of continuously deploying software, while ensuring that it is always ready to be released and deployed to the customers [17]. Continuous monitoring enables monitoring of quality parameters throughout the software development, thereby facilitating easier detection and solving of problems [47]. Continuous feedback is facilitated by continuous monitoring of both infrastructure and user behavior. The data related to infrastructure performance and user behavior when interacting with the service will be obtained by continuous monitoring. This data will be used as input to the planning and development processes for improving the service [42].

Virmani [47] notes that different teams in a single organization might need to use different tools to adopt the DevOps practices. According to McCarthy et al. [30] a change in the mindset of everyone involved is crucial to successfully implement DevOps in an organization. It is important that everyone shares the responsibility for both success

and failure of products [30]. Smeds et al. [42] state that adopting DevOps in an organization is not an easy task, and each organization has a unique way of making the adoption process successful. Virmani [47] stresses that it might not be practical for an organization to move to a full-DevOps state from a non-DevOps state in a brief period of time. He states that organizations should analyse the current workflows and target those areas which have a scope for optimization [47]. The related work is discussed in the next section.

1.2 Related Work

Farroha et al. [14] proposed a framework for ensuring that the overall enterprise performance and security are not compromised while adopting DevOps, as it was observed that DevOps lacks the discipline that engineers are normally used to. Stillwell and Coutinho [45] describe a DevOps approach that was used for integration of software components in a European research project. They have found that the DevOps workflow has caused the development teams to be more open to changes and operate autonomously, providing more frequent updates. They claim that the approach reduced the communication overhead and increased the delivery speed. They note that it is hard to completely quantify the effects, as they just started collecting metrics [45]. Bang et al. [3] identified the Knowledge, Skills and Abilities (KSA) needed for both developers and operations personnel to support the main aspects of DevOps: Culture, Automation, Measurement and Sharing. Erich et al. [13] conducted a mapping study on the cooperation between development and operations teams in organizations. They found that DevOps improves the performance of both development and operations teams, and also improves the quality assurance performance. They note that more research is needed to quantify these benefits [13]. Roche [38] describes about the adoption of DevOps practices into software quality assurance. He states that extending the collaboration between development and operations teams to quality assurance helps the teams in making better decisions on the product development and releases.

Gottesheim [19] identifies the most common issues that occur during software development and describes about enabling performance focused DevOps in organizations. He states that problems like guesswork and finger pointing between teams when faced with issues can be avoided by defining and sharing performance metrics across all teams. Shahin [40] states that DevOps and continuous deployment can prove to be challenging for software architects, as applications should be re-architected for supporting the various DevOps practices. McCarthy et al. [30] proposed a framework to incrementally improve the existing DevOps practices into more collaborative and cohesive ones, and measure the collaboration value. Fitzgerald and Stol [17] identified several continuous activities such as continuous integration, deployment, testing, improvement etc. which are crucial for software development in the present business environment.

Smeds et al. [42] identified some challenges faced by software organizations during DevOps adoption. Olszweska and Walden [34] describe about how formal modelling works in DevOps and how quality can be achieved in formal modelling in DevOps. A new language called DevOpSlang was introduced by Wettinger et al. [53] for implementing DevOps in organizations. Lwakatare et al. [29] have identified the main aspects of DevOps as Collaboration, Automation, Measurement and Monitoring, and also developed a framework to provide an understanding of how DevOps works. Wettinger et al. [51] describe about how diverse application environments can be used for implementing DevOps.

Waller et al. [49] state that benchmarks and monitoring should be included into software development right at the beginning, without waiting until problems occur during release. They claim that automated benchmarks help in earlier detection of issues, thereby prevent them from spreading through the deployment pipeline. Cois et al. [7] proposed a model of automated DevOps for guiding organizations on responsibilities and the DevOps infrastructure. They state that automation improves the communication rate and knowledge transfer. Hussaini [21] proposed a model to identify the common key objectives of both Dev and Ops and improving the collaboration between the development and operations units in an organization.

Wahaballa et al. [48] identified a problem called conceptual deficit, which is caused by the collaboration between development and operations teams during software deployment. They proposed a Unified DevOps Model (UDOM) for overcoming this problem. Dyck et al. [10] differentiate between release engineering and DevOps, and provide definitions for both terms. Wettinger et al. [50] proposed a holistic approach for capturing DevOps knowledge into a knowledgebase and managing it. Tessem and Iden [46] stated that poor cooperation between development and operations teams leads to reduced user satisfaction and reduced productivity of both the teams, well before the term DevOps came into light. Erich et al. [12] note that there is no specific way of doing DevOps that all companies can follow. Rather, there are various principles and practices, and the organizations need to include the required principles and practices into their development processes [12]. Mohamed [32] introduced a DevOps maturity model for altering the software development life cycle to adapt with DevOps. Mohamed [31] also proposed a tool called DevOps Maturity Calculator (DOMC) for calculating the DevOps maturity level of organizations. He claims that DOMC enables the organizations to include the DevOps maturity model into their processes, and respond faster to customer requirements.

Bass et al. [4] argue that along with communicating with operations teams, developers should also look into several other aspects like standards, organizational process descriptions, studies about types of operations failures, etc. in order to understand the issues associated with operations. Kerzazi and Adams [24] identified the main tasks of release and DevOps engineers across the world by conducting an empirical analysis

of online job listings. They found that automation or scripting is the most important task. Babar et al. [1] proposed a model that aids organizations to select customized DevOps processes to fit their needs. Virmani [47] describes about DevOps applied to various stages of software delivery life cycle, the things organizations need to consider while adopting DevOps, and the perceived benefits of implementing DevOps in organizations. Jones et al. [22] identified the challenges to DevOps adoption by conducting a case study in a UK based company. Lwakatare et al. [28] identified the challenges to DevOps adoption in the embedded systems domain.

It has been noted that not much literature is available on DevOps, and most of the studies are of low quality. There are very few studies which provide a concrete understanding about how DevOps works in an organization. Smeds et al. [42] identified the possible challenges than organizations may face while adopting DevOps. However, their study was conducted on only one organization, and in the initial stages of DevOps adoption. They mentioned the need to include more organizations in further research for generalizing the challenges of DevOps adoption in organizations. They state that more research needs to be done on the later stages of adoption. There is a need to provide solutions for overcoming the challenges in DevOps adoption. Further, the DevOps principles and practices are not clearly explained in literature. We try to delve deeper into the functioning of the different principles and practices of DevOps. We try to confirm if any of the perceived benefits of adopting DevOps are actually achieved in organizations which practice agile development. No studies have addressed the challenges that DevOps teams may face during the continuous activities like continuous integration, deployment and testing. We try to investigate the challenges faced by teams during the different continuous activities. We attempt to address this research gap by answering the research questions stated in the next chapter.

1.3 Aim and Objectives

These days, companies need to deliver the software products continuously in order to allow the customers to utilize their services. To do so, they need to follow a structured software development method. With most of the software companies being attracted to the values and principles of agile, product development using agile project management is increasing. To reach the expectations and needs of the customers, the organizations need to advance their development cycle and reduce the product time to market [47]. However, the organizations face many challenges owing to the time pressure, which affects the software quality. The main reason behind this is due to the gaps in between understanding the needs of customer, the development and testing teams and the operations teams. DevOps, which is a collaboration between the development and operations teams aims to reduce these gaps [7]. This approach acts as a bridge between the development teams and the operations teams. It has been found that DevOps can increase the communication, reliability and trust between the development

and operations teams [33]. By overcoming the barriers between the development and operations teams, organizations can respond quickly to the customer demands and support various changes which help them in yielding better outcomes [51].

The main aim of this thesis is to learn about how DevOps works in an organization. The focus of the thesis is on identifying how implementation of DevOps helps organizations where agile development is in practice, the DevOps adoption challenges and solutions to overcome them, the different DevOps practices, and the problems that DevOps teams may face during continuous integration, deployment and testing.

The objectives of this research are as follows:

1) To identify the benefits of implementing DevOps in organizations where agile is in practice.
2) To identify the challenges faced by organizations while adopting DevOps.
3) Identifying the strategies/ solutions to overcome the DevOps adoption challenges.
4) Identifying the practices that organizations must include for adopting DevOps.
5) To identify the problems that are faced by the DevOps teams during continuous integration, deployment and testing.

1.4 Research Contributions

The contributions of this research are as follows:

- A list of the benefits of implementing DevOps in organizations where agile development is in practice.
- A list of the challenges faced by organizations in adopting DevOps.
- A list of strategies/solutions to overcome the challenges of adopting DevOps.
- A list of practices to be included in an organization to adopt DevOps.
- A list of the problems faced by the DevOps teams during continuous integration, deployment and testing.

1.5 Overview of the thesis

The remainder of the thesis is structured as follows:

In Chapter 2, the research method is discussed. The research questions are mentioned, along with the motivation behind each question. The chosen mixed methods approach is elucidated, and the rationale for choosing the method is specified. The selected qualitative and quantitative research methods (SLR, interviews and survey) are elucidated

and how the research questions are answered using different methods is explained.

In Chapter 3, the Systematic Literature Review is discussed. The execution of Systematic Literature Review is explained, and the results of SLR are elucidated. The discussions and conclusion are presented. The possible threats to validity of the SLR results are discussed.

In Chapter 4, the analysis of interview data is explained. The execution of interviews is explained. The interview results, discussions, and validity threats are presented.

In Chapter 5, the analysis of survey results is explained. The execution of the online survey is explained, and the results are presented. The possible threats to validity of the survey results are discussed.

In Chapter 6, the results obtained from SLR, interviews and survey are mapped to answer the research questions.

In Chapter 7, the conclusions made from the thesis are explained, and the scope for future work is discussed.

Chapter 2

Research Method

2.1 Research Questions

The thesis aims to address the research gap by answering the following research questions:

RQ1) How does implementing DevOps in the organizations where agile is in practice help them?
Motivation: This research question aims at looking at the benefits of adopting DevOps in organizations where agile is in practice. The perceived benefits have a crucial role in determining whether or not to implement DevOps in an organization. We aim to look at the perceived benefits of adopting DevOps which are listed in literature, and also the actual benefits achieved in practice by organizations.

RQ2) What are the challenges faced by the organizations while adopting DevOps?
Motivation: It is very important to identify the challenges to the DevOps adoption process. As this thesis aims at identifying the challenges in the DevOps adoption process, it can contribute to the research pool and also help the organizations by letting them know about the risks involved with adopting DevOps.

RQ3) How to overcome the challenges faced by the organizations in adopting DevOps?
Motivation: Identifying the strategies to overcome the challenges inherent with DevOps adoption can effectively guide the organizations whenever they are faced with an impediment. Also, several problems can be avoided in the first place if the organizations know about the challenges in DevOps adoption and their respective mitigation strategies. Thus, there is a need to identify the strategies to overcome the challenges faced by the organizations in adopting DevOps.

RQ4) What are the practices that an organization needs to include while adopting DevOps?
Motivation: There is a need to know about what exactly are the practices that an organization needs to include as a part of the DevOps adoption process. There is no proper guidance in existing literature about the practices and principles of DevOps. This research question aims at identifying the practices that an organization needs to include while adopting DevOps.

RQ5) What are the problems faced by the DevOps teams during continuous integration, deployment and testing?
Motivation: DevOps involves continuous integration, deployment and testing [47]. The DevOps teams need to work together to continuously integrate, deploy and test the code. This may prove to be challenging for the DevOps teams. Answering this research question can help to identify the problems that the teams may face during continuous integration, deployment and testing.

2.2 Research Method

We employed a mixed methods approach for our research. Generally, there are three types of research methods: Qualitative, Quantitative and Mixed methods [8]. One major difference between the qualitative and quantitative methods is that qualitative methods use words and open-ended questions, while quantitative methods use numbers and closed-ended questions. The qualitative methods are flexible in nature, and they involve open-ended questions. The data obtained will be either in text or audio-visual format, and is analysed based on themes and patterns [8]. Some examples of qualitative methods are grounded theory analysis, interviews and literature reviews. Quantitative methods are more structured, with closed-ended questions. The data obtained will be numeric, and is statistically analysed. Examples of quantitative methods include experiments and surveys [8].

The mixed methods approach combines both qualitative and quantitative data. This approach is particularly useful when a researcher wants to generalize his findings to a population and also give a detailed understanding about a concept to the readers. The research can take more time when using this method, because both qualitative and quantitative data needs to be collected and analysed [8]. The motivation behind choosing this approach is that we can get more reliable and valid results than using either a qualitative or quantitative approach alone. The qualitative research methods used in this research are Systematic Literature Review and Skype Interviews. A survey involving an online questionnaire is employed as the quantitative research method. RQ1, RQ2 and RQ4 are answered using SLR. All the research questions, RQ1, RQ2, RQ3, RQ4 and RQ5 are answered using interviews and survey.

2.3 Rationale for choosing mixed methods approach

Our research aims at identifying all possible challenges that an organization may face while adopting DevOps, during both the initial and later stages of adoption, and providing solutions to overcome these challenges. We also try to identify the practices which organizations include as a part of DevOps adoption process, the benefits of adopting DevOps which are achieved in practice by organizations, and the problems faced by

DevOps teams during various continuous activities. DevOps is a relatively new concept, and not much literature is available. There are very few good quality studies which provide a concrete understanding of DevOps. There is a need to provide comprehensive and reliable results, and thus we employed the mixed methods approach.

We follow an exploratory sequential mixed methods design. In this process, we first gather the qualitative data, analyse it and then use the results in the quantitative phase. The purpose of following this process is to validate if the data from a small population sample can be generalized to a larger sample [8]. So, first we conduct a Systematic Literature Review, followed by Interviews. Based on the results of the qualitative methods, we devise the survey questionnaire, which will be used to gather the quantitative data. Because we are using both quantitative and qualitative methods to obtain results, the data will be triangulated. Thus, the research will be validated [41].

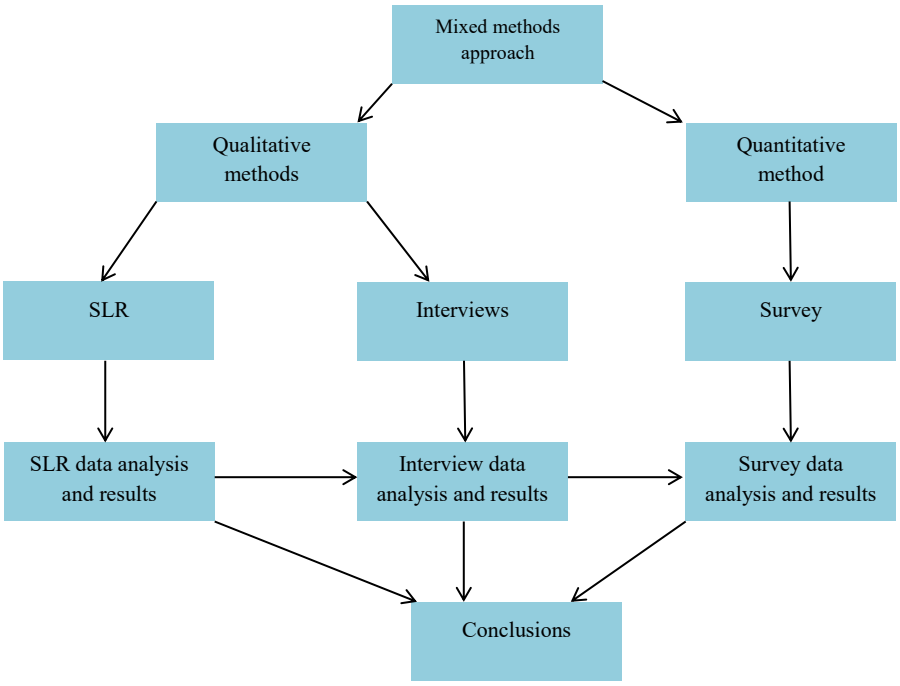

Figure 2.1: The mixed methods approach design

2.4 Qualitative Research Methods: SLR and Interviews

2.4.1 Systematic Literature Review

A systematic literature review (SLR) was conducted to identify the perceived benefits of implementing DevOps in organizations, DevOps adoption challenges, and the vari-

ous DevOps principles and practices. The Kitchenham guidelines [25] were followed to conduct the SLR. Kitchenham et al. [25] defined systematic literature review as "a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest". SLR is mostly used for summarizing all the existing evidence related to a topic, identifying gaps in current research, and to provide a background for planning the new research activities accordingly. Using SLR, all the existing evidence can be summarized in an unbiased and thorough manner [25]. The steps usually followed when conducting a systematic review are defining the topic, identifying search parameters, finding evidence, analysing evidence and integrating evidence [41]. We followed a review protocol for conducting the SLR, which is explained in detail in the next chapter. The motivation behind choosing SLR as the qualitative research method is to gather reliable and valid data. The SLR findings helped us to identify the perceived benefits of implementing DevOps in organizations, adoption challenges, the principles and practices. The SLR results are also validated by the interviews and survey. In this research, we answer RQ1, RQ2 and RQ4 using SLR.

There are other alternate qualitative research methods like systematic mapping study and literature review. Systematic mapping study shares some similarities with the systematic review method, but they have different approaches to data analysis. Also, quality assessment is more common in systematic reviews, whereas there is no need to perform it in mapping studies [36]. Mapping studies are more suited for topics that have broad scope [25]. Finally, the results obtained through systematic reviews will be more reliable and valid because the goal is to identify and aggregate all the relevant evidence. Literature reviews are used for summarizing the evidence on a given topic, and the results are often prone to researcher bias. Also, in literature reviews, no quality assessment or data analysis will be done, unlike in systematic reviews [35]. The data obtained from systematic reviews is less prone to bias and can be more reliable [35] [25]. Thus, we employed SLR as the qualitative research method.

2.4.2 Interviews

Interviews are chosen as the second qualitative research method. Generally, in qualitative interviews, face-to-face or telephonic interviews are conducted [8]. At least one researcher talks to at least one respondent in an interview. Interviews can be either structured or semi-structured [41]. Structured interviews are closed-ended; the researcher frames some questions, and asks them exactly as they are with no deviations. The structured interview data is often statistically analysed. Semi-structured interviews are more open-ended. Open-ended questions which facilitate more interaction are asked, and new questions may be framed as new information is gathered. The semi-structured interview data is often analysed using qualitative analysis methods. Semi-structured interviews are more interactive, and we can get some unexpected responses. Also, higher quality responses can be obtained if the interviewer and the

respondent have rapport [41].

The researcher forms an interview protocol for recording the information gathered. Audio recording, video recording or note-taking can be done for recording the information [8]. In our research, we conducted semi-structured interviews, audiotaped and transcribed them. We employed convenience sampling, a non-probabilistic sampling method for selecting the interview respondents. 12 respondents from 11 different software organizations were interviewed using Skype. We were given permission for audio recording by all the respondents. The respondents are software practitioners experienced with DevOps. Most of the interviews lasted for around 45-60 minutes, while two interviews lasted 120 minutes. Thematic analysis was employed as the data analysis method. The steps involved in data analysis are transcribing the data, organizing the data, reading through all the data, coding the data, identifying themes, and interpreting the meaning of themes [8].

All of the research questions, RQ1, RQ2, RQ3, RQ4 and RQ5 are answered using the interview data. The perceived benefits of adopting DevOps in organizations are identified from the literature. We also ascertain the actual benefits which are achieved in practice by software organizations by interviewing DevOps practitioners. Smeds et al. [42] identified some DevOps adoption impediments by conducting research on one software organization. We aim to identify more challenges, and see if the challenges identified by Smeds et al. [42] can be generalized by including 11 different organizations into our study. We also try to identify the strategies to overcome these challenges by interviewing the practitioners about the measures they use to overcome the adoption challenges. We look into the different DevOps practices that organizations include into their development processes. Also, we identify the challenges that DevOps teams face during continuous integration, deployment and testing from the interviews.

2.5 Quantitative Research Method: Survey

Online survey is the chosen quantitative research method. The results obtained from the qualitative methods, SLR and interviews were used to design the survey questionnaire. The online survey was conducted to validate if the SLR and interview results could be generalized to a larger population sample, and to identify the aspects not covered in the qualitative data. According to Shull et al. [41] surveys are used to "identify the characteristics of a broad population of individuals."

Survey is chosen because respondents from all over the world can answer it. The steps followed to conduct the survey are: defining the survey objectives, planning the survey, designing the survey questionnaire, validating the questionnaire, collecting data, and analysing the data [41]. The survey process is thoroughly explained in the later parts of the document. The questionnaire was designed using google docs, and the collected

data was statistically analysed. The survey questionnaire was validated by piloting it with three DevOps practitioners. Then it was broadcasted to personal contacts, and practitioners identified using LinkedIn and Facebook. The link to survey was posted in several online groups and forums related to DevOps (e.g. DevOps Professionals, DevOps Matters in LinkedIn). Further, the respondents were requested to forward the questionnaire to their personal contacts who are DevOps practitioners. As the survey is prepared using google forms, the responses are stored directly in an excel spread sheet. The target population is software practitioners that have/ are working with DevOps. Convenience sampling was done to select the respondents. 94 respondents were sent the survey personally. 50 complete responses out of 53 were considered for statistical analysis. The three responses were discarded because the two respondents have left a blank (-) in the experience in DevOps field. Another respondent commented 22 years in the experience field. However, DevOps only came into play about six years back. Thus, the responses were not considered. The statistical analysis methods include descriptive statistics and inferential statics [41]. In our study, descriptive statistics was used to analyse the quantitative data.

Creswell [8] notes that the sample for quantitative phase should be different from the qualitative phase sample, as the purpose of exploratory sequential design is to validate if the data from smaller sample can be generalized to a larger population sample. Hence, it was ensured that the survey respondents are different from the interviewees. The alternative quantitative research methods include case study and experiment. Case studies are costly, and often the results from a particular context cannot be generalized to other contexts [41]. Thus, case study was not chosen. To conduct an experiment, we require a hypothesis, certain parameters and variables, which is not the case for our study [41]. Thus, experiment is not chosen. All the research questions, RQ1, RQ2, RQ3, RQ4 and RQ5 are answered using survey, and conclusions are drawn.

Chapter 3

Systematic Literature Review

3.1 Overview of the SLR Process

The SLR was conducted by following the guidelines specified by Kitchenham et al. [25]. Initially, a review protocol was developed for conducting the systematic literature review. The Figure 3.1 provides an overview of the SLR process.

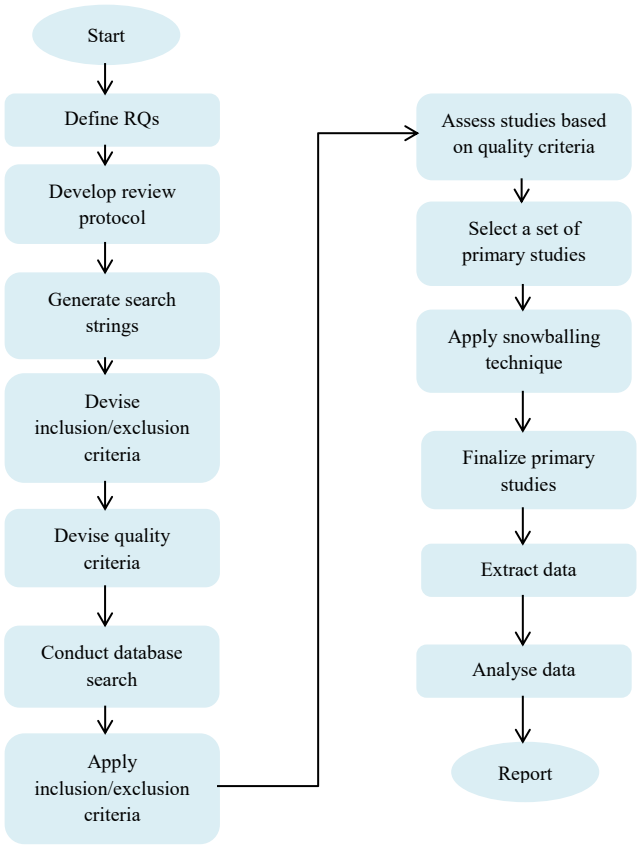

Figure 3.1: The SLR process

The steps we followed to conduct the SLR are identification of keywords, formulation of search strings, devising of inclusion/exclusion criteria, devising of quality criteria, conducting the database search, selection of initial set of primary studies, applying snowballing technique to the primary studies, obtaining a final set of primary studies, data extraction and analysis. The snowballing approach was opted along with database search to ensure no important studies were missed.

3.2 Search Strategy

Initially, a database search was conducted to obtain the primary studies, and then snowballing technique was performed to ensure that all relevant studies were covered. Keywords were identified from the research questions, and search strings were generated by combining these keywords using the Boolean operators "AND" and "OR". Synonyms of the keywords were also added to the search strings to ensure that all relevant studies are obtained.

The keywords identified for this study are: DevOps, Benefits, Challenges, Principles, Practices, Software Development, Development, Operations.

The following scientific databases were chosen for conducting the search of relevant studies:

- ACM Digital Library
- IEEE Xplore
- Inspec
- Scopus
- SpringerLink

The different databases were selected to ensure complete study coverage. Inspec covers most of the articles from IEEE Xplore and ACM databases. Scopus covers most of the articles from ACM, IEEE Xplore and SpringerLink. However, since there is not much literature available on DevOps, all of these databases were searched thoroughly by running search strings individually on each database to ensure no studies are missed out. The search strings used on each database are listed in the Table 3.1.

ACM Digital Library	(+DevOps adoption principles practices methods techniques methodologies processes hindrances challenges impediments benefits advantages software IT organizations development operations)
IEEE Xplore	((DevOps) AND (adoption OR impediments OR challenges OR hindrances OR principles OR practices OR processes OR techniques OR methods OR methodologies OR benefits OR advantages OR software OR IT OR organizations OR development OR operations))
Inspec	(((DevOps) AND (adoption OR impediments OR challenges OR hindrances OR principles OR practices OR processes OR techniques OR methods OR methodologies OR benefits OR advantages OR software OR IT OR organizations OR development OR operations)) WN KY)
Scopus	(TITLE-ABS-KEY(DevOps) AND TITLE-ABS-KEY(adoption OR impediments OR challenges OR hindrances OR principles OR practices OR processes OR techniques OR methods OR methodologies OR benefits OR advantages OR software OR IT OR organizations OR development OR operations))
SpringerLink	DevOps AND "DevOps" AND (adoption OR principles OR practices OR methods OR techniques OR processes OR methodologies OR hindrances OR challenges OR impediments OR benefits OR advantages OR software OR IT OR organizations OR development OR operations)

Table 3.1: Search Strings

3.3 Inclusion/Exclusion Criteria

The following inclusion/exclusion criteria has been defined to remove irrelevant and duplicate studies from the database search results. This criteria is later applied to the search results to obtain a set of primary studies.

3.3.1 Inclusion Criteria

- Studies that are available in English language.
- Studies that are available in full text.

- Conference articles and journal articles.
- Studies published between 2011-2016 (as the research started from 2011 only).
- Peer-reviewed studies.
- Studies that focus on the organizational aspects of DevOps.
- Studies that focus on the benefits, challenges of adopting DevOps in organizations.
- Studies that focus on the principles and practices of DevOps.

3.3.2 Exclusion Criteria

- Studies that are not available in English language.
- Studies that are not available in full text.
- Duplicate studies.
- Grey literature.
- Studies that focus only on cloud computing.

After the database search, a total of 643 studies were obtained from all the selected databases. Of these, 67 studies were obtained from ACM Digital Library, 88 from IEEE Xplore, 81 from Inspec, 163 from Scopus, and 244 from SpringerLink. After applying the inclusion/exclusion criteria, we were left with 27 studies.

3.4 Quality Assessment Criteria

The following quality criteria has been devised to assess the quality of the selected studies. Kitchenham et al. [25] state that quality assessment is critical to determine the strength of the inferences of primary studies. The quality criteria is applied to the finalized set of primary studies obtained after the database search and snowballing procedures.

- Are the aims and objectives of the research clear?
- Does the study report on the benefits of adopting DevOps in software organizations?
- Does the study report on the challenges faced while adopting DevOps in software organizations?
- Does the study describe about the various DevOps principles and practices?
- Is the research methodology clearly specified?
- Are the results of the research clear?
- Are the conclusions of the research clearly stated?

- Are the threats to validity mentioned?

For each study, we check if the above criteria is met by answering each question with either "Yes", "No" or "Partially".

3.5 Snowballing

Snowballing is another way of conducting searches to identify primary studies. In this search approach, new studies are identified by looking into the references and citations of selected studies. The principal reason in choosing snowballing along with the database search is that formulating a search string with the terminology may not provide us with all the relevant studies. Performing a search in the databases with the search string may give a huge number of studies which can be irrelevant, unnecessary and sometimes can be beyond the perspective [54] [25]. The database search requires creating different search strings for different databases, because the databases are confined to particular methods of searching. The changes in search strings may not provide all the relevant literature, and the threat of losing important literature needs to be mitigated. In his research, Wohlin [54] stated with an example that there is a chance of losing the literature while performing the database search. He also explicated that these references can be found by using the snowballing procedure. Hence, snowballing method is considered along with the database search, based on the claims made by Wohlin [54] in this research.

Using the references of selected studies to find new studies is called backward snowballing, while using the citations of selected studies to find new studies is called forward snowballing [2]. The backward and forward snowballing should be carried out in iterations. The newly identified studies should be added to the selected start set, and snowballing of the new papers should be done again in the next iterations, until no papers are found [2]. In our study, we conducted backward and forward snowballing to the set of 27 studies obtained after applying the inclusion/exclusion criteria to the database search results.

Backward snowballing was conducted in one step, and forward snowballing was conducted in one step. Seven new studies were identified after using the snowballing technique. The citations of studies were identified with the help of Google scholar. Of the seven obtained studies, four were selected as primary studies after applying the inclusion/exclusion criteria. All the new studies were obtained in a single iteration because their references were either grey literature or matched our specified exclusion criteria, and their citations were also found to be irrelevant after applying the inclusion/exclusion criteria. The Figure 3.2 describes the selection of primary studies after database search and snowballing.

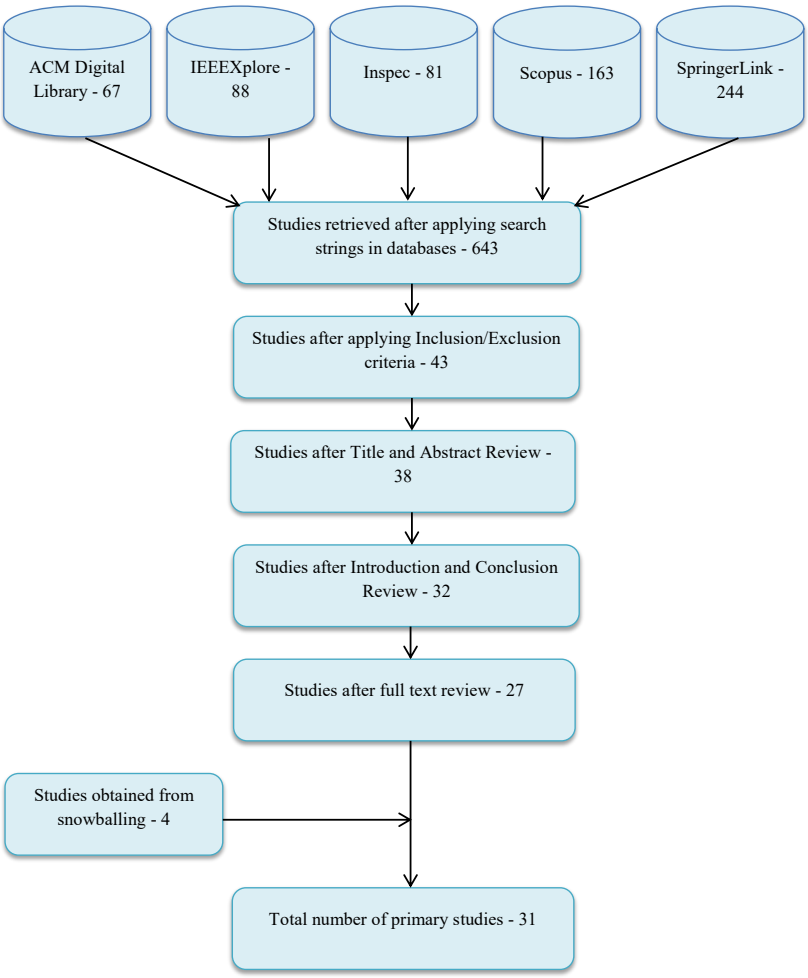

Figure 3.2: Selection of Primary Studies

3.6 Data Extraction and Synthesis

31 primary studies were selected for conducting the systematic literature review. A data extraction form was designed according to the guidelines specified by Kitchenham et al. [25] to record the data obtained from the primary studies. The following form was designed to extract the data relevant to RQ1, RQ2 and RQ4.

Article Title	Specify Name
Author(s)	Specify name(s)
Type of Article	Specify type
Year of Publication	Specify year
Research Method	Specify if mentioned
Research Questions	Specify if mentioned
Does the study specify the benefits of adopting DevOps	Yes/ No. Explain if Yes
Does the study specify the challenges faced while adopting DevOps	Yes/ No. Explain if Yes
Does the study specify the practices of DevOps	Yes/ No. Explain if Yes
Study Conclusions	Specify conclusions

Table 3.2: Data Extraction Form

3.7 SLR Results

31 primary studies were identified for this research. The primary studies and their identifiers are listed in the appendix A. After conducting the database search and snowballing, 31 primary studies relevant to the study were found. All the studies were published between the years 2011 to 2016, while one study [S29] was published in 2008. The year wise distribution of primary studies is presented in the following figure.

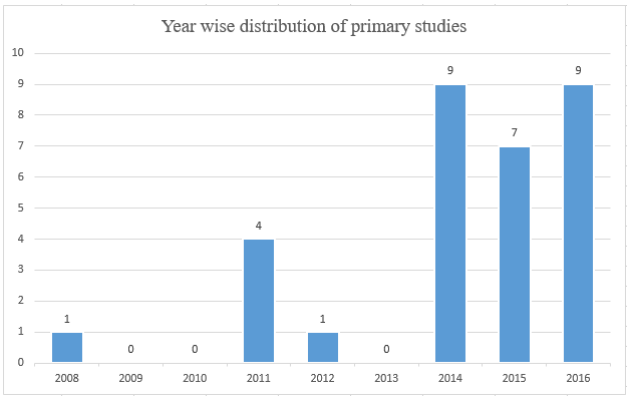

Figure 3.3: Year wise Distribution of Primary Studies

Four studies were published in 2011, and one study was published in 2012. It can be seen that the research on DevOps has increased since 2014. Nine studies were published in 2014, seven in 2015 and nine in 2016. Among the 31 studies, 22 studies report on the benefits of adopting DevOps in organizations, 7 studies report on the DevOps adoption challenges, and 10 studies address the principles and practices of DevOps. A detailed analysis of the literature is provided in the next sections. Thematic analysis was followed to analyse the extracted data. The extracted data was color coded, and the identified codes were categorized into the themes "Benefits", "Challenges" and

"Practices". A detailed analysis of the literature is presented in the next subsections.

3.7.1 Analysis of Literature Regarding the Benefits of Adopting DevOps in Organizations

In this section, we analyse the primary studies regarding the benefits of adopting DevOps in organizations. One objective of this study is to identify how implementing DevOps in organizations where agile development is in practice helps them. The perceived benefits of DevOps adoption are listed by the studies [S1] [S2] [S3] [S4] [S5] [S6] [S7] [S8] [S9] [S10] [S11] [S12] [S13] [S14] [S16] [S17] [S18] [S19] [S20] [S29] [S30] [S31].

It is to be noted that most of the identified benefits are the perceived benefits of adopting DevOps in organizations. Many companies have started to adopt DevOps these days in an attempt to leverage the benefits [22]. Stilwell and Coutinho [S2] reported on the benefits achieved after implementing DevOps in a research project. They observed a qualitative difference in the way teams operate after implementing a DevOps work flow. They found that teams could meet the project milestones with more speed and efficiency. Virmani [S9] reported on a case study conducted at IBM, and he has identified several benefits of adopting DevOps in organizations. Callanan and Spillane [S14] reported that after implementing DevOps in their company Wotif, there was a significant improvement in the average release cycle time. Erich et al. [S3] conducted a mapping study on the cooperation between the development and operations teams in organizations, and they found that DevOps has a positive effect on the performance of both development and operations teams, and also quality assurance teams. They also found that DevOps has a positive impact on web service development. Nybom et al. [S20] conducted interviews in an organization and identified the benefits of shared responsibilities between Dev and Ops. They found that DevOps increases the trust and collaboration, and improves the work flow. Tessem and Iden [S29] identified that good cooperation between development and operations teams is essential for successful software deployment and operation, way before the term DevOps was coined.

The following benefits have been identified from the primary studies:

- The communication, collaboration and trust between development and operations team members increases [S1] [S9] [S10] [S20].
- Improved quality of software deployment [S1] [S17].
- Responsiveness to business needs increases [S1].
- The software reliability increases [S1].
- The code quality will be improved [S1].
- Implementing the customer requirements throughout the software development process becomes easier [S1].

- Development teams will be able to operate autonomously, while being more open to changes and providing frequent updates [S2].
- The communication overhead for making coordinated changes reduces [S2].
- DevOps is supported by structures and standards, and it also facilitates the realization of the structures and standards which are advantageous for development and operations [S3].
- Frequent software releases [S1] [S4] [S6] [S30] [S31].
- Overcomes the gap between development and operations teams [S4] [S5] [S6] [S7].
- It solves important issues like fear of change and risky deployments [S8].
- The time to market is reduced [S9] [S31].
- Continuous feedback will be enabled [S9].
- Cost and quality will be balanced out [S9].
- Increased predictability in releases [S9].
- The overall organization's efficiency will be increased [S9].
- The cycle times of operations activities such as deployment and change controls will be reduced [S10].
- Continuous improvement will be encouraged as a means to automation of tools and processes, and also for adapting to the quick changes in the software environment [S10].
- Increases revenue of the organizations [S11].
- Reduces costs by reducing the need for rework [S11].
- Problems can be detected earlier, and can be solved efficiently and effectively [S12].
- Reduced time to software release [S13].
- Reduces the average release cycle time from weeks to hours [S14].
- Rapid delivery of products [S16].
- Increased customer satisfaction [S30].
- DevOps quantifies the aspects of the development process, and the focus on metrics subsequently improves the product development [S18].
- DevOps helps to deliver higher quality services in a more efficient way [S19].
- Smooth and faster work flow [S20].

3.7.2 Analysis of Literature Regarding the Challenges of Adopting DevOps in Organizations

In this section we analyse the primary studies regarding the challenges faced while adopting DevOps in organizations. Identifying the challenges to DevOps adoption in organizations is another aim of this research. The possible challenges to DevOps adoption are identified by the studies [S1] [S20] [S21] [S22] [S23] [S24] [S25].

Only few studies address the challenges to DevOps adoption in organizations. Several hindrances to the DevOps adoption process in the cloud services area have been identified by Smeds et al. [S21]. They conducted interviews in an IT organization with over 1000 employees, and identified 11 challenges in the initial stage of DevOps adoption process. Jones et al. [S24] conducted a case study in a SME (Small and medium-sized enterprise) having over 200 employees and identified the challenges to DevOps adoption within that organization. At that time, the organization had a legacy system, and a new system was being developed as a replacement for the legacy system. Lwakatare et al. [S25] identified the challenges to adoption of DevOps in the embedded systems domain by using a multi-case study approach with four companies.

The challenges that occur while adopting DevOps in organizations are:

- A lack of proper management structure in an organization can hinder the adoption process [S1] [S24].
- There is no proper training in place for DevOps in several organizations, so the DevOps engineers are usually hired based on their experience as system administrators and software developers [S1].
- Unclear definition and goals of DevOps adoption [S21].
- Organizational structure may impact the DevOps adoption process [S21].
- In some cases, the customer might fancy a different approach of development, such as including different practices and process with strict procedures on deployment. DevOps might not be suitable for product development in these cases [S21].
- Geographical distribution of the development and operations teams can hinder the DevOps adoption [S21].
- People may perceive DevOps as a buzzword, and this negative perception may cause them to try to resist its adoption [S21].
- Developers may be afraid that DevOps will overburden them with operations responsibilities, and reduce their working efficiency [S21].
- Because DevOps requires all personnel to have both development and operations skills, people may not be open to the change due to fear of not having in-depth expertise in both areas [S21] [S20].

- People may be interested in only their area of expertise, not in what other teams do [S21].
- Monolithic system architecture can be hindering to the various DevOps practices like continuous builds, testing and deployment [S21].
- Differences between the development, testing and production environments can complicate the collaboration and also continuous delivery and deployment [S21].
- Multiple production environments can cause complexity, and hinder the use of common tools and processes, which can affect continuous delivery [S21].
- Moving to DevOps is more difficult with legacy software [S22] [S24].
- Specifying the software architectures to make them work in DevOps scenarios can be challenging for software architects [S23].
- Lack of buy-in from senior management can be challenging [S24].
- Resistance from employees can also hinder the adoption process [S24].
- Hardware dependency and compatibility with different software versions [S25].
- Limited visibility of customer production environments when configuring test environments [S25].
- Lack of automation tools for deployment of new features in the embedded systems domain [S25].
- If the system performance data collected by companies does not contain feature usage data, it hinders DevOps adoption [S25].

Several challenges were identified by Smeds et al. [S21] after interviewing software practitioners in a single software organization. They note that if the teams do not have a clear view on the definition of DevOps, they might misunderstand the goals and the actions needed to achieve these goals. They identified organizational structure as another challenge. Another addressed challenge was that the DevOps processes and practices might not be compatible with the processes and practices required by the customer. Geographical distribution of development and operations teams was identified as another challenge. If the teams are in different time zones, it can lead to communication issues and process related issues. The perception of DevOps as a buzzword by the personnel was identified as another challenge.

Smeds et al. [S21] note that because of the fuzzy definition of DevOps, many personnel lacked trust in the concept, and there was an opinion that the activities termed under DevOps were not that different from the activities practised before the term was coined. They state that this perception could lead to resisting change. Another challenge identified was that developers were concerned that their workload might increase due to DevOps, and the new responsibilities might affect their focus and productivity. Lack of interest in what the other teams do was identified as a challenge. It was noted

that several personnel were interested only in their area of expertise.

Smeds et al. [S21] identified monolithic architecture as a huge challenge. They state that system architecture affects the way the system is developed, tested and deployed, and monolithic architecture can hinder the continuous builds, testing and deployment. Another addressed challenge was the difference between development, testing and production environments. If the development and testing environments do not reflect the production environments, there is a risk of the software not being properly validated before it is deployed to production. This can complicate not only continuous delivery and deployment, but also the collaboration. The developers are familiar with development and testing environments, but not production environments. Thus, because of the differences between environments, the use of production environments will be challenging to the developers, which affects the collaboration. Having multiple production environments was identified as another challenge. They state that the differences between multiple production environments hinder continuous delivery. It was noted that use of different environments causes complexity, and also hinders automation and use of common tools and processes.

In their case study, Jones et al. [S24] identified that lack of proper management structure can hinder the adoption process. In the organization on which they conducted their research, the adoption of DevOps was a bottom up process, led by a software development manager. The software development manager had to convince the senior manager on the benefits of DevOps. Further, there was resistance from the operations teams in that organization because they had the opinion that coding was a developer responsibility. They also found that maintenance of legacy systems is a major issue faced by the developers, and it also affects learning of new technologies. In this case study, the developers were developing a new system, and they expressed the opinion that maintenance of legacy system interrupted the new system development. Thus, Jones et al. identified the DevOps adoption challenges as lack of proper management structure, lack of buy-in from senior management, legacy systems and resistance from employees.

Lwakatare et al. [S25] identified the challenges to DevOps adoption by conducting a multi-case study in four software companies that developed embedded systems. The first challenge is hardware dependency and compatibility with different versions of software. They identified that because of hardware dependencies, the companies had silos of development teams for different modules, which resulted in longer development cycles. This led to delays in release of new features to customers. It was also identified that the companies had limited visibility of the customer production environments, which complicated the configuration of testing environments. DevOps stresses on testing features in a production-like environment, and the limited visibility of production environments in embedded systems hinders the adoption process. Another challenge identified was the lack of automation tools to continuously deploy new fea-

tures in the embedded systems domain. It was noted that the companies collected the system performance data, but feature usage data was missing in it. The feature usage data helps to continuously improve products. Thus, lack of feature usage data was identified as a challenge to DevOps adoption.

3.7.3 Analysis of Literature Regarding the Practices of DevOps

In this section, we analyse the literature regarding the various practices of DevOps. Identifying the practices to be included by the organization as a part of the adoption process is another aim of this research. The different practices of DevOps are listed by the studies [S9] [S15] [S18] [S21] [S22] [S25] [S26] [S27] [S28] [S30].

For an organization, it takes a lot of effort to process the adoption of a different methodology. They need to change their infrastructure, the organizational structure, the development strategy, and continuous involvement of operation team. The organization must include some practices into their development processes as a part of the adoption process. Adoption of DevOps takes a great amount of collaboration with in the organization as it requires revamping of that organization. The focal part in DevOps adoption is that the integration between software development and its operation deployment should happen continuously [16]. In order to embrace this change, the organizations must follow some principles and practices.

The adoption of DevOps takes down the wall in between the development and the IT operations by making them work towards achieving a common goal throughout the process [30]. DevOps is a process that includes the principles and practices to synergize the development and operations departments in the organization [12]. The organization should enable the communication and coordination in between the development and operations departments as it leads to better performance of both the teams [19]. As the continuous work in between the DevOps teams takes place, the organization should get accustomed to some activities such as continuous planning, continuous integration, continuous testing, continuous deployment which supports continuous innovation [17].

The principles and practices in DevOps are related to the aspects of Culture, automation, measurement, and sharing [16].

Culture: DevOps is more of a merger of development and operations. So, both the departments hold the joint responsibility of delivering high quality software to the end user. Consequently, the code can no longer be “thrown over the wall”.

Automation: To achieve quicker delivery, feedback from end users and short lead time, DevOps counts on complete automation of the build, deployment and testing.

Measurement: Comprehending the capability of current delivery and setting goals for

its improvement can only be done by measuring. This measurement varies from monitoring business metrics to test coverage and deployment time of a new software version.

Sharing: Sharing of knowledge, tools, culture, infrastructure and success as well can make the development and operations team closely acquainted.

Smeds et al. [S21] defined DevOps as a "set of engineering process capabilities supported by certain cultural and technological enablers." They have identified the enablers as automation, sharing, and communication. The capabilities, in other words, are the practices that an organization should include [S21]. Thus, the organization needs to follow the principles of automation, sharing and communication in order to support the technical practices.

The following practices of DevOps have been identified from the literature [S9] [S15] [S18] [S21] [S22] [S25] [S26] [S27] [S28] [S30].:

- Continuous Planning
- Continuous Integration
- Continuous Deployment
- Continuous Delivery
- Continuous Testing
- Continuous Monitoring and Feedback

Continuous Planning:
To aid the development and operations teams to adjust to the rapid changes in business environments, there is a need to have agile business plans in place. DevOps enables this by having a product backlog which can be prioritized all the time, and a continuous channel for feedback from customers. It is important that planning is done frequently to cope with quick changes in business environments, and there needs to be tight integration between planning and execution. The continuous planning is a cyclic process, and achieved by planning small batches, executing, obtaining feedback, responding to the feedback and adjusting the plan if needed.

Continuous Integration:
Continuous integration necessitates a connection between development and operations teams. To achieve this, integration must be done early, changes should be shared with teams without keeping them localized for long, and code should be validated continuously. Continuous integration requires some form of automation, and has interconnected steps like compiling code, running unit and acceptance steps, validating code coverage and building deployment packages. Automation must be done in such way that as soon as a developer makes a change, it is detected by the build system and a build is triggered, sanity tests are carried out, and the build is posted to a repository.

This needs to be a repeatable and continuous process. It is important that there are highly visible artefacts in case of continuous integration failures so that the problems which caused the failure are fixed as early as possible by the responsible personnel.

Continuous Deployment:
Continuous deployment involves automatically deploying the software to some environment, but not necessarily to the customers. Continuous deployment is a prerequisite for achieving continuous delivery. Virmani [47] states that continuous deployment is a critical step in overall software delivery optimization. DevOps principles recommend automation of deployment and provisioning of hardware. Manually configuring the hardware to test builds can take up to several weeks of time, and the deployment processes will not be consistent. Another principle of DevOps comes into play in continuous deployment; Infrastructure as code (IaC), which implies that infrastructure is maintained as code in the source code repository.

Continuous Delivery:
Continuous delivery involves automatically releasing valid software builds to the customers. Continuous deployment is necessary for achieving continuous delivery, but the reverse is not usually the case. Continuous delivery is the ability to deliver software whenever the organization wants to. In continuous delivery, as soon as new features are finished, they are deployed into production.

Continuous Testing:
Automation of all test cases is necessary to achieve continuous testing. Existing manual test cases should be automated, and the test cases should be executed on every generated software build without any manual intervention. Continuous testing aims at eliminating root causes and involves automation of testing cases so that the time between introduction and detection of errors is reduced.

Continuous Monitoring and Feedback:
Continuous monitoring of infrastructure and user-behavior provides feedback loops for the planning and development processes to improve and optimize the service. Different quality parameters can be observed, and we can react to any surprises in a timely manner. In DevOps, feedback is the data collected from operating the service as an input in planning and development. This data contains information about the performance of infrastructure, and how and when the users interact with the service.

3.8 Discussion and Conclusion

Among the 31 primary studies identified through database search and snowballing, 22 studies address the benefits of adopting DevOps in organizations, 7 studies discuss about the DevOps adoption challenges, and 10 studies discuss about the principles and

practices of DevOps. There is currently very little high-quality literature related to DevOps as it is a very new topic, but a slight growth in the number of studies in this domain since the past two years has been observed. Many authors have stressed on the need to conduct further and extensive research on DevOps. The main objectives of this study are to identify the benefits of adopting DevOps, the challenges faced by organizations while adopting DevOps, and the different practices that organizations must include for adopting DevOps. By conducting the SLR, we were able to answer three of our research questions, RQ1, RQ2 and RQ4.

Our research aims at identifying the benefits of implementing DevOps in organizations where agile development is in practice. However, the perceived benefits of adopting DevOps in organizations in general were identified from the literature. Several studies mentioned the perceived benefits of adopting DevOps. It was observed that many organizations have started to adopt DevOps because of the perceived benefits like increased frequency of software releases, improved quality of software deployments, reduced time to market, increased customer satisfaction, etc [47] [22]. Stilwell and Coutinho [S2] noted a qualitative difference in the way teams operate, and observed that the teams were able to meet project milestones with more speed and efficiency after implementing DevOps in a research project. Callanan and Spillane [S14] reported a significant reduction in the average release cycle time after adopting DevOps in their company. Nybom et al. [S20] noted that DevOps increases trust and collaboration between teams, and improves the workflow. Several other perceived benefits of adopting DevOps were identified from the literature. The benefits identified from the literature will be ascertained after interviewing DevOps practitioners.

While the popularity of DevOps is attracting many companies to adopt it, it has been observed that adoption of DevOps is not easy. Despite the perceived benefits, organizations may face several challenges while adopting DevOps. Only 7 studies report on the DevOps adoption challenges. Smeds et al. [S21] identified several DevOps adoption challenges after conducting interviews in an IT organization. Jones et al. [S24] identified the challenges to DevOps adoption after conducting a case study in a SME. The challenges to DevOps adoption in the embedded systems domain were identified by Lwakatare et al. [S25] after conducting a multi-case study in four companies. The mostly stated challenges were identified as lack of a proper management structure, resistance from employees, fear of change, and legacy software. Several other generic challenges like lack of training related to DevOps, unclear definition and goals of DevOps adoption, geographical distribution of Dev and Ops teams, monolithic system architecture, lack of proper automation tools, etc. were identified from the literature. The DevOps adoption challenges identified from the literature will be further ascertained after conducting interviews.

Another aim of the study is to identify the practices that an organization must include while adopting DevOps. It has been observed that organizations need to adopt certain

principles, and include certain technical practices as part of the DevOps adoption process. 10 studies address the principles and practices of DevOps, but it was seen that there is no clear definition of the various DevOps principles and practices in the literature. The principles of DevOps were observed to be culture, communication, sharing and automation. As specified by Smeds et al. [S21], the principles are the enablers of the technical practices. To include the technical practices, organizations must mainly follow the principles of culture, automation, sharing and communication. The technical practices were identified to be continuous planning, continuous integration, continuous deployment, continuous delivery, continuous testing, continuous monitoring and feedback. The principles are needed to support the technical practices of DevOps. The identified DevOps practices will be validated, and explained in detail after interviewing the DevOps practitioners.

3.9 Threats to Validity

While performing the SLR, there are some validity threats that could occur. These threats need to be addressed in order to make the data more definite, and because disregarding these threats can make the interpretation go wrong. These threats are identified and are narrowed down by using the validity categories provided by Wohlin et al [55]. They are:

1) Internal Validity:
The authors have performed snowballing along with database search to acquire maximum literature in the research area. While performing this search, internal validity can be the one of the major threats for this study. This threat can be mitigated by performing some procedures such as formulating a search string. The search string was formulated with the help of librarian, and verified by the supervisor. With the help of search string, a database search was conducted and the literature obtained from the search was gathered. This literature was refined by applying inclusion and exclusion criteria, where the articles which are irrelevant to the research area were excluded. For example, the studies related to cloud were not considered. In addition to database search, snowballing was performed to not skip any important studies relevant to our work. The snowballing process was carried out in iterations. After performing the above mentioned steps, quality criteria was defined to assess the studies obtained from the database search and snowballing procedures. Thus the threat of publication bias is reduced. The finalized list of primary studies was taken into consideration after the quality assessment. The whole process was carried under the observation of the supervisor. The supervision aided us as the process was checked again, and the steps performed were also verified by the supervisor.

2) Construct Validity:
Construct validity aims at possible confounding factors, whether the study has the ca-

pability of focusing on its aim and objectives. These threats were reduced by planning the sections accordingly based on the research questions and delineating the sections. Another threat is the search string, as the relevant studies should be searched for thoroughly, or there is a risk of missing the literature. This was eliminated by formulating the search string with the help of librarian, and getting it verified by the supervisor. Another threat which needs to be considered is, while performing snowballing there is a possibility of finding the same author, or same studies. To reduce this risk, those studies were excluded.

The other threats that could happen are that there is a chance of missing significant information from the literature. While focusing on documenting, the authors might skip the information in the literature, which may lead to missing the important information for the research. To mitigate this threat, the authors have maintained an excel sheet where every important data related to the primary study were listed out separately, which avoids the uncertainty in finding the information when required. The data extraction forms aided the authors in listing out the information.

The other possible threat that might affect the literature is the domain applicability. Since there is less literature available in the research area, the search string may provide the articles related to cloud computing, health care, agile methodologies. As the study is completely focused on the adoption of DevOps in organizations, those studies should be excluded. To mitigate this threat, the search string was framed by using the keywords related to the domain, and quality assessment was performed so that the studies which have irrelevant literature and improper data are excluded. The quality assessment helped the authors in finding quality literature and contributed well to the documentation of the research.

Chapter 4

Analysis of Interview Data

In this chapter, the data analysis and results of the interviews conducted are described.

The target population for interviews is software engineering practitioners that have worked with DevOps. A non-probabilistic sampling method, convenience sampling was employed for selecting the interview respondents. The interviewees were selected from personal contacts, and practitioners on LinkedIn. Convenience sampling refers to gathering responses from people who are available and willing to respond [41]. The motivation behind choosing non-probabilistic sampling is because the target population is specific and limited. DevOps is a relatively new phenomenon, and organizations are only recently becoming familiar with it. It is thus hard to rely on a random sample.

A semi-structured interview questionnaire having both open-ended and close-ended questions was initially designed based on the SLR findings, and improvised according to the feedback from the supervisor. Questions related to interviewee demographics and the present study were included. The questions were related to the benefits of adopting DevOps in organizations, challenges faced during the adoption process, strategies to overcome these challenges, the different DevOps practices, and the problems that DevOps teams face during continuous integration, deployment and testing.

A total of 12 interviews were conducted with 12 different software practitioners from 11 different organizations. All the 12 interviewees were asked the same set of questions, and spontaneous questions were asked whenever needed. The interviews were conducted using Skype. The interviewees were promised confidentiality to get them to give genuine and detailed answers. Audio recording was done in all interviews to collect the data. All the interviewees gave permission for audio recording. Thematic analysis was done to analyse the collected data. The following steps are followed to analyse the interview data:

- Transcribing data
- Organizing data
- Reading through all the data
- Coding the data
- Identifying themes
- Interpreting the meaning of themes

Transcribing refers to writing down the recorded audio data. All the recorded data from each interview was transcribed into separate word documents. Everything spoken by the interviewees was transcribed manually, without leaving any word in order to avoid bias. The transcripts were organized by assigning individual IDs for each transcribed interview document, to avoid confusion and facilitate easier data analysis. All transcripts were thoroughly read to familiarize with the data, and get a clear understanding. Codes in each transcript were identified by manually examining the transcripts. Each code was highlighted with a different colour, based on the question. The identified codes were classified into themes based on their similarities and relevance. The authors have defined the code inclusion criteria as frequency and relevance. In our research context, frequency refers to the responses specified by more than one respondent, and relevance refers to both relevance to the literature, and the relevance to our research.

4.1 Demographics of Interviewees

The demographics of the interviewees are presented in this section. To preserve the anonymity of the interviewees, they are assigned individual IDs, and are referred to as Interviewee 1, 2, etc. Among the 12 interviewees, six are from India, four are from Sweden, and two from USA. As most of the respondents are not in Sweden, face-to-face interviews were not possible. Also, the interviewees from Sweden were not located in the same part of Sweden as the researchers. Thus, Skype interviews were conducted. Most of the interviews lasted for 45-60 minutes. The interviewees 2 and 9 are from the same organization. The roles of the interviewees, size of their organizations, and the experience of the interviewees with DevOps is presented in the Table 4.1.

Interviewee ID	Location	Role	Organization Size	Experience
I1	India	Developer	> 5000	1 year
I2	India	Build/ Release Engineer	> 3000	3 years
I3	USA	DevOps Consultant	2500	3 years
I4	India	Build/ Release Engineer	> 5000	1.5 years
I5	India	System Administrator	50	7 months
I6	Sweden	Professional Services Engineer	50	1 year
I7	USA	Senior Specialist	> 5000	1 year
I8	Sweden	Senior Software Engineer	50	2 years
I9	India	Operations	> 3000	1.5 years
I10	Sweden	Software Developer	50	6 months
I11	Sweden	DevOps Consultant	10	4 years
I12	India	Build/ DevOps Engineer	> 5000	1 year

Table 4.1: Demographics of Interviewees

4.2 Codes and Themes for RQ1

A total of 20 codes for the benefits of adopting DevOps in organizations where agile development is in practice were identified from the 12 interview transcripts. The codes were selected based on the inclusion criteria relevance to study, and frequency of occurrence. The identified codes were categorized into the theme "Benefits". The Table 4.2 presents the codes and themes identified for RQ1.

Codes	Code Inclusion Criteria	Themes
Faster feedback	Frequency, Relevance	Benefits
Reduced release cycle time	Frequency, Relevance	Benefits
Reduced gap between Dev and Ops teams	Frequency, Relevance	Benefits
Rapid Delivery	Frequency, Relevance	Benefits
Automation	Frequency, Relevance	Benefits
Increased empathy between Dev and Ops teams	Relevance	Benefits
Increased communication between Dev and Ops teams	Frequency, Relevance	Benefits
Rapid time to market	Frequency, Relevance	Benefits
Easier tracking and collaboration	Relevance	Benefits
Breaks down silos	Frequency, Relevance	Benefits
Improved code quality	Frequency, Relevance	Benefits
Increased system stability	Relevance	Benefits
Minimizes manual intervention	Frequency, Relevance	Benefits
Saves time and effort	Frequency, Relevance	Benefits
Reduced human error	Relevance	Benefits
Increased agility	Relevance	Benefits
Implementing business requirements in a faster way	Relevance	Benefits
Earlier detection of failures	Relevance	Benefits
Reduces cost	Frequency, Relevance	Benefits
Faster workflow	Frequency, Relevance	Benefits

Table 4.2: Codes and Themes for RQ1

4.2.1 Analysis of Interview Responses for RQ1

In this section, we analyse the codes related to the benefits of adopting DevOps in organizations where agile development is in practice. 20 codes were identified from the 12 interviews, and grouped into the theme "Benefits".

Benefits: All the interviewees discussed about the advantages of implementing DevOps where agile development is in practice. It is to be noted that agile methodologies were followed in organizations of all the interviewees before adopting DevOps. Three interviewees 1, 2, 9 expressed that DevOps enables faster feedback. Interviewee 2 said,

"Feedback is one of the best practices of DevOps." Interviewee 9 said, "Previously, we used to release only after the entire product is complete. Now (after adopting DevOps), we deliver small Minimum Viable Products (MVPs). We release each MVP and the customers give us feedback. Based on that feedback, we improve the product, which will incrementally improve the value."

Reduced release cycle time was identified as another benefit from interviews 1 and 12. Interviewee 1 said, "Previously, the release cycle was two weeks. After adopting DevOps, we were able to bring down the release cycle to three days. One of the reasons to adopt DevOps is to expedite the release process." Reduced gap between Dev and Ops teams was identified from 1, 12 interviews. Rapid delivery was identified from interviews 2, 4, 7, 8, 12. Interviewee 2 said, "With DevOps, builds, deployments and testing will be done in an automated fashion. Because of this, the time taken will be reduced from 6 hours to at the most, 30-35 minutes." Interviewee 4 said that speed of delivery was the ultimate achievement in their organization. Interviewee 10 stated that the delivery speed in their organization increased by 30-40 percent after adopting DevOps. Interviewee 12 commented that because of automation, the delivery speed increased in their organization.

It has been noted that many interviewees perceived automation in DevOps as a benefit. Interviewee 2 said, "Another good thing about DevOps is automation. There is no manual intervention in any of the stages." Interviewee 6 said, "Automation saves time and gives quick results. Previously, we had to maintain test beds, test servers, systems and so on. For that, one system takes around 45 minutes. Once we automated these processes, it hardly takes about 5 minutes for everything." Interviewee 9 said, "Minimum human intervention and maximum automation yields better results in terms of delivery speed and quality." Interviewee 9 also noted that some people argue that automation causes unknown side effects. He said that some people may have a misconception that automation does all the work, and thus overlook things which need to be checked rigorously. Interviewee 12 commented that because of automation, human intervention would be reduced, and delivery speed would increase.

Increased empathy between Dev and Ops teams was identified as another benefit. Interviewee 11 commented "What does it really need for the developer to understand the situation of the tester or an operations person? Empathy. So, one of the key ingredients for DevOps is empathy towards other human beings that you have to work with." Increased communication between Dev and Ops teams was identified from interviews 2 and 3. Rapid time to market was identified as a benefit by interviewees 8 and 9. Easier tracking and collaboration was identified as a benefit. Interviewee 4 commented that "Earlier, the developer used to handover code to the repository, and builds were made, then QA (quality assurance) team used to test it and push it to production, and so on. But now the control is under everyone's hand, and it is easy to track and collaborate and work to get the job done." Another benefit identified was that DevOps breaks down si-

los. Interviewee 2 commented that "Before DevOps, development and operations were separate silos. They did not collaborate. Now they are a single team." Interviewee 4 commented that "DevOps broke the silos. It is about collaboration between teams and ensuring that the job gets done."

Improved code quality and increased system stability were identified from interview 4. Interviewee 4 stated that the code quality is high as it gets checked at every batch set, and the system is more stable. Interviewee 12 said, "There will not be any walls between the teams, and everyone can see what is going on. Because of this visibility, collaboration will be enhanced and problems can be easily solved. This will improve the code quality." Interviewee 5 commented that DevOps minimizes manual intervention. He pointed out that as a result, human error can be avoided. He said that "Human error could mess up the environments." Another advantage identified from interview 5 was that DevOps saves time and effort. Increased agility in the organization was identified as an advantage from interview 6. Interviewee 9 commented that using DevOps, business requirements can be implemented in a faster way. He also added that DevOps enables earlier detection of failures. Interviewees 10 and 12 stated that DevOps reduces the delivery time and development cost. Interviewees 4, 11 and 12 commented that DevOps results in a faster workflow.

4.3 Codes and Themes for RQ2

A total of 20 codes for the challenges faced by organizations while adopting DevOps were identified from the 12 interview transcripts. The codes were selected based on the inclusion criteria relevance to study, and frequency of occurrence. The identified codes were categorized into the theme "Challenges". The Table 4.3 presents the codes and themes identified for RQ2.

4.3.1 Analysis of Interview Responses for RQ2

In this section, we analyse the codes related to the challenges faced while adopting DevOps in organizations. 20 codes were identified from the 12 interviews, and grouped into the theme "Challenges".

Challenges: All the 12 interviewees agreed that DevOps adoption can be challenging for organizations that are implementing it for the first time. Changing architecture to support DevOps was stated as a major challenge by interviewees 2 and 9. Interviewee 2 commented by saying, "In our organization, we have huge applications, and the system architecture is complex. When you adopt DevOps, you need to convert your systems into micro services or a modularized architecture. The advantage with modularized architecture is that if there is any change in the code, only that corresponding module

Codes	Code Inclusion Criteria	Themes
Changing architecture to support DevOps	Frequency, Relevance	Challenges
Insufficient knowledge on applications and environments	Relevance	Challenges
Culture and mindset of long timers	Frequency, Relevance	Challenges
Resistance from employees	Frequency, Relevance	Challenges
Training	Frequency, Relevance	Challenges
Security clearances	Relevance	Challenges
Technical debt	Relevance	Challenges
Lack of collocated teams	Relevance	Challenges
Making everyone accustomed to a new workflow	Relevance	Challenges
Tool adoption	Frequency, Relevance	Challenges
Learning curve	Frequency	Challenges
Buy-in from management	Frequency, Relevance	Challenges
Writing custom scripts for automating tasks	Relevance	Challenges
Over-burdened personnel	Relevance	Challenges
Hidden complexities and hidden costs	Relevance	Challenges
Fear of measurement in employees	Relevance	Challenges
Integrating different tools	Frequency, Relevance	Challenges
Migrating to cloud	Frequency, Relevance	Challenges
Lack of proper resources to configure environments	Relevance	Challenges
Fear of failure in managers	Relevance	Challenges

Table 4.3: Codes and Themes for RQ2

needs to be built. There is no need to build the entire code and risk the schedule. However, to support this conversion, the entire system architecture has to be changed. For this, the entire code base needs to be re-written. This is the main challenge that we faced." Interviewee 9 noted that architecting for DevOps is challenging.

Insufficient knowledge on applications and environments was identified as a challenge to DevOps adoption by interviewee 3. He commented that this applies for outsourcing companies. He added that in outsourcing companies, instead of hiring a full time developer, they employ a consulting company that will bring consultants on demand. That means they will hire a consultant only when there is work to be done. "That way, I don't have to pay a full time salary, or provide any benefits." He stated that there is no guarantee that the same consultant comes in for all projects, and whoever is available on the bench will be brought in by the consulting companies. "For DevOps to happen, you need to have an intricate knowledge on the applications or environments have been set up. Without that, when someone new comes in they have to learn about them, which takes time. This is not easy when new people are coming in to develop all the time."

Culture and mindset of long timers was identified as another challenge by interviewee 3. He added, "Many organizations will not have the luxury of young development team. They have a lot of people with a long tenure. The culture and mind set of these long timers will be a biggest challenge." He cited two other examples which were hindered by mindset of long timers; Disruption from main frames to open systems, and Changing from waterfall to agile methodologies. Interviewee 7 stated that adopting to the DevOps culture change is challenging. Interviewee 5 also stressed on the importance of having a DevOps mindset. He commented that by mindset, he means how much people are motivated to try a new strategy. Interviewee 4 also stated that change in mindset of people can be challenging. He added, "In my own company, everyone was quite reluctant about following DevOps because all code has to go through certain sets of checks and builds, and essentially the process of landing the change into the repository was delayed or slowed down because of this. But later on, they realized the trade-off between the time taken to fix bugs in production versus fixing the bugs well in advance in Dev phase was way more practical than later phases. Then we got a buy in." Interviewee 9 also stated that cultural shift was a challenge. Interviewee 11 said, "You can't force culture." He added that culture change takes time. Interviewee 12 also shared a similar view. He commented that establishing DevOps culture takes time.

Resistance from employees was identified as a challenge by interviewees 9 and 11. Interviewee 11 said, "There are people who resist the change. They will be like I'm a developer and I have this job description, my responsibilities. I'm a developer and now I have to worry about all these other things. I don't care about operations or test cases." Interviewee 12 also stated that resistance from employees can be challenging. He said, "Some employees will not really be open to new changes. Because they have this particular job and they know how to do it. When you try to bring a new change, for example, if you tell an operations employee he has to suddenly take part in development activities also, he will try to resist it. The reason is they are scared because they have to learn something new and entirely different. In other words, they are afraid step out of their comfort zones."

Training was identified as a challenge to DevOps adoption by interviewees 3, 4 and 9. Interviewee 9 added, "Training some 7000-8000 people in the organizational level is not an easy task. People come from different backgrounds like technical, non-technical, financial, business management, etc. Making them to understand things from their level of perspective and bringing them to collaborate is not easy. It takes years of effort. Like 2-3 years of effort."

Interviewee 7 stated that security clearance was a challenge faced in their organization. Their organization handles very sensitive data like credit card information. "Now we are using an IBM tool which can be replaced by Ansible. But for deploying through Ansible, we need root access, for which there is no clearance from the security team. For deploying packages into public cloud like AWS, we need legal clearance from the

US security team, which is pending. Though we have inbuilt clouds, it is not as efficient as AWS. We are currently trying to get clearance."

Interviewee 3 identified technical debt as a challenge to DevOps adoption. He added, "There is a right way of coding and a quick and dirty way of doing it. There is no incentive for the developer to do it the right way if it takes longer. He will try to clear backlog asap. Quick and dirty way might not survive for long, which means someone again in the future has to remedy it. So, you are incurring debt now, which you have to pay off in the future. Also, talking about performance goals, if not everyone is measured on the same level, then there is no incentive for the developer to do the right thing in DevOps. In the waterfall methodology, we have peer code review, so someone will uncover it and the developer can be held responsible. This is not the case in DevOps." He commented that technical debt is a very bad practice.

Lack of collocated teams was identified as a challenge by interviewee 3. He added that if the teams are not collocated at the same geographical location, the number of hand-offs between teams will increase, and communication becomes difficult. This will hinder DevOps adoption.

Interviewee 4 identified making everyone accustomed to a new workflow as a challenge to DevOps adoption. He commented by saying, "We faced a problem because we had to organize certain trainings in place to bring everyone on the same page. Because no one was used to working in this kind of workflow."

Tool adoption was identified as a challenge to DevOps adoption by several interviewees. Interviewee 3 commented that most new tools are cloud-based, which is a new concept for long timers. Interviewee 4 stated that adoption of technologies and tools was a challenge faced in their organization. He commented, "Learning tools was pretty challenging. Once we had full DevOps activity in place, the teams had to learn about puppet (tool)." Interviewee 9 added, "Consider open source tools. Unlike in commercial tools, you have to customize and configure them. For that, you need to have a certain level of understanding. Adopting them will not be that easy." Interviewee 12 said, "These days cloud based tools, for example Infrastructure as a Service IaaS tools like Amazon Web Services and google cloud are becoming very popular. Many companies have some predefined tools, called 'Legacy tools' and if they want to adopt DevOps they should make sure that their existing tools are compatible with these IaaS tools."

Learning curve was identified as a challenge to DevOps adoption by interviewees 4, 5 and 8. Interviewee 8 added, "The developers need to know how to set up servers. The operation guys can be new to many things; they knew about all about infrastructure, how to set up servers and how to do it manually. But they need to learn the code. So it can be challenging for everyone."

Buy in from management was identified as a challenge by interviewees 4 and 11. Both the interviewees expressed the view that convincing management to implement DevOps can be challenging.

Interviewee 5 stated that writing custom scripts for automating tasks can be challenging. Interviewee 9 expressed that personnel may feel over-burdened with DevOps. He also pointed out that hidden complexities and hidden costs could occur in the later stages of adoption. Fear of measurement in employees was identified as another challenge. Interviewee 9 said, "Measurement tools will log details like the number of hours a particular resource worked, number of lines of code he wrote. The progress of the developer will be recorded. So the developer might focus more on the progress, i.e. delivering something rather than being creative. Indirectly, he will be afraid of what the manager might think. Not everyone might be like that, but some people will. Also, with these tools, we can easily identify what is going wrong. This will increase stress level among employees."

Integration of different tools was specified as a challenge by interviewees 9 and 12. Interviewee 12 said, "DevOps will need new workflows, so the existing workflows of the organization should be changed. For this, again the organization should invest in new tools. Another thing is companies will see tools as solutions to different problems. After getting a new tool, integrating it with the other existing tools will become a huge challenge."

Migrating existing applications to cloud was specified as a challenge by interviewees 2 and 10. Interviewee 10 also added that not having proper resources to configure environments (production, testing) can cause the environments to fail. Interviewee 12 stated that fear of failure in managers can be challenging to DevOps adoption.

4.4 Codes and Themes for RQ3

A total of 19 codes for the solutions to overcome the challenges faced by organizations during DevOps adoption were identified from the 12 interview transcripts. The codes were selected based on the inclusion criteria relevance to study, and frequency of occurrence. The identified codes were categorized into the theme "Solutions". The Table 4.4 presents the codes and themes identified for RQ3.

4.4.1 Analysis of Interview Responses for RQ3

In this section, we analyse the codes related to the solutions to overcome the challenges faced while adopting DevOps in organizations. 18 codes were identified from the 12 interviews, and grouped into the theme "Solutions".

Codes	Code Inclusion Criteria	Themes
Specify release criteria for each project	Relevance	Solutions
Document all challenges	Relevance	Solutions
Using a flexible system architecture	Relevance	Solutions
Better to start DevOps in small projects	Relevance	Solutions
Maintain an open communication channel	Relevance	Solutions
Minimize handoffs using a good alignment of teams	Relevance	Solutions
Maintain dashboards	Relevance	Solutions
Include QA people	Relevance	Solutions
Have a Proof of Concept (POC) in place	Relevance	Solutions
Train people	Frequency, Relevance	Solutions
Hire people with a DevOps mindset	Relevance	Solutions
Clarify issues as soon as they arise	Relevance	Solutions
Understanding of business requirements	Relevance	Solutions
Document errors and their solutions	Relevance	Solutions
Make decisions prior to the start of projects	Relevance	Solutions
Having a proper management structure	Relevance	Solutions
Conduct short meetings to encourage communication	Relevance	Solutions
Estimate adoption costs and risks beforehand	Relevance	Solutions
Use tools to bridge gap between old and new tools	Relevance	Solutions

Table 4.4: Codes and Themes for RQ3

Solutions: Interviewee 1 proposed specifying release criteria for each project as a solution. He added, "I believe, in order to get into DevOps, we need to have a proper release criteria. Release criteria is used to determine when the code is ready to be pushed into production. For example, you may specify 70 percent unit test case coverage, 80 percent functional test case coverage as the release criteria for a backend code. It is different for each project. It cannot be defined in a book. Every project should have its own release criteria. The code should be rolled into production only if the release criteria is met." He also suggested that documenting the challenges that pop up in different projects can be helpful for future projects.

Interviewee 2 proposed using a flexible system architecture. He said, "For example, today we use AWS and if Amazon ups the price, then we will move to another cloud based architecture like google cloud. Again, this will have a different kind of architecture. We need to make sure that our architecture must be in such a way that if we need to move to any other service, we can do so without any big changes."

Interviewee 3 suggested that it is better to start small. He added, "You can't just take a company that practices waterfall methodology, and start DevOps in it straight away.

Start small. Smaller project is better. Small in terms of complexity and time."

Interviewee 3 also suggested that using a good alignment of teams can minimize the handoffs. He also suggested that maintaining an open communication channel can help. He added, "After forming a team, there should be norming of the team also. Everyone must get used to how everybody else works, and there should be an open communication channel so that everybody talks to each other." He also suggests maintaining dashboards. He states that teams should communicate about queues or backlogs present. He said, "There should be a transparency of communication of the same information among all the team members. For that you can use dashboards. Everyone has to be able to see how many builds have failed or passed, how and why, etc."

Interviewee 3 also proposed to include Quality Assurance (QA) people into DevOps teams. He said, "Devops only talks about developer and operations. It is probably best to include QA people into that one. That way integration testing might still happen at the end, but at least there will be additional set of eyes looking at the test cases, making sure a holistic way of testing is done."

Interviewee 4 suggested that having a Proof of Concept (POC) can help in getting buy in from managers and customers. He said, "You just take some small component and apply DevOps to it, and showcase the effort you have put in, the amount of time you are saving, the stability of code, etc. Essentially explain that with one module and get them to buy in."

Interviewees 5, 9 and 12 suggested that training people can help to reduce fear of change. Interviewee 6 suggested that hiring people with a DevOps mindset can help. He added, "Look for those people who have the mindset of DevOps. They should be agile, have a mindset of doing automation, should be communicative."

Interviewee 9 also suggested that clarifying issues as soon as they arise can help. He added, "A single person cannot resolve issues at all levels, so more collaboration is required between team members." He also suggested that providing an understanding of business requirements to every employee can help. He said, "Previously, it was enough when only the manager or the architect understood about requirements. Now that is not the case."

Interviewee 3 also suggested that documenting errors and their solutions can help. He commented that having proper error logs for listing errors, how and why they occurred, and how to prevent them from reoccurring can help.

Interviewee 10 suggested that decisions should be made prior to the start of projects. He commented that changing decisions in the middle of the project could affect the environment.

Interviewee 11 suggested having a proper management structure would help. He also stated that having proper communication channels can help whenever issues occur.

Interviewee 12 suggested that conducting short meetings can encourage teams to share knowledge and ideas. He also suggested that estimating DevOps adoption costs and possible risks beforehand can help to reduce the fear of failure in managers. He added that based on the estimation, managers can determine whether or not to adopt it. He also proposed a solution to bridge the compatibility gap between tools. He said, "There are some new tools like CloudShell and Helion CloudSystem that will bridge the compatibility gap between the old company tools and new cloud-based tools."

4.5 Codes and Themes for RQ4

A total of 11 codes for the practices that an organization needs to include for adopting DevOps were identified from the 12 interview transcripts. The codes were selected based on the inclusion criteria relevance to study, and frequency of occurrence. The identified codes were categorized into the themes "Principles" and "Practices". The Table 4.5 presents the codes and themes identified for RQ4.

Codes	Code Inclusion Criteria	Themes
Automation	Frequency, Relevance	Principles
Continuous Integration	Frequency, Relevance	Practices
Continuous Deployment	Frequency, Relevance	Practices
Continuous Testing	Frequency, Relevance	Practices
Culture	Frequency, Relevance	Principles
Collaboration	Frequency, Relevance	Principles
Sharing	Frequency, Relevance	Principles
Infrastructure as Code	Relevance	Principles
Continuous Delivery	Frequency, Relevance	Practices
Continuous Monitoring	Frequency, Relevance	Practices
Continuous Feedback	Frequency, Relevance	Practices

Table 4.5: Codes and Themes for RQ4

4.5.1 Analysis of Interview Responses for RQ4

In this section, we analyse the codes related to the practices that an organization needs to include while adopting DevOps. 11 codes were identified from the 12 interviews, and grouped into the themes "Principles" and "Practices".

Principles: Automation, Culture, Collaboration, Sharing and Infrastructure as a Code were identified as the principles of DevOps from the interviews.

Automation was identified as a principle of DevOps from all the interviews. Interviewee 2 commented that automation is a good thing about DevOps, as there is no manual intervention. Interviewee 8 said that automation makes tasks easier, and saves time. Interviewee 5 stated that automation is advantageous as it reduces chances of human error. Interviewee 6 commented that automation saves time and gives quick results. Interviewee 9 commented that minimal human intervention and maximum automation improves the delivery speed and quality. Interviewee 12 also stated that automation results in faster delivery.

Culture was identified as another principle of DevOps from the interviews 8 and 11. Interviewee 8 commented that DevOps is more like a culture. He said that DevOps increases responsibility among all the teams involved. Interviewee 11 also shared a similar view. He commented that because DevOps is a culture, you don't force it. He added that creation of conditions which break down the barriers between teams should be done from the management.

Collaboration was identified as another DevOps principle from the interviews 2, 4, 9 and 12. Interviewee 4 said that "DevOps is about collaboration between teams, and ensuring that the job gets done." Interviewee 9 commented that, "There are several teams in the organization like development, business analysts, QA teams, testing teams and operations teams. If there is no proper collaboration between teams, what we deliver will be much different from what is expected by the customer." He added that DevOps increases collaboration by removing silos in the organization. Interviewee 2 also commented that DevOps breaks down silos. Interviewee 12 said, "The main idea behind DevOps concept is collaboration between development and operations teams. There won't be any walls between the teams, and everyone can see what is going on. Because of this visibility, collaboration will be enhanced and problems can be easily solved."

Sharing was identified as another principle of DevOps from interviews 11 and 12. Interviewee 11 commented, "All the teams have to work together to figure out what needs to be done, to make sure all the changes are done in the right way." Interviewee 12 said, "Sharing of ideas and knowledge is very important."

Infrastructure as Code (IaC) was identified as another DevOps principle from interview 5. The interviewee stated that Infrastructure as Code (IaC) refers to writing code for automating tasks.

Practices: Continuous Integration, Continuous Deployment, Continuous Testing, Continuous Delivery, Continuous Monitoring and Continuous Feedback were identified as the practices of DevOps from the interviews.

Continuous Integration was identified as a DevOps practice from all the interviews. Interviewee 11 said, "You have multiple people who want to change the same file, all at once. Which order should the changes go in? You can use Git source control tool for that. To collaborate with all team members. What happens when there is a conflict between one developer's change and the next developer's change? What should you do then? Things like code review. It all goes through source control. Things such as approvals. These things are very important to continuous integration. Continuous integration requires multiple people trying to change the same piece of software to integrate the changes with each other, and test them to make sure that all the different changes are not breaking one another." Interviewee 3 commented, "Before continuous integration, when a large project was going on, developers used to develop and push the code into their development channels and to promote from one channel to the other, it used to take a very long time. The Source Code Management (SCM) people used to gather all the latest changes from SCM tools, then they used to compile, and push it into the environment. To push it to the environment, SCM people had to collaborate with the infrastructure people because the environment has to be ready, and all the paths must be set, the databases must be set and so on. Continuous integration came into play because of these issues."

Continuous Deployment was identified as another DevOps practice from all the interviews. Interviewee 11 commented "The analogy of DevOps is, to break down the silos. A silo is a part of an organisation, where you have developers who do all the software development and software change, and then they throw it over the wall to the next silo the QA. So the testers, take what the developers have built and they do their thing. They are doing agile, right? And then the developers do their two week sprint and then deliver to QA. The QA then does their two weeks of testing. Then the developers start working on their next sprint and the QA start working on theirs, and then once the testers are finished with their work, they hand it over to the operations guys to deploy to production. And it just blows up. There are many reasons for this and the primary reason is the divisions of the people. So that organisational division, with the team division, work division, how things are set up are really destructive in terms of trying to deliver a working product or a working system. I will clarify that this is mainly for the internet based systems. Most stuff these days is mostly connected to the internet, internet technologies. So really we are talking about what can be seen in a web browser. So, as these ideas come around, the language starts to form and DevOps appears. The cloud revolution happened and the tools started coming around, and people using these tools realise there is a better way to do these things, and all of these aspects come together where the tools in place to build this thing called deployment pipeline. But in order to support that deployment pipeline, the technical things about how you get from a development environment to a test environment to a staging environment to a production environment, all those environments should look as similar as possible. So basically they should all try to look like production. They should have

the same type of configuration, application, same type of data, same OS, same kind of network as possible, as much as reasonable."

Continuous Testing was identified as a DevOps practice from all the interviews. Interviewees 1, 2, 6, 7, 8 9, 10 and 12 commented that all test cases should be automated. Interviewees 2 and 7 added that they use Selenium tool for automating test cases in their organizations. Interviewee 4 said, "We have implemented the Openstack infrastructure development. The team basically works on the DevOps activities. We have tools like Jenkins, and the whole code is based on python. So, we have builds with respect to python and every change that goes into the Git repository. Every change that goes into the repository is hit by lot of test cases which are powered by a tool. So, we at the end of the day ensure that when a developer pushes a code to the repository, it gets checked at every phase. When a developer pushes a change sometimes, before the code even gets landed into Git, they run certain prerequisite checks. It could be just the compilation of the code, or it could be a unit test, or something else. Once that change has been approved, the developers take a look at the changes and run tests which involve full iteration or full system tests. Once those tests get approved, the change gets pushed into the repository.

Continuous Delivery was identified as another DevOps practice from all the interviews. Interviewee 11 commented, "The DevOps part of continuous delivery is really about bringing the change management from all of the different organisational divisions, all the way back to the start of the deployment pipeline. It is called a pipeline because, how many directions does fluid flow in a pipeline? One. So here the fluid is the changes in the pipeline, and DevOps is the people. It is a cultural team. We build something, double check the pipeline, whether it is what it is supposed to be, test it, and verify before we deploy into production. So you have two things, DevOps and continuous delivery. You cannot do one without the other. Customers ask me we want to deploy daily, we want to deploy faster, with higher quality. If I want to do those things, I need DevOps. The technical things, how you set up a server, different tools etc., they are the easy things. They do not work if the organisation is divided with different silos. You will not be able to do continuous delivery, because you cannot bring all that change management back to the start of the deployment pipeline. All these people; developers, QA and operations people have to work together to figure out what needs to be done, to make sure all the changes are done in the right way, test it together. Different people collaborate on the changes together on the same file, check the source code, integrate the stuff and run it through continuous integration and deploy that change and shouldn't have to change it at every stage of the deployment. It is the same chain that repeats to get us through the deployment pipeline.

Continuous Monitoring was identified as a DevOps practice from interviews 3, 9 and 12. Interviewee 3 suggested that monitoring of performance metrics is a good practice. Interviewee 12 added that they enabled continuous monitoring in their organization.

Continuous Feedback was identified as a DevOps practice from interviews 2, 9 and 12. Interviewee 2 commented that feedback is one of the best practices of DevOps. Interviewee 9 suggested that if companies release Minimum Viable Products (MVPs), the customers will give feedback about what they are expecting, what must be changed and so on. Based on the feedback, companies can improve the product, which will incrementally improve the value.

All the interviewees stated that there is no specific way of doing DevOps. Interviewee 2 commented "To do DevOps, we need to be agile. DevOps is a framework of practices, and you should adopt them based on your needs." Interviewee 3 commented "You can definitely use DevOps within any IT organization. All you you have to do is change management structure, change the culture, the thought process or the mindset." Interviewee 4 added, "If an organization wants to adopt DevOps, they need to break the silos between each teams, and they should change their whole workflow. They need to sit down with the development team, QA team, build team, production or operations team and find out how they can streamline the entire release process. They need to figure out what would work well for all the teams, and follow that process." Interviewee 5 commented that "We cannot just limit DevOps to only continuous integration or operations and development. It is a mindset which people should adopt to."

Interviewee 8 commented, "I think the definition of DevOps is not clear. Everyone has their own definition of Devops, every company has its own culture. It changes according to the company." Interviewee 9 suggested that based on the organizational structure, communication flow, and requirements, the organizations should develop a framework. He added that organizations do not have to include all the principles, just only what they need.

Interviewee 12 added, "DevOps is not just a defined process that you can simply implement. It is more like a mindset which will change the way different teams in the organization interact. In this case, the development and operations teams. With DevOps, there will be visibility on what is happening in the project to both the teams and they can collaborate easily. There is no specific way of doing this and we will adopt the different tools we need based on our project needs. What practices an organization should include will depend on the organization structure, requirements, its capabilities and project type."

4.6 Codes and Themes for RQ5

A total of 14 codes for the problems faced by DevOps teams during continuous integration, deployment and testing have been identified from the 12 interview transcripts. The codes were selected based on the inclusion criteria relevance to study, and fre-

quency of occurrence. The identified codes were categorized into the theme "Problems". The Table 4.6 presents the codes and themes identified for RQ5.

Codes	Code Inclusion Criteria	Themes
Compatibility issues while integrating different tools	Frequency, Relevance	Problems
Testing environment flakiness	Relevance	Problems
Reverting from database changes	Relevance	Problems
Setting up infrastructure for the first time	Relevance	Problems
Production failures	Relevance	Problems
Teams might not be collocated	Relevance	Problems
Connectivity issues during integration	Relevance	Problems
Getting resources for solving workflow problems	Relevance	Problems
Lack of proper communication between teams	Relevance	Problems
Learning new technologies	Relevance	Problems
Having to manually check for application logs in Jenkins	Relevance	Problems
Accessibility	Relevance	Problems
Finding the right tools	Relevance	Problems
Time plans of projects might overlap	Relevance	Problems

Table 4.6: Codes and Themes for RQ5

4.6.1 Analysis of Interview Responses for RQ5

In this section, we analyse the codes related to the problems faced by DevOps teams during continuous integration, deployment and testing. 14 codes were identified from the 12 interviews, and grouped into the theme "Problems".

Problems: Compatibility issues while integrating different tools was identified as a problem faced by DevOps teams from the interviews 9 and 12. Interviewee 12 commented, "If the company's legacy tools are not compatible with the new cloud-based tools then the DevOps teams must struggle."

Interviewee 1 stated testing environment flakiness as a problem faced by DevOps teams. He explained, "In continuous integration, when the developer writes a code, unit test cases should run. If all the unit test cases are green, then it should build the code. Let us say all the unit test cases should be passed in step 1, right? So every time I write code, the unit test coverage reduces because I am increasing the number of lines. In very few cases, you decrease the number of lines. But if I increase the number of lines, I may not add a unit test case. That means my unit test coverage comes down. Let us say earlier it was 60 percent. Now if I write 100 more lines of code, it comes down to 58 percent. So that means I should have another checkpoint over there, like my unit test coverage should be same or more than the previous build."

Interviewee 2 stated that reverting from database changes could be a problem. "If the database version changes, it could be a problem. When database changes are made, like SQL insert or delete any server, reverting is a challenge. In case you move to a container based model (like Docker), the biggest challenge is making the database changes. "

Setting up infrastructure for the first time was identified as a problem by interviewee 4. He commented, "Setting up the continuous deployment infrastructure for the first time was challenging for us." Interviewee 3 stated that production failures can be a problem. He added, "Reproducing production failures will be very difficult in DevOps, because your build has already gone, and narrowing down what build has caused production failure, going back and trying to replicate that in the lower environment is a huge challenge." He also stated that the DevOps teams may face problems if they are not collocated, due to lack of proper communication. Interviewee 5 stated that during integration, the teams might face connectivity issues to the server where they deploy the code.

Interviewee 4 also stated that getting resources for solving workflow problems could be a problem. He commented, "It is hard to get DevOps professionals. We have to get good engineers to solve problems with respect to the workflow. Finding the resources. It was one problem we faced."

Interviewee 12 stated that lack of proper communication between teams could be a problem. He said, "If the teams do not communicate properly, deployments will be pushed back or delayed for weeks or months." Interviewee 4 stated that learning new technologies is another problem DevOps teams face.

Interviewee 5 said that having to manually check for application logs in Jenkins is a problem. He added, "We deploy the code to the server using Jenkins. The custom jobs in Jenkins are written such that it will not show the logs of the server post deployment. It would only say the code is deployed successfully. We have manually login into the server and check for the application logs."

Accessibility was stated as a problem by interviewee 7. He said, "In our organization, deployment teams are still waiting for access to the servers." It is to be noted that the organization handles sensitive data.

Finding the right tools was specified as a problem by interviewee 8. He said, "We used to do unit tests and functional tests. We used to try TDD (Test Driven Development). The only problem we faced using unit test is we need to write functions we are using. The functional test is very tricky, they were randomly sailing and we had a question of what to use. So, we were using selenium and lettuce testing. Lettuce

is a library in which we can write them like specific domain and specific language. Lettuce is very buggy, so we moved from lettuce. We then started using cucumber. The changes caused us some problems. So these are the problems we faced during functional testing; to find the suitable tool."

Interviewee 9 stated that if time plans of projects overlap, it could be a problem. He said, "They give you a 2-3 month time plan for a project, which contains details about the number of resources, and everything. Sometimes, you may complete the project before the expected 2 months. Sometimes, it may not be completed even after 3 months. There is a lot of difference between an idealistic plan and a realistic plan. If the manger comes up with a new plan for the project, that may overlap with other project time plans. Would you allocate resources to the new project or the old one? Would you shelve this existing project and take up a new one? How would you address these issues? You have to make a call. We face these kind of problems."

4.7 Discussion

A comprehensive discussion of the results obtained from the two qualitative research methods, interviews and SLR is provided in this section. The interview results are compared with the SLR results to validate them. Thus, the qualitative data will be validated and further used to obtain the quantitative data, which is the next step in the exploratory sequential design.

Identifying the benefits of implementing DevOps in organizations where agile development is in practice is an aim of this research. In addition to the benefits identified from the literature, the benefits identified from the interviews will also be included, and both results will be compared. The benefits identified from the literature are not specific to organizations which practice agile development in particular. Whereas, the benefits identified from the interviews are specific to organizations where agile development in practice. This is because in all the organizations of the interviewees, agile development was in practice prior to DevOps adoption. 20 benefits were identified from the interviews, and they were compared to those identified from the SLR. The benefits identified from both SLR and interviews are assigned individual IDs for reference. The final list of benefits after comparing and mapping the interview and SLR results is presented in the Table 4.7.

The benefits identified from both SLR and interviews were considered as the main benefits of implementing DevOps in organizations where agile development is in practice. The benefits identified from interviews are considered as the other important benefits, followed by those identified from SLR. Thus, increased collaboration between Dev and Ops teams, reduced time to market, faster feedback, reduced release cycle time, improved code quality, smooth and faster workflow, rapid product delivery, and easier implementation of business requirements were identified as the main benefits of

ID	Benefits	Source
B1	Increased collaboration between Dev and Ops teams	SLR, Interviews
B2	Frequent software releases	SLR
B3	Improved quality of software deployments	SLR
B4	Reduced time to market	SLR, Interviews
B5	Faster feedback	SLR, Interviews
B6	Reduced release cycle time	SLR, Interviews
B7	Empathy between Dev and Ops teams	Interviews
B8	Breaks down silos	Interviews
B9	Easier tracking	Interviews
B10	Easier detection of failures	SLR, Interviews
B11	Increased software reliability	SLR
B12	Improved code quality	SLR, Interviews
B13	Smooth and faster workflow	SLR, Interviews
B14	Increased system stability	Interviews
B15	Rapid product delivery	SLR, Interviews
B16	Increased customer satisfaction	SLR
B17	Increased overall organizational efficiency	SLR
B18	Increased responsiveness to business needs	SLR
B19	Balance between cost and quality	SLR
B20	Reduced need for rework	SLR
B21	Minimal manual intervention	Interviews
B22	Reduced communication overhead	SLR
B23	Increased predictability of software releases	SLR
B24	Automation	Interviews
B25	Easier implementation of customer requirements	SLR, Interviews
B26	Increased revenue	SLR

Table 4.7: Benefits identified from SLR and Interviews

implementing DevOps in organizations where agile development is in practice. The other benefits were identified as empathy between Dev and Ops teams, breaking down of silos, easier tracking, detection of failures, increased system stability, automation, and minimal manual intervention.

Identifying the challenges faced by organizations while implementing DevOps is another aim of this research. In addition to the challenges identified from the literature, the challenges identified from the interviews will also be included, and both results will be compared. 20 challenges were identified from the interviews, and they were compared to those identified from the SLR. The challenges from SLR and interviews are assigned individual IDs for reference. The final list of challenges after comparing and mapping the interview and SLR results is presented in the Table 4.8. The chal-

ID	Challenges	Source
C1	Lack of proper management structure	SLR
C2	No proper training	SLR, Interviews
C3	Geographical distribution of Dev and Ops teams	SLR, Interviews
C4	Integrating new tools with existing company tools	SLR, Interviews
C5	Legacy software	SLR, Interviews
C6	Resistance from employees	SLR, Interviews
C7	Monolithic system architecture	SLR
C8	Organizational structure	SLR
C9	Hidden complexities and hidden costs	Interviews
C10	Complexity due to multiple production environments	SLR
C11	Designing software architectures to make them work with DevOps	SLR, Interviews
C12	Technical debt	Interviews
C13	Buzzword perception	SLR
C14	Unclear definition and goals of DevOps adoption	SLR
C15	Security clearances	Interviews
C16	Buy-in from senior management	SLR, Interviews
C17	Buy-in from customer	SLR
C18	Fear of change	SLR, Interviews
C19	Fear of failure in managers	Interviews
C20	Insufficient knowledge on applications and environments	Interviews
C21	Making everyone accustomed to a new workflow	Interviews
C22	Writing custom scripts for automating tasks	Interviews
C23	Learning curve	Interviews
C24	Lack of proper resources to configure environments	Interviews
C25	Fear of measurement in employees	Interviews
C26	Culture and mindset of long timers	SLR, Interviews
C27	Limited visibility of production environments in embedded systems	SLR
C28	Lack of proper automation tools in embedded systems domain	SLR

Table 4.8: Challenges identified from SLR and Interviews

lenges identified from both SLR and interviews were considered as the most important challenges of DevOps adoption in organizations. The challenges identified from interviews were considered as the other important challenges, followed by those identified from SLR. No proper training for DevOps in organizations, geographical distribution of Dev and Ops teams, legacy software, resistance from employees, designing architectures to make them work with DevOps, buy-in from senior management, fear of change, culture and mindset of long timers are the most important challenges faced by organizations while adopting DevOps. Then, the tool adoption problems like integrating new tools with existing company tools, hidden complexities and costs, technical debt, fear of failure in managers, insufficient knowledge on applications and environ-

ments in case of outsourcing projects, making everyone accustomed to a new workflow, writing custom scripts for automating tasks, learning curve, security clearances, lack of proper resources to configure environments, and fear of measurement in employees are the other important challenges faced during DevOps adoption.

Several new challenges to DevOps adoption were identified from the interviews. The reason behind obtaining different responses could be pertained to the different roles and organizations of interviewees. Thus, the responses should be validated using the quantitative research method, survey. The challenges will be included in the survey questionnaire, and assigned an ordinal scale to identify the importance level of each challenge. As the number respondents for survey will be higher than the interviewees, the results can be validated.

Identifying the solutions/ mitigation strategies to overcome the DevOps adoption challenges is another objective of this research. There is no relevant literature to the solutions to overcome the DevOps adoption challenges, but the solutions were identified from the interviews. Thus, it was not possible to compare the interview findings with literature. The 18 solutions identified from the interviews are assigned individual IDs for reference, and are listed in the Table 4.9. Several solutions to overcome the adop-

ID	Solutions	Source
S1	Specify release criteria for each project	Interviews
S2	Document all challenges	Interviews
S3	Using a flexible system architecture	Interviews
S4	Better to start DevOps in small projects	Interviews
S5	Maintain an open communication channel	Interviews
S6	Minimize handoffs using a good alignment of teams	Interviews
S7	Include QA people	Interviews
S8	Have a Proof of Concept (POC) in place	Interviews
S9	Train people	Interviews
S10	Hire people with a DevOps mindset	Interviews
S11	Clarify issues as soon as they arise	Interviews
S12	Understanding of business requirements	Interviews
S13	Document errors and their solutions	Interviews
S14	Make decisions prior to the start of projects	Interviews
S15	Having a proper management structure	Interviews
S16	Conduct short meetings to encourage communication	Interviews
S17	Estimate adoption costs and risks beforehand	Interviews
S18	Use tools to bridge gap between old and new tools	Interviews

Table 4.9: Solutions identified from Interviews

tion challenges were proposed by the interviewees. The online survey will be used to

further validate the solutions, and identify any new solutions.

Identifying the practices that an organization should include for adopting DevOps is another aim of this research. Several practices (technical) were identified from the literature and interviews. In addition to the practices identified from the literature, the practices identified from the interviews will also be included, and both results will be compared. 6 practices were identified from the interviews, and they were compared to those identified from the SLR. The practices from SLR and interviews are assigned individual IDs for reference. The final list of practices after comparing and mapping the interview and SLR results is presented in the Table 4.10. The practices continuous integration, deployment, testing, delivery, monitoring and feedback were identified from both SLR and interviews, while continuous planning was identified from the SLR. The practices will be further validated using survey.

ID	Practices	Source
P1	Continuous Planning	SLR
P2	Continuous Integration	SLR, Interviews
P3	Continuous Deployment	SLR, Interviews
P4	Continuous Testing	SLR, Interviews
P5	Continuous Delivery	SLR, Interviews
P6	Continuous Monitoring	SLR, Interviews
P7	Continuous Feedback	SLR, Interviews

Table 4.10: Practices identified from SLR and Interviews

It was noted that DevOps teams face several problems during the continuous activities of DevOps like continuous integration, deployment and testing. There is no literature related to the problems faced by DevOps teams, but the problems were identified from the interviews. It was hence not possible to compare the interview findings with literature. The 14 problems identified from the interviews are assigned individual IDs for reference, and are listed in the Table 4.11.

ID	Problems	Source
Pb1	Compatibility issues while integrating different tools	Interviews
Pb2	Testing environment flakiness	Interviews
Pb3	Reverting from database changes	Interviews
Pb4	Setting up infrastructure for the first time	Interviews
Pb5	Production failures	Interviews
Pb6	Teams might not be collocated	Interviews
Pb7	Connectivity issues during integration	Interviews
Pb8	Getting resources for solving workflow problems	Interviews
Pb9	Lack of proper communication between teams	Interviews
Pb10	Learning new technologies	Interviews
Pb11	Having to manually check for application logs in Jenkins	Interviews
Pb12	Accessibility	Interviews
Pb13	Finding the right tools	Interviews
Pb14	Time plans of projects might overlap	Interviews

Table 4.11: Problems identified from the Interviews

Several problems faced by DevOps teams during continuous integration, deployment and testing were specified by the interviewees. The online survey will be used to further validate the problems, and identify any new problems.

4.8 Threats to Validity

The threats to validity for the interviews are presented in this section.

1) Internal Validity:
The internal validity threats to the interview data could be related to the selection of interviewees, design of the interview questionnaire, and incorrect interpretation of the collected data. To mitigate the threat of selection of interviewees, the target population was defined as software engineering practitioners that have worked with DevOps. Convenience sampling was done to select the interviewees. Some personal contacts who are working with DevOps were selected as the interviewees, and DevOps practitioners identified using LinkedIn were chosen. To mitigate the threat of the design of interview questionnaire, a semi-structured interview questionnaire having both open-ended and close-ended questions was designed based on the SLR findings. The questions were designed to be easily understandable, and continuously improvised by taking feedback from the supervisor. Questions related to the demographics of interviewees, and the present study were included. Semi-structured questionnaire was designed as it facilitates more interaction. All the 12 interviewees were asked the same set of questions, and new questions were spontaneously framed and asked whenever needed. The interviewees were promised complete confidentiality to obtain genuine and valid answers.

To mitigate the threat of incorrect interpretation of collected data due to researcher bias, audio recording was done in all interviews after taking permission from the interviewees. The data transcription was manually done without leaving out any sentence. Thematic coding was done to analyse the collected data. All transcripts were thoroughly read to familiarize with the data. Codes in each transcript were identified by manually examining the transcripts. Each code was highlighted with a different colour, based on the question, and categorized into themes based on the code inclusion criteria, frequency and relevance.

2) Construct Validity:
The construct validity threat to the interview data is related to the interview questionnaire not being able to obtain the aims and objectives of the research. To mitigate this threat, the questionnaire was designed based on the SLR findings. It was designed in the simplest way possible, to ensure understandability. Also, feedback from the supervisor was taken, and the questionnaire was continuously improved according to the feedback. The interviewees were also asked if the questions were understandable or not after completing the interviews, to ensure understandability of the questionnaire.

3) External Validity:
The external validity threat is related to the generalizability of the results. To mitigate this threat, interviewees having different roles and different experience were chosen. Also, it was ensured that no more than two people from the same organization were interviewed. Thus, we tried to make the results generalizable.

Chapter 5

Analysis of Survey Questionnaire Results

In this chapter, the data analysis and results of the online survey are described.

The steps followed for conducting the survey were defining the survey objectives, planning the survey, identifying the target population and sampling, creating the online questionnaire, validating the questionnaire, sending the questionnaire to respondents, collecting data and analysing the data [41]. Initially, the survey objectives were identified from the research questions. The objectives of the survey are

- To identify the benefits of implementing DevOps in organizations where agile development is in practice
- To identify the challenges faced by the organizations while adopting DevOps
- To identify the solutions to overcome the adoption challenges
- To identify the practices of DevOps that an organization must include while adopting DevOps
- To identify the problems faced by DevOps teams during continuous integration, deployment and testing
- To identify whether the results obtained from the SLR and interviews can be generalized to a larger population sample

Based on the survey objectives, the planning of survey was done. The survey was scheduled for a period of four weeks because of the time constraint.

5.1 Target Population and Sampling

The target population should be defined to obtain a sample [41]. The target population is the group or individuals who can answer the survey. For our research, the target population is software engineering practitioners that have worked with DevOps. Shull et al [41] note that "a valid sample is a representative subset of the target population".

There are probabilistic and non-probabilistic sampling methods. For our study, non-probabilistic sampling was employed for choosing the survey respondents. As the

target population is specific and limited, it is hard to rely on a random sample. We followed two non-probabilistic sampling methods for selecting the survey respondents, convenience sampling and snowball sampling. Convenience sampling refers to gathering responses from people who are available and willing to respond [41]. Snowball sampling refers to asking people who have answered the survey to recommend the survey other people who they think would answer it [41]. First, the population sample was selected from personal contacts, and practitioners from LinkedIn and Facebook. It was ensured that the population sample for survey is different from the interview respondents to get valid data. The link to the survey was posted in several online forums and groups related to DevOps on LinkedIn and Facebook. Additionally, the respondents were requested to forward the survey to their contacts who could answer it.

5.2 Designing the Survey Questionnaire

The survey questionnaire was designed based on the objectives, with both open-ended and closed questions. The questions were related to the demographics of the respondents, and the present study. The questionnaire was prepared using Google docs. The responses were stored in an excel sheet, which facilitated easier data analysis. The questions related to demographics include the type of organization, geographical location of the organization, size of the organization, role of the respondent in the organization, years of work experience of the respondent, and years of the respondent's experience with DevOps.

The questions related to the benefits of implementing DevOps in organizations include an open-ended question about the development methodology followed in the organization before adopting DevOps, a closed question about the benefits of implementing DevOps in organizations, which has options from which the respondents can choose. The benefits identified from both SLR and interviews were added as the options. The question was designed so that the respondents can select more than one option. Open-ended questions were also provided for the respondents to list out any other benefits of adopting DevOps.

The questions related to the challenges faced by organizations while adopting DevOps include a closed question which has the challenges identified from SLR and interviews listed as the options, and open-ended questions to identify any other challenges. In the closed question, each challenge was assigned an ordinal scale to identify the level of importance, with the options very challenging, challenging, somewhat challenging, not challenging and cannot specify. Another closed question was also included to identify whether DevOps adoption can be challenging for organizations that are implementing it for the first time. It was accompanied with an ordinal scale with the options strongly disagree, disagree, neither agree nor disagree, agree and strongly agree.

The questions related to the solutions to overcome the adoption challenges include a closed question which has the solutions identified from the interviews as the options. Open-ended questions were also included to identify any other solutions to overcome the adoption challenges.

A closed question with the practices of DevOps identified from SLR and interviews listed as the options, and open-ended questions to identify any other practices that an organization needs to include for adopting DevOps were included.

The questions related to the problems faced by DevOps teams during continuous integration, deployment and testing include a closed question with the problems identified from the interviews as the options, and open-ended questions to identify any other problems that DevOps teams face.

All the questions related to demographics and the study were marked as required to ensure complete responses. A text box was added at the end of the survey for the respondents to fill in their names and email ids, if they wanted to receive a compilation of the survey results.

5.3 Survey Questionnaire Validation

The survey questionnaire was validated to ensure the reliability of the questionnaire, and that the questions were clear and understandable. This way of evaluating the questionnaire is also called pre-testing [41]. The survey questionnaire was reviewed by the supervisor, and changes were made according to his feedback. After that, a pilot study of the survey was conducted. Pilot surveys are conducted by sending the questionnaire to a small sample. Pilot surveys can identify any problems with the questionnaire, determine the response rate, and also help in assessing the reliability of the questionnaire [41].

The pilot study was conducted on three software engineering practitioners that have worked with DevOps to ensure that the questions were understandable, and also to calculate the approximate time taken to answer the questionnaire. The criteria for validating the questionnaire was: understandability of the questions and options, questionnaire length and time taken for completing the survey. 25 questions were included in the questionnaire. All the three respondents commented that the questionnaire was understandable, and that it took around 10-15 minutes to answer it. Thus, the questionnaire was validated. After the pilot study, the respondents felt that some options seemed repetitive/ unnecessary. Such options were excluded.

After validating the questionnaire, it was sent to the population sample via emails and social networking sites. The link to the survey was also posted in several online

forums and social networking groups related to DevOps. To motivate the respondents to provide complete and accurate answers, a statement explaining the purpose of the research and its relevance to the respondents, importance of their responses, maximum time taken for answering the survey, and ensuring the anonymity of respondents was added to the questionnaire. 50 complete responses were obtained for the survey.

The questionnaire was prepared using google docs because it is free, and the responses are directly stored in an excel spreadsheet. Also, the spreadsheet is shareable. Both the researchers can access and edit the form and the results at the same time.

5.4 Results of the Survey

5.4.1 Analysis of the Demographics of the Respondents

The respondents were asked to mention their geographical location in the survey. Respondents from 15 countries have participated in the survey. Of them, most of the participants are from India (30%), USA (22%) and Sweden (16%). The list of other countries includes UK (6%), Israel (4%), Norway (4%), Germany (2%), Ireland (2%), Australia (2%), Uruguay (2%), Brazil (2%), Netherlands (2%), Poland (2%), Finland (2%) and Nigeria (2%). The geographical location of the 50 respondents is presented in the following figure.

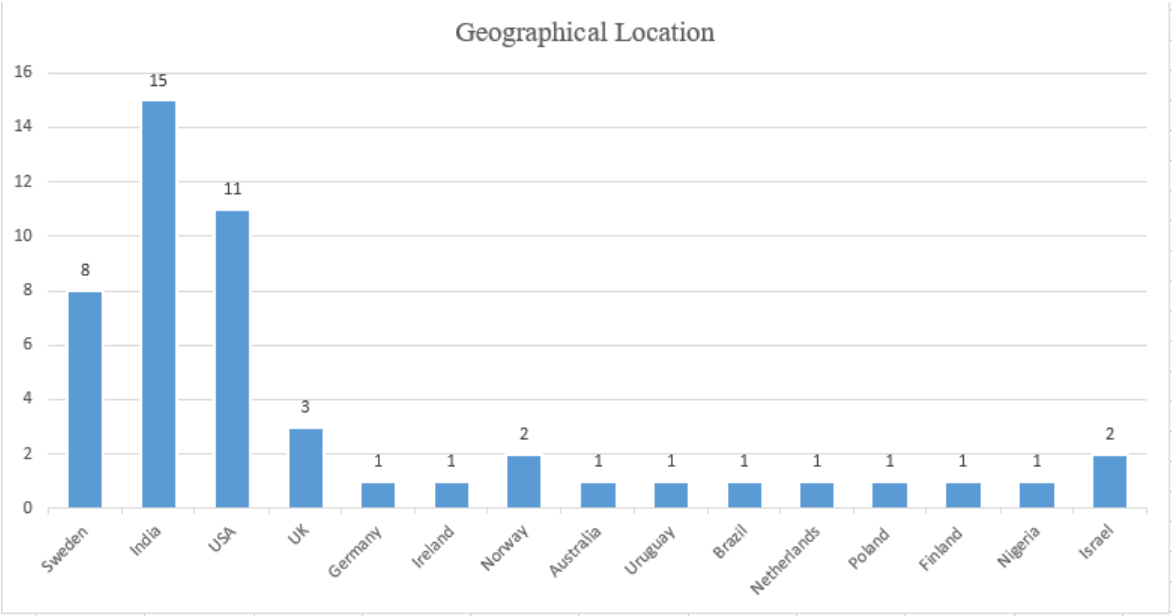

Figure 5.1: Geographical Location of Respondents

As the target population is software engineering practitioners that have worked with DevOps, the respondents were also asked to mention their role in the organization. The respondents had several roles, DevOps Consultant (14%), Network Engineer (6%), DevOps Engineer (22%), DevOps architect (4%), System Developer (2%), DevOps Configuration/ Release Manager (2%), DevOps developer (2%), DevOps delivery lead (6%), ALM/ DevOps practitioner (2%), IT Architect (4%), SSME (2%), Build and Release Engineer (2%),Technical lead (4%), Continuous delivery engineer (2%), IT Director (4%), Architect/ Developer (2%), IT Manager/ DevOps (10%), Lead Developer (2%),Lead DevOps (2%), Principal Architect/ Cloud & DevOps (2%), Senior Software Engineer (2%). The roles of the respondents are presented in the following table.

Roles	Frequency	Percentage of Responses
DevOps Consultant	7	14
Network Engineer	3	6
DevOps Engineer	11	22
DevOps Architect	2	4
System Developer	1	2
System Engineer	1	2
DevOps Configuration/ Release Manager	1	2
DevOps Developer	3	6
DevOps Delivery Lead	1	2
ALM/ DevOps Practitioner	1	2
IT Architect	2	4
SSME	1	2
Build and Release Engineer	1	2
Technical Lead	2	4
Continuous Delivery Engineer	1	2
IT Director	2	4
Architect/ Developer	1	2
IT Manager/ DevOps	5	10
Lead Developer	1	2
Lead DevOps	1	2
Principal Architect/ Cloud & DevOps	1	2
Senior Software Engineer	1	2

Table 5.1: Roles of the Respondents

The respondents were also asked to mention the size of the organizations, in terms of number of people. It was seen that most of the respondents come from organizations with size greater than 5000 (38%). The other respondents come from organizations with size 50-500 (22%), followed by 500-5000 (20%), 10-50 (12%), and less than 10 (8%). The size of organizations of respondents is presented in the following figure.

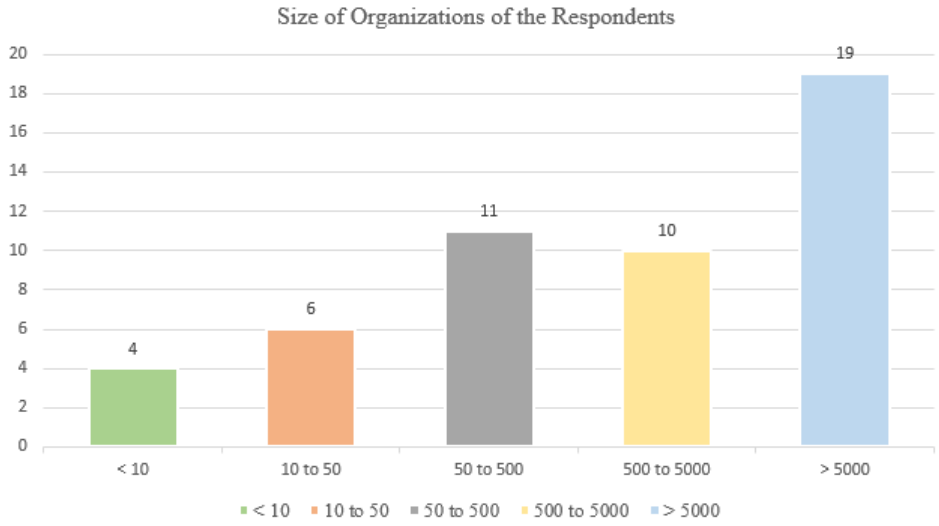

Figure 5.2: Size of Organizations of Respondents

The respondents were also asked to mention their overall work experience, in number of years. Most of the respondents have 4-9 years (30%), 10-15 years (30%) and 15-20 years (22%). The other respondents have greater than 20 years (10%) and less than 3 years (8%). The work experience of the respondents is presented in the following figure.

Figure 5.3: Work Experience of Respondents

As our research focuses on the adoption of DevOps in organizations, the respondents should have knowledge on DevOps, which makes their experience with DevOps an important factor to be considered. Thus, the respondents were also asked to mention the number of years of experience in working with DevOps. It was seen that most of

the respondents have 1-2.5 years (44%) and 2.5-4 years (36%). The other respondents have more than 4 years (20%) of experience with DevOps. The experience with DevOps of all the respondents is presented in the following figure.

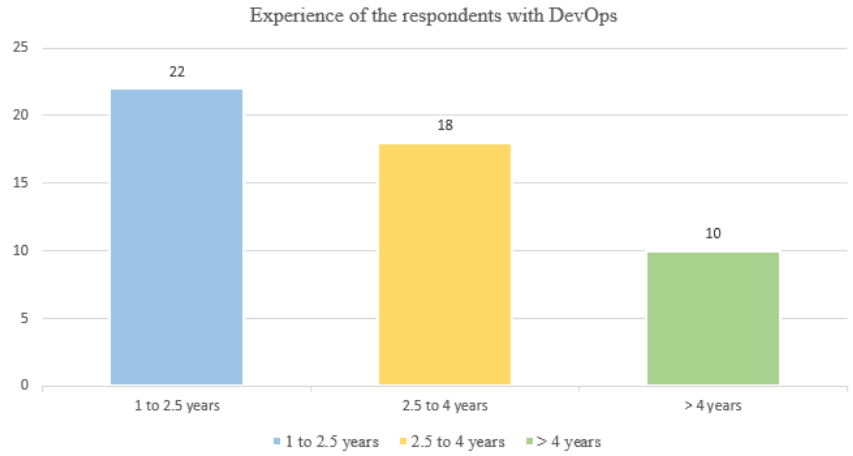

Figure 5.4: Experience with DevOps

Also, the respondents were asked to mention the development methodology followed in their organizations before adopting DevOps. Most of the respondents answered that they followed Agile methodologies (72%) and hybrid methodologies (22%). The other respondents answered that they followed plan-driven methodologies (6%). The methodologies followed in the respondents' organizations are presented in the following figure.

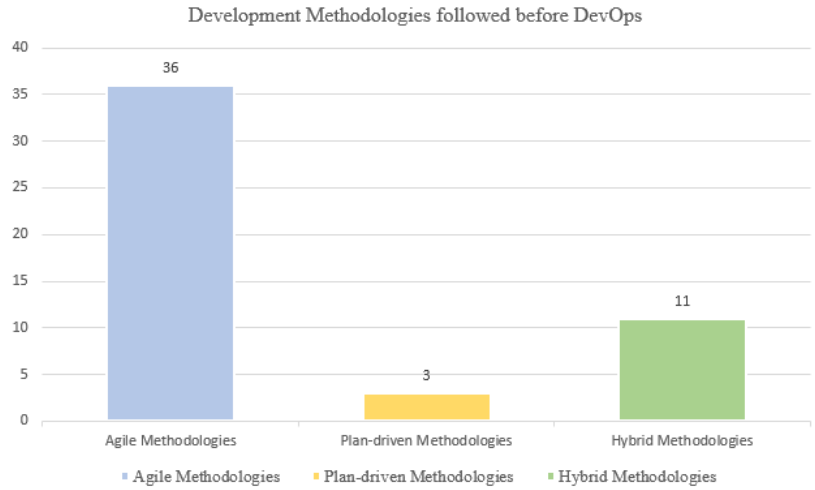

Figure 5.5: Methodology followed before DevOps

5.4.2 Benefits of Implementing DevOps in Organizations

One of the objectives of this research is to identify the benefits of implementing DevOps in organizations where agile development is in practice. Thus, the respondents were asked to identify the benefits which they know can be achieved after implementing DevOps in organizations. The benefits identified from SLR and interviews were included, and the respondents had to choose from those options. Additionally, they were also asked to specify any additional benefits of adopting DevOps, which were not listed in the options.

From the responses, it was seen that 84% of the respondents (42) identified increased collaboration between Dev and Ops teams as a benefit. Frequent software releases was identified as a benefit by 84% of respondents (42). 78% of respondents (39) identified improved quality of software deployments as a benefit. Reduced time to market was identified by 74% of respondents (37). Faster feedback was identified by 72% of respondents (36) and reduced release cycle time was identified by 70% of respondents (35). The other benefits were identified as empathy between Dev and Ops teams, breaking down silos, easier tracking, easier detection of failures, increased software reliability, improved code quality, smooth and faster workflow, increased system stability, rapid product delivery, increased customer satisfaction, increased overall organizational efficiency, and increased responsiveness to business needs by 48-58% of the respondents. 38% of respondents (19) identified balance between cost and quality as a benefit. Reduced need for rework was identified as a benefit by 40% of respondents (20). Minimal manual intervention was identified by 34% of respondents (17). Reduced communication overhead was identified as a benefit by 34% of respondents (17). 32% (16) identified increased predictability of software releases as a benefit, and 32% (16) identified automation as a benefit. Only 28% of respondents (14) identified easier implementation of customer requirements as a benefit. Only 24% of respondents (12) identified increased revenue as a benefit.

The following figure provides the benefits of implementing DevOps in organizations, as identified by the respondents.

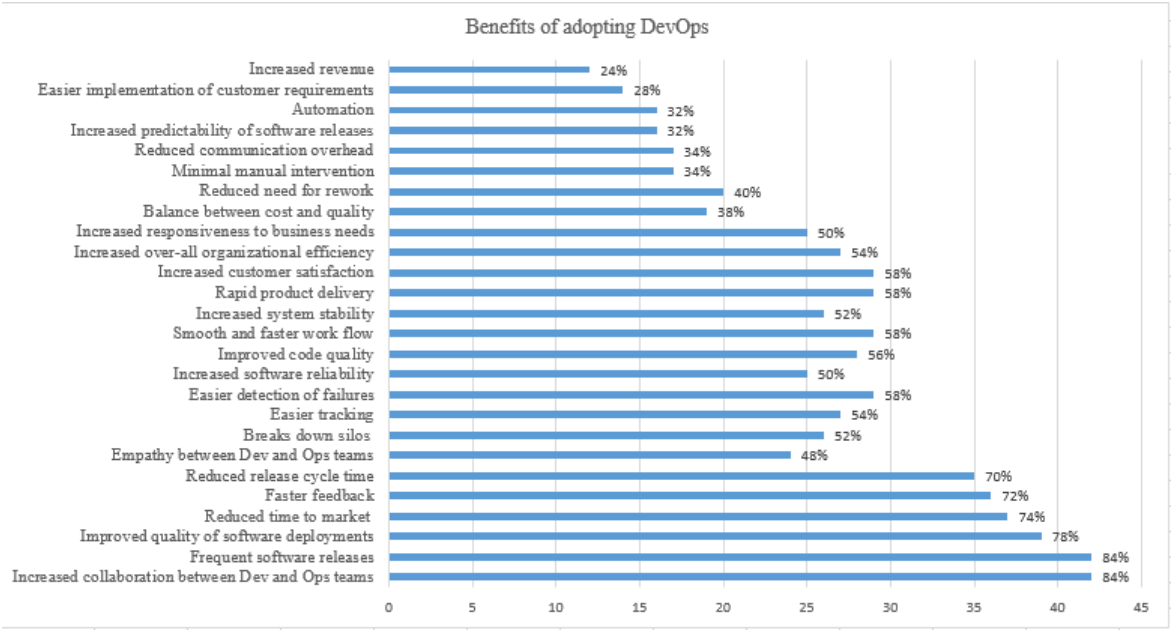

Figure 5.6: Benefits of adopting DevOps

5.4.3 Challenges faced by Organizations while adopting DevOps

Another objective of this research is to identify the challenges faced by organizations while adopting DevOps. The respondents were asked whether DevOps adoption can be challenging for organizations that are implementing it for the first time. They were given an ordinal scale of options, strongly disagree, disagree, neither agree nor disagree, agree and strongly agree. Also, the respondents were asked to identify which challenges can be faced by organizations while adopting DevOps. The challenges identified from SLR and interviews were listed as the options. They were also asked to specify the level of difficulty for each challenge by providing an ordinal scale with the options very challenging, challenging, somewhat challenging, not challenging and cannot specify.

The following figure presents the challenges faced by organizations while adopting DevOps, and their level of difficulty as specified by the respondents.

CHALLENGE	Very Challenging	Challenging	Somewhat Challenging	Not Challenging	Cannot Specify
Lack of proper management structure	17	24	7	2	0
No proper training	13	21	13	3	0
Geographical distribution of Dev and Ops teams	6	13	23	8	0
Integrating new tools with existing company tools	8	25	14	3	0
Legacy software	18	18	10	3	1
Resistance from employees	14	21	10	5	0
Monolithic system architecture	13	19	14	3	1
Organizational structure	11	19	14	5	1
Hidden complexities and hidden costs	4	19	18	8	1
Complexity due to multiple production environments	8	21	13	7	1
Designing software architectures to make them work with DevOps	9	22	14	5	0
Technical debt	2	28	13	5	2
Buzzword perception	3	10	21	12	4
Unclear definition and goals of DevOps adoption	8	17	18	6	1
Security clearances (when handling sensitive data)	9	20	13	7	1
Buy-in from senior management	7	19	15	6	3
Buy-in from customer	5	16	15	12	2
Fear of change	12	20	14	4	0
Fear of failure in managers	9	16	19	2	4
Insufficient knowledge on applications and environment (in case of outsourced projects)	15	17	13	3	2
Making everyone accustomed to a new workflow	10	18	17	5	0
Writing custom scripts for automating tasks	7	16	20	7	0
Learning curve	16	23	8	3	0
Lack of proper resources to configure environments	12	21	15	2	0
Fear of measurement in employees	8	13	24	3	2
Culture and mindset of long timers	19	16	13	2	0
Limited visibility of production environments in embedded systems domain	15	19	7	4	5
Lack of proper automation tools in embedded systems domain	12	16	14	3	5

Figure 5.7: DevOps adoption Challenges

The responses are presented as heat maps. The heat maps show the number of responses obtained for each challenge, with respect to its level of difficulty. The highest values are assigned a dark colour, and the least values are assigned a light colour. From the figure, it can be seen that legacy software and culture and mindset of long timers are very challenging for organizations adopting DevOps. Also, lack of proper management structure, no proper training, integrating new tools with existing company tools, resistance from employees, organizational structure, hidden complexities and costs, complexity due to multiple production environments, designing architectures to make them work with DevOps, and technical debt can be challenging for organizations.

Technical debt, security clearances, buy-in from management and customers, insufficient knowledge on applications and environments, making everyone accustomed to a new workflow, learning curve can be challenging for organizations adopting DevOps. Lack of proper resources to configure environments, limited visibility of production environments and lack of automation tools in embedded systems domain are challenging for organizations.

It can be seen that geographical distribution of Dev and Ops teams, buzzword perception, unclear definition and goals of DevOps adoption, fear of failure in managers, writing custom scripts for automating tasks, and fear of measurement in employees are not very challenging.

5.4.4 Solutions to overcome the Challenges faced during DevOps adoption

Identifying the solutions to overcome the challenges faced by organizations during DevOps adoption is another aim of this research. Thus, the respondents were asked to identify the solutions to overcome DevOps adoption challenges. The solutions identified from interviews were listed as the options. Additionally, they were also asked to specify any additional solutions to overcome the DevOps adoption challenges, which were not listed in the options.

88% of respondents (44) specified that it is better to start DevOps in small projects. 68% of respondents (34) identified conducting short meetings to encourage communication between teams as a solution. Proper training of employees was identified as a solution by 64% of respondents (32). Clarifying issues that arise as soon as possible was identified by 56% of respondents (28). 56% of respondents (28) identified hiring people with a DevOps mindset as a solution. 54% of respondents (27) identified using a flexible architecture as a solution. 52% of respondents (26) identified providing a better understanding of business requirements to every employee as a solution. 52% of respondents (26) identified having a proper management structure as a solution. 46% of respondents (23) identified having a Proof of Concept (PoC) in place as a solution.

42% of respondents (21) specified to minimize hand-offs using an open communication channel, and a good alignment of teams. 42% of respondents (21) identified specifying release criteria for each project, and documenting the challenges as a solution. 40% of respondents (20) identified documenting errors and their solutions. 38% of respondents (19) specified to estimate adoption costs and risks before adopting DevOps. 36% of respondents (18) identified including QA people as a solution. 14% of respondents (7) identified making decisions prior to start of projects as a solution. Only 12% of respondents (6) identified using tools like CloudShell and Helion CloudSystem for bridging the compatibility gap between legacy tools and the new cloud-based tools

as a solution.

The following figure presents the solutions to overcome the DevOps adoption challenges, as specified by the respondents.

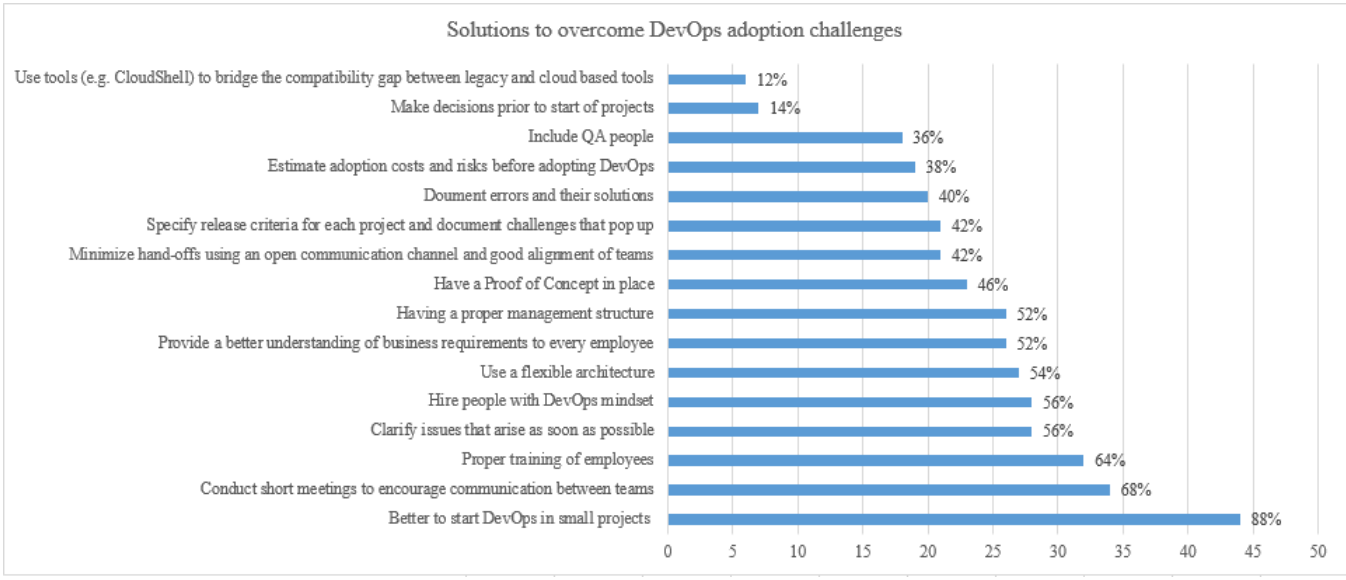

Figure 5.8: Solutions to overcome the DevOps adoption Challenges

5.4.5 Practices to be included by organizations for adopting DevOps

Another aim of this research is to identify the practices that an organization must include for adopting DevOps. Thus, the respondents were asked to identify the practices to be included by organizations. The practices identified from SLR and interviews are listed as the options. Also, the respondents were asked to specify any other practices which were not listed in the options.

94% (47) of respondents identified continuous integration as a practice. 88% of respondents (44) identified continuous testing as a practice. Continuous delivery was identified as a practice by 86% of respondents (43). Continuous deployment was identified as a practice by 84% of respondents (42). 76% of respondents (38) identified continuous monitoring as a practice. Continuous feedback was identified as a practice by 60% of respondents (30). Continuous planning was identified as a practice by 58% of respondents (29). We refer to continuous integration as CI, continuous deployment as CD and continuous testing as CT.

The following figure presents the practices to be included by organizations while adopting DevOps, as identified by the respondents.

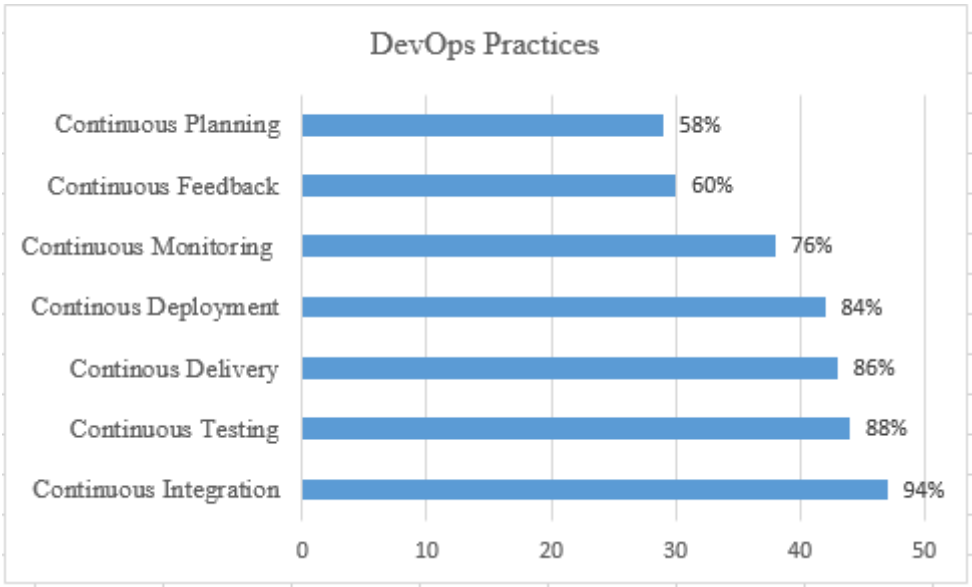

Figure 5.9: DevOps practices

5.4.6 Problems faced by DevOps Teams during Continuous Integration, Deployment and Testing

Another objective of this research is to identify the problems that DevOps teams face during continuous integration, deployment and testing. Thus, the respondents were asked to identify the problems faced by DevOps teams during continuous integration, deployment and testing. The problems identified from the interviews were listed as the options. Also, the respondents were asked to specify any other problems faced by DevOps teams, which were not listed in the options.

76% of respondents (38) identified configuring infrastructure for the first time as a problem. 74% of respondents (37) identified tool-related issues as a problem. 70% (35) of respondents identified production failures as a problem. 60% of respondents (30) identified learning new technologies as a problem. 56% of respondents (28) identified lack of proper communication between teams as a problem. 52% of respondents (26) identified testing environment flakiness as a problem. 50% of respondents (25) identified connectivity issues during integration as a problem. 44% of respondents (22) identified reverting from database changes as a problem. 40% of respondents (20) identified getting resources for solving problems related to workflow as a problem. 32% of respondents (16) identified lack of collocated teams as a problem. 28%

of respondents (14) identified overlapping time plans of different projects as a problem. 24% of respondents (12) identified accessibility as a problem. Only 12% of respondents (6) identified having to manually check for application logs in Jenkins as a problem.

The following figure presents the problems faced by DevOps teams during continuous integration, deployment and testing, as specified by the respondents.

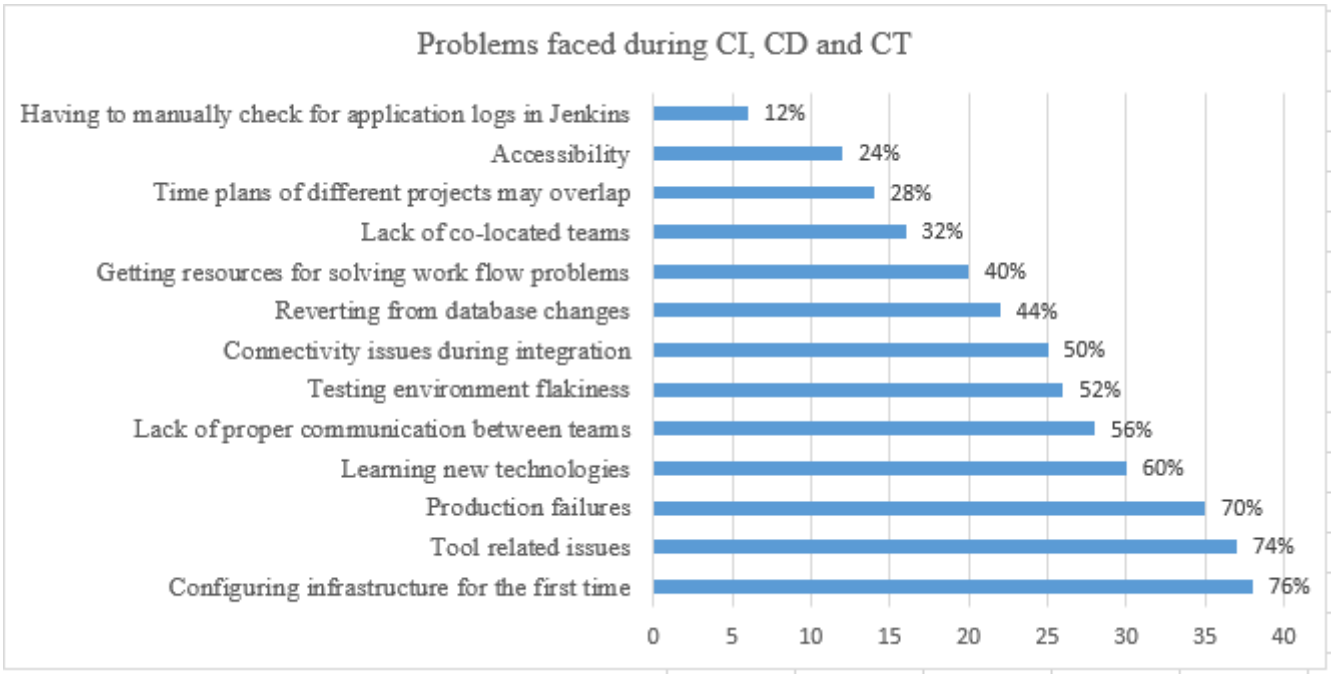

Figure 5.10: Problems faced by DevOps teams during CI, CD and CT

5.4.7 Results of the Open-ended Questions

In addition to the closed questions, open-ended questions were also included in the questionnaire so that the respondents can provide additional information, which was not listed in the options. The open-ended questions were included for the benefits of adopting DevOps, challenges faced by organizations during DevOps adoption, solutions to overcome the challenges, practices of DevOps, and the problems faced by DevOps teams during continuous integration, deployment, and testing. In this section, the results of the open-ended questions are elucidated. Thematic coding was done to analyse the responses obtained to the open-ended questions. Different answers were obtained for each question, and only the frequently stated terms were identified and coded. The code inclusion criteria was specified as frequency of occurrence.

The respondents were asked to specify the benefits of adopting DevOps, which were

not listed in the options of the closed question. Some respondents have specified new benefits, while some respondents commented that the benefits were mostly covered in the closed question options. For example, one respondent commented, "Transparency, clarity and visibility on team members actions which enables to differentiate between stars and slackers."

Another respondent commented, "Improving IT reputation within the business after countless failed IT project with the waterfall method."

Another respondent commented, "End of blame game. Developers better understanding the Operations related challenges and vice versa."

Among all the responses, only the mostly stated benefits are included. The benefits identified are:

- Less stressful and faster deployments.
- Customer and employee satisfaction.
- Lesser roll backs.
- Better performance of both people and systems.
- Budget control.
- Faster resolution of problems.
- More stable operating environments.
- Advantage over rival companies that do not do DevOps.
- Minimized deployment related downtime.

The respondents were asked to specify the challenges faced while adopting DevOps, which were not listed in the options of the closed question. Some respondents have specified new challenges, while some respondents commented that the challenges were mostly covered in the closed question options.

One respondent commented, "With DevOps the line of responsibility becomes blurred, which can be confusing in the beginning."

Another respondent commented, "Mindset change; Devs and Ops have traditionally been separated, and it takes some time and strong and committed leadership to bring them out of their comfort zones."

Another respondent commented, "Not just buy-in from customer/management alone, but a good grass roots company wide culture is needed for successful adoption of DevOps."

Another respondent commented, "Investment for transformation, and waiting till ROI is realised."

Another respondent commented, "Picture this. A lot of engineers and consultants have just finished acquiring Agile practices and some organisations have not even adopted them completely. DevOps comes along and now Agile teams become DevOps teams.. lot of resistance from employees to learn due to fear of failure. Not many brave team members put forward for DevOps POCs."
Among all the responses, only the mostly stated challenges are included. The challenges identified are:

- Unawareness of right tools for doing tasks.
- Insufficient knowledge on how to change/simplify current architecture to suit better into modern time automated environment.
- Finding the right and experienced personnel.

The respondents were asked to specify the solutions to overcome the challenges faced while adopting DevOps, which were not listed in the options of the closed question. Some respondents have specified new solutions, while some respondents commented that the solutions were mostly covered in the closed question options.

One respondent commented, "Have 'internships' between Dev and Ops teams. Ops should adopt the best practices from the Devs, e.g., test-driven development and pair-programming. Devs should learn about the challenges of running the software in the real-world where things often go wrong."

Another respondent commented, "Have representative of Dev in Ops and vice versa to influence the software architecture as well as operation aspect of the software."

Another respondent commented, "Cultural training to increase ownership from all teams on all issues. It isn't your problem, it is our problem."

Another respondent commented, "Start small, preferably 1 or max 2 teams with adopting DevOps. Then scale up to 3 or 4 and keep on going like an oil stain."

Among all the responses, only the mostly stated solutions are included. The solutions identified are:

- Encourage bonding of people and breaking down of silos.
- Communication is basic.
- Start small.
- Training and hiring those who possess a greater degree of empathy.

The respondents were asked to specify the practices that organizations must include while adopting DevOps, which were not listed in the options of the closed question. Some respondents have specified new practices, while some respondents commented that the practices were mostly covered in the closed question options.

One respondent commented, "Continuous communication and collaboration. DevOps is more than Dev and operations, it involves everyone in the team."

Another respondent commented, "Remember DevOps is not just about technical practices. Culture and communication is also very important."

Among all the responses, only the mostly stated practices are included. It was seen that the respondents only specified the non-technical practices, i.e. the DevOps principles. They are:

- Infrastructure as Code.
- Automation.
- Collaboration.
- Culture.
- Sharing.

The respondents were asked to specify the problems faced by DevOps teams during continuous integration, deployment and testing, which were not listed in the options of the closed question. Some respondents have specified new problems, while some respondents commented that the problems were mostly covered in the closed question options.

One respondent commented, "Often operations do not understand developers' problems, and sometimes even the nomenclature, and vice versa. There are often easier ways to integrate and test the software but it requires some architecture changes in the software."

Another respondent commented, "Every system may include industry standard tools and methods, but every architecture is unique and distinct, which makes tackling issues very tough for those new to a given architecture."

Among all the responses, only the mostly stated problems are included. The problems identified are:

- Entrenched differences/ antagonism between teams.
- There are many number of tools available in market, picking up the right tools is challenging for DevOps teams.

- Changing and complex software architectures.
- Undocumented legacy systems with no enough knowledge.
- Lack of ownership by various teams.
- Getting support for open source tools.

5.5 Threats to Validity

Basing on the study by Wohlin [55], the possible threats for survey which affect the execution of the research are identified and are listed. They are:

1) Internal Validity:
Internal validity is the threat related to the survey questionnaire, questions provided in the survey and the data equipped to answer those questions for the responses. These threats occur when the respondents do not find any particular question understandable, or could not answer the question because they feel the options are inappropriate, or not comfortable with the survey questionnaire. To mitigate these threats, the survey has been designed to be easily understandable. If the respondent did not find any suitable options within the question, an open-ended question "Please specify any other benefits/ challenges/ solutions/ practices/ problems which are not listed in the question" is provided, so that the respondents can conveniently answer in it. In addition to the closed question responses, the responses provided in the open ended question i.e., "Please specify any other which are not listed in the question" are also considered.

2) External Validity:
External validity is the risk regarding the target population of the survey. There is a risk involved in this area because the authors need to look for the people who are either practising, or practised DevOps. There are not many people familiar with this concept. So, looking for the people can be a problem. To mitigate this, the authors have posted the link through Facebook and LinkedIn in the groups related to DevOps. The authors have also sent the link to the DevOps practitioners in person to make sure that the survey is answered by the eligible respondents. The groups have DevOps practitioners all over the world and the particular respondents were chosen based on their LinkedIn profile information, and through personal contacts. The analysis and the information obtained from the survey can be inaccurate if the respondents are not familiar with the concept. These threats are mitigated carefully by choosing the eligible respondents, which enabled the authors to rely on the answers given by the respondents.

3) Construct Validity:
Construct Validity is the threat that occurs when the authors have not correctly interpreted the purpose of the questionnaire, or prepare questions that are irrelevant to the literature, or the questions provided do not add any information to the research. This

threat was mitigated by analysing the information obtained through SLR and the interviews, and formulating the survey questionnaire based on that information. The SLR and interview responses were mapped, as listing out several individual options may expand the length of the survey, which later affects the survey quality. The responses obtained from both SLR and interviews were mapped, and the options were provided in a simple way so that the respondents can understand them easily. The questions were framed in the survey in according to the research questions and objectives. A pilot study of the survey was conducted before sending it to the respondents. The survey was submitted to three respondents for the pilot study, and review from those respondents was taken. This helped the authors in assessing the understandability and the readiness of the survey.

4) Conclusion Validity:
Conclusion validity is the threat related to the trustiness, usefulness and quality of the results obtained from the survey. To mitigate this threat, the results obtained were analysed carefully and interpreted using heat maps and descriptive statistics. Using heat maps, the individual responses for each question can be observed. By including the mostly stated responses, the authors have carefully analysed the results of the open ended questions. The descriptive statistical analysis is used in considering the results obtained. With this analysis, the differences in between the responses are observed, and it is seen that they have occurred due to different roles of the respondents, and the differences in experience.

Chapter 6

Discussion

In this section, the results obtained from the mixed methods approach are used for answering the research questions. The results obtained from the qualitative and quantitative methods are mapped and presented along with the research questions.

6.1 Answering the RQs

RQ1: How does implementing DevOps in the organizations where agile is in practice help them?

The aim of this research question is to look into the benefits of adopting DevOps in organizations where agile development is in practice. The perceived benefits have a crucial role in determining whether or not to implement DevOps in an organization. The perceived benefits of adopting DevOps in organizations in general were identified from the literature, and also the actual benefits achieved in practice by organizations were identified from the interviews and survey. Agile development was in practice prior to DevOps adoption in all organizations of the interviewees. Whereas, agile and hybrid methodologies were in practice in organizations of 47 survey respondents. Before conducting the survey, the benefits identified from SLR and interviews were mapped together, and included in the survey questionnaire. Also, an open ended question was added in the survey to identify any additional benefits of implementing DevOps in organizations.

Based on the responses obtained from the survey, a finalized list of the benefits of implementing DevOps in organizations is created. The mostly stated responses are added first, followed by the lesser stated responses and so on. The final list of benefits of implementing DevOps in organizations where agile development is in practice is presented in the following table

Benefits of implementing DevOps in organizations where agile is in practice
Increased collaboration between Dev and Ops teams
Frequent software releases
Improved quality of software deployments
Reduced time to market
Faster feedback
Reduced release cycle time
Easier detection of failures
Smooth and faster workflow
Increased customer satisfaction
Rapid product delivery
Improved code quality
Increased overall organizational efficiency
Easier tracking
Increased system stability
Breaks down silos
Increased software reliability
Increased responsiveness to business needs
Empathy between Dev and Ops teams
Reduced need for rework
Balance between cost and quality
Minimal manual intervention
Reduced communication overhead
Automation
Increased predictability of releases
Easier implementation of customer requirements
Increased revenue
Additional benefits identified through survey
Less stressful and faster deployments
Employee satisfaction
Lesser roll backs
Better performance of both people and systems
Budget control
Faster resolution of problems
More stable operating environments
Advantage over rival companies that do not do DevOps
Minimized deployment related downtime

Table 6.1: Final List of Benefits

RQ2: What are the challenges faced by the organizations while adopting DevOps?

Identifying the challenges to the DevOps adoption process is another aim of this research. The challenges faced while adopting DevOps in organizations in general were identified from the literature, also several new challenges were identified from the interviews and survey. Before conducting the survey, the challenges identified from SLR and interviews were mapped together, and included in the survey questionnaire. Also, an open ended question was added in the survey to identify any additional challenges faced during DevOps adoption in organizations.

Based on the responses obtained from the survey, a finalized list of the challenges of implementing DevOps in organizations is created. The most important challenges identified from the survey responses are listed first, followed by the lesser important challenges and so on. The final list of challenges faced by organizations while adopting DevOps is presented in the following table.

Challenges faced during DevOps adoption
Culture and mindset of long timers
Legacy software
Technical debt
Integrating new tools with existing company tools
Lack of proper management structure
Learning curve
Designing software architectures to make them work with DevOps
No proper training for DevOps
Resistance from employees
Complexity due to multiple production environments
Lack of proper resources to configure environments
Security clearances when handling sensitive data
Fear of change
Monolithic system architecture
Limited visibility of production environments
Buy-in from senior management
Organizational structure
Hidden complexities and hidden costs
Making everyone accustomed to a new workflow
Insufficient knowledge on applications and environments in case of outsourced projects
Buy-in from customer
Lack of proper automation tools in embedded systems domain
Fear of measurement in employees
Geographical distribution of Dev and Ops teams
Buzzword perception
Writing custom scripts for automating tasks

Fear of failure in managers
Unclear definition and goals of DevOps adoption
Additional challenges identified from survey
Unawareness of right tools for doing tasks
Insufficient knowledge on modifying current architecture to suit an automated environment
Finding the right and experienced personnel

Table 6.2: Final List of Challenges

RQ3: How to overcome the challenges faced by the organizations in adopting DevOps?

Another aim of this research is to identify the solutions/ mitigation strategies to overcome the challenges faced during DevOps adoption. Identifying the solutions to overcome the challenges inherent with DevOps adoption can guide organizations whenever faced with a challenge. Also, several problems can be avoided in the first place if the organizations have an idea about the DevOps adoption challenges and their mitigation strategies. There is no literature related to the solutions to overcome the DevOps adoption challenges, but the solutions were identified from the interviews and survey. The solutions identified from the interviews were added to the survey questionnaire. Also, an open ended question was added in the survey to identify any additional solutions to overcome the DevOps adoption challenges.

Based on the responses obtained from the survey, a finalized list of the solutions to overcome the DevOps adoption challenges is created. The mostly stated solutions identified from the survey responses are listed first, followed by the lesser stated solutions and so on. The final list of solutions to overcome the DevOps adoption challenges is presented in the following table.

Solutions to overcome the DevOps adoption challenges
It is better to start DevOps in small projects
Conduct short meetings to encourage communication between teams
Proper training of employees
Clarify issues that arise as soon as possible
Hire people with a DevOps mindset
Use a flexible architecture
Provide a better understanding of business requirements to every employee
Having a proper management structure
Have a Proof of Concept in place
Minimize hand-offs using an open communication channel and a good alignment of teams
Specify release criteria for each project and document challenges that pop up

Document errors and their solutions
Estimate DevOps adoption costs and risks before adopting DevOps
Include Quality Assurance people
Make decisions prior to start of projects
Use tools (e.g. CloudShell) to bridge the compatibility gap between legacy and cloud based tools
Additional solutions identified from survey
Encourage bonding of people and breaking down of silos
Communication is basic
Training and hiring those who possess a greater degree of empathy

Table 6.3: Final List of Solutions

RQ4: What are the practices that an organization needs to include while adopting DevOps?

Identifying the practices that organizations must include as part of the DevOps adoption process is another aim of this research. The different practices of DevOps were identified from the literature, and also from the interviews and survey. Before conducting the survey, the practices identified from SLR and interviews were mapped together, and included in the survey questionnaire. Also, an open ended question was added in the survey to identify any other practices of DevOps.

Based on the responses obtained from the survey, a finalized list of the practices of DevOps is created. The mostly stated practices from the survey responses are listed first, followed by the lesser stated practices and so on. The final list of DevOps practices is presented in the following table.

To support the above mentioned technical practices, it has been noted that organiza-

DevOps Practices
Continuous Integration
Continuous Testing
Continuous Delivery
Continuous Deployment
Continuous Monitoring
Continuous Feedback
Continuous Planning

Table 6.4: Final List of Practices

tions should adopt the principles of Culture, Automation, Collaboration, Measurement, and Sharing. Also, it has been noted that is no specific way of doing DevOps, and organizations will adopt the different tools and practices based on the organizational structure, capabilities, project needs, requirements, and project type.

RQ5: What are the problems faced by the DevOps teams during continuous integration, deployment and testing?

The DevOps teams need to work together to continuously integrate, deploy and test the code, during which the teams may face some problems. Identifying the problems faced by the DevOps teams during the continuous activities of integration, deployment and testing is another aim of this research. There is no literature related to the problems faced by DevOps teams during continuous integration, deployment and testing, but the problems were identified from the interviews and survey. The problems identified from the interviews were added to the survey questionnaire. Also, an open ended question was added in the survey to identify any additional problems faced by the DevOps teams.

Based on the responses obtained from the survey, a finalized list of the problems faced by the DevOps teams is created. The mostly stated problems identified from the survey responses are listed first, followed by the lesser stated problems and so on. The final list of problems faced by DevOps teams is presented in the following table.

Problems faced by DevOps teams during CI, CD and CT
Configuring infrastructure for the first time
Tool related issues
Production failures
Learning new technologies
Lack of proper communication between teams
Testing environment flakiness
Connectivity issues during integration
Reverting from database changes
Getting resources for solving workflow problems
Lack of collocated teams
Overlapping of different project time plans
Accessibility
Having to manually check for application logs in Jenkins
Additional problems identified from survey
Entrenched differences/ antagonism between teams
Picking up the right tools
Changing and complex software architectures
Undocumented legacy systems with no enough knowledge
Lack of ownership by various teams
Getting support for open source tools

Table 6.5: Final List of Problems

Chapter 7

Conclusions and Future Work

DevOps is a new phenomenon that gained huge popularity in the past few years. Several major companies like Facebook, Netflix and Yahoo are implementing DevOps [29] [3]. However, there is very little literature available on DevOps, and the studies are often of low quality. This research is thus focused on contributing towards a better understanding of how DevOps works in an organization.

The study has focused on identifying the benefits of implementing DevOps in organizations where agile development is in practice, challenges that are faced by organizations during DevOps adoption, the solutions/ strategies to overcome these challenges, the different DevOps practices, and the problems faced by DevOps teams during continuous integration, deployment and testing.

A mixed methods approach has been employed to conduct the research. An exploratory sequential mixed methods design is followed, where qualitative research will be conducted first, followed by quantitative research. Systematic Literature Review (SLR) and interviews were conducted first, followed by an online survey to validate the qualitative results. Using SLR, answers were found for RQ1, RQ2 and RQ4. It was noted that, the perceived benefits of implementing DevOps in software organizations in general were listed in most of the literature. Only the studies [45] [47] [5] [33] reported on the benefits achieved in practice by DevOps adoption in organizations. The DevOps adoption challenges were elucidated in the studies [42] [22] [28]. Smeds et al. [42] identified the challenges to DevOps adoption in the cloud services domain. But their study was limited to only one organization, which was in the initial stages of DevOps adoption. Jones et al. [22] conducted a case study and identified the DevOps adoption challenges in the embedded systems domain. Smeds et al. [42] mentioned the need to conduct further research by including more organizations for generalizing the DevOps adoption challenges in organizations.

Also, it was observed that no studies addressed the solutions to overcome the DevOps adoption challenges. There is no clear understanding of the DevOps practices in literature. No studies have reported on the problems faced by DevOps teams during continuous integration, deployment and testing. To identify the benefits of adopting DevOps

in organizations where agile development is in practice, interviews were conducted. 12 different practitioners from 11 different software organizations were interviewed, and the DevOps adoption benefits were identified after thematically coding the interview data. Also, several new challenges to DevOps adoption, and solutions to overcome the challenges were identified from the interviews. A clear understanding of the DevOps practices was gained after analysing the interview data. Further, many problems faced by DevOps teams during continuous integration, deployment and testing were identified from the interviews. Thus, all the research questions were answered using the interview data.

The purpose of exploratory sequential design is to validate if the data from a small population sample can be generalized to a larger population sample. Hence, the results from both SLR and interviews were mapped together, and included in the online survey questionnaire. 50 complete responses were considered for statistical analysis. Descriptive statistics was used to analyse the survey responses. Using the survey, the qualitative results are validated, and some new results were also identified. The survey results helped in identifying the benefits of adopting DevOps in organizations, challenges faced by organizations while adopting DevOps, solutions to overcome the challenges, the different practices of DevOps, and the problems faced by DevOps teams during continuous integration, deployment, and testing. Increased collaboration between Dev and Ops teams, frequent software releases, improved quality of software deployments, reduced time to market, and faster feedback were identified as benefits by most of the respondents. Also, the respondents were asked to specify the level of difficulty of each challenge. It was identified that culture and mindset of long timers, legacy software, technical debt, tool integration, lack of proper management structure, and learning curve are the most challenging factors for DevOps adoption. The mostly stated solutions were identified as starting DevOps in small projects, conducting meetings to encourage communication between teams, proper training of employees, clarifying issues immediately, and hiring people with a DevOps mindset. The DevOps practices were identified as continuous integration, testing, delivery, deployment, monitoring and feedback. The problems faced by DevOps teams were identified as configuring infrastructure for the first time, tool related issues, production failures, learning new technologies, and lack of proper communication between teams.

The results were categorized into final lists for each research question, and the mostly stated answers were listed first, followed by the lesser stated ones. Also, some additional benefits, challenges, solutions, practices and problems were specified by the respondents in the open-ended questions included in the survey. Only the most frequently specified answers were included in the final lists. Thus, the challenges identified by Smeds et al. [42] were validated by including more organizations in the interviews and survey. Also, several additional benefits and challenges of DevOps adoption were identified. The final data is valid, as the SLR and interview findings were validated using the survey.

It was learnt that companies should not adopt DevOps just because it is a fancy buzzword, and they need to keep in mind the risks and challenges inherent with DevOps adoption. Based on the organizational structure, capabilities, project type, requirements and needs, the adoption of DevOps tools and practices varies. This finding is in-line with the study [47]. Thus, the study provides researchers and software practitioners a clear idea on the benefits of implementing DevOps in organizations where agile is in practice, the challenges that may occur during DevOps adoption, some solutions to overcome the challenges, the practices of DevOps and the problems faced by DevOps teams during continuous integration, deployment and testing.

7.1 Future Work

It has been observed that there are very few high quality studies related to DevOps, as it is a relatively new concept. There is a lot of scope for empirical work in this domain. In our study, the benefits of adopting DevOps, challenges, solutions, practices and problems were studied in general DevOps adoption scenarios. As the adoption of DevOps tools and practices in organizations is dependent on factors like organizational structure, capabilities, project needs, requirements, and project types, some of the identified results could be contextual. For example, the challenges faced in a small organization adopting DevOps might vary from the challenges faced in a large organization. A case study can be performed to validate the results. The cause effect factors that may impact the results identified can be investigated by conducting an empirical study. For example, the impact of culture on DevOps adoption can be investigated. Also, no studies address the solutions to overcome DevOps adoption challenges, and problems faced by DevOps teams during continuous integration, deployment and testing. We have covered this gap, but more research could be performed in this aspect. Further, it is learnt that for transitioning to DevOps, a lot of changes must be made in an organization. It would be interesting to investigate what changes must be made in an organization for transitioning into a full DevOps state. There is currently no proper research on the tools needed for supporting DevOps. The availability of DevOps tools, and their functioning can be studied.

References

[1] Z. Babar, A. Lapouchnian, and E. Yu. *Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions*, pages 322–337. Springer International Publishing, Cham, 2015.

[2] D. Badampudi, C. Wohlin, and K. Petersen. Experiences from using snowballing and database searches in systematic literature studies. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, EASE '15, pages 17:1–17:10, New York, NY, USA, 2015. ACM.

[3] S. K. Bang, S. Chung, Y. Choh, and M. Dupuis. 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*, RIIT '13, pages 61–62, New York, NY, USA, 2013. ACM.

[4] L. Bass, R. Jeffery, H. Wada, I. Weber, and L. Zhu. Eliciting operations requirements for applications. In *Release Engineering (RELENG), 2013 1st International Workshop on*, pages 5–8, May 2013.

[5] M. Callanan and A. Spillane. Devops: Making it easy to do the right thing. *IEEE Software*, 33(3):53–59, May 2016.

[6] G. G. Claps, R. B. Svensson, and A. Aurum. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology*, 57:21 – 31, 2015.

[7] C. A. Cois, J. Yankel, and A. Connell. Modern devops: Optimizing software development through effective system interactions. In *2014 IEEE International Professional Communication Conference (IPCC)*, pages 1–7, Oct 2014.

[8] J. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2013.

[9] E. Di Nitto, P. Jamshidi, M. Guerriero, I. Spais, and D. A. Tamburri. A software architecture framework for quality-aware devops. In *Proceedings of the 2Nd International Workshop on Quality-Aware DevOps*, QUDOS 2016, pages 12–17, New York, NY, USA, 2016. ACM.

[10] A. Dyck, R. Penners, and H. Lichter. Towards definitions for release engineering and devops. In *Release Engineering (RELENG), 2015 IEEE/ACM 3rd International Workshop on*, pages 3–3, May 2015.

[11] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. Devops. *IEEE Software*, 33(3):94–100, May 2016.

[12] F. Erich, C. Amrit, and M. Daneva. Cooperation between information system development and operations: A literature review. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, pages 69:1–69:1, New York, NY, USA, 2014. ACM.

[13] F. Erich, C. Amrit, and M. Daneva. *A Mapping Study on Cooperation between Information System Development and Operations*, pages 277–280. Springer International Publishing, Cham, 2014.

[14] B. S. Farroha and D. L. Farroha. A framework for managing mission needs, compliance, and trust in the devops environment. In *2014 IEEE Military Communications Conference*, pages 288–293, Oct 2014.

[15] D. G. Feitelson, E. Frachtenberg, and K. L. Beck. Development and deployment at facebook. *IEEE Internet Computing*, 17(4):8–17, July 2013.

[16] B. Fitzgerald and K. Stol. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 2014.

[17] B. Fitzgerald and K.-J. Stol. Continuous software engineering and beyond: Trends and challenges. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, RCoSE 2014, pages 1–9, New York, NY, USA, 2014. ACM.

[18] L. Fitzpatrick and M. Dillon. The business case for devops: a five-year retrospective. *Cutter IT Journal*, 24(8):19 – 27, 2011/08/.

[19] W. Gottesheim. Challenges, benefits and best practices of performance focused devops. In *Proceedings of the 4th International Workshop on Large-Scale Testing*, LT '15, pages 3–3, New York, NY, USA, 2015. ACM.

[20] J. Humble and J. Molesky. Why enterprises must adopt devops to enable continuous delivery. *Cutter IT Journal*, 24(8):6 – 12, 2011/08/.

[21] S. W. Hussaini. Strengthening harmonization of development (dev) and operations (ops) silos in it environment through systems approach. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 178–183, Oct 2014.

[22] S. Jones, J. Noppen, and F. Lettice. Management challenges for devops adoption within uk smes. In *Proceedings of the 2Nd International Workshop on Quality-Aware DevOps*, QUDOS 2016, pages 7–11, New York, NY, USA, 2016. ACM.

[23] K. Kaur, A. Jajoo, and Manisha. Applying agile methodologies in industry projects: Benefits and challenges. In *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on*, pages 832–836, Feb 2015.

[24] N. Kerzazi and B. Adams. Who needs release and devops engineers, and why?

[25] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering, 2007.

[26] T. A. Limoncelli and D. Hughes. Lisa'11 theme—"devops: New challenges, proven values".

[27] Y. Liu, C. Li, and W. Liu. Integrated solution for timely delivery of customer change requests: A case study of using devops approach. *International Journal of U-and E-Service Science and Technology*, 7(2):41–50, 2014.

[28] L. E. Lwakatare, T. Karvonen, T. Sauvola, P. Kuvaja, H. H. Olsson, J. Bosch, and M. Oivo. Towards devops in the embedded systems domain: Why is it so hard? In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 5437–5446, Jan 2016.

[29] L. E. Lwakatare, P. Kuvaja, and M. Oivo. *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings*, chapter Dimensions of DevOps, pages 212–217. Springer International Publishing, Cham, 2015.

[30] M. A. McCarthy, L. M. Herger, S. M. Khan, and B. M. Belgodere. Composable devops: Automated ontology based devops maturity analysis. In *Services Computing (SCC), 2015 IEEE International Conference on*, pages 600–607, June 2015.

[31] S. I. Mohamed. Devops maturity calculator domc-value oriented approach.

[32] S. I. Mohamed. Devops shifting software engineering strategy value based perspective.

[33] K. Nybom, J. Smeds, and I. Porres. *On the Impact of Mixing Responsibilities Between Devs and Ops*, pages 131–143. Springer International Publishing, Cham, 2016.

[34] M. Olszewska and M. Waldén. Devops meets formal modelling in high-criticality complex systems. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, QUDOS 2015, pages 7–12, New York, NY, USA, 2015. ACM.

[35] M. Pai, M. McCulloch, J. D. Gorman, N. Pai, W. Enanoria, G. Kennedy, P. Tharyan, and J. M. Colford, Jr. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *The National medical journal of India*, 17(2):86, 2004.

[36] K. Petersen, S. Vakkalanka, and L. Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1 – 18, 2015.

[37] T. Peuraniemi. Review: Devops, value-driven principles, methodologies and tools. *Data-and Value-Driven Software Engineering with Deep Customer Insight*, page 43.

[38] J. Roche. Adopting devops practices in quality assurance. *Commun. ACM*, 56(11):38–43, Nov. 2013.

[39] G. Rong, H. Zhang, and D. Shao. Cmmi guided process improvement for devops projects: An exploratory case study. In *Proceedings of the International Workshop on Software and Systems Process*, ICSSP '16, pages 76–85, New York, NY, USA, 2016. ACM.

[40] M. Shahin. Architecting for devops and continuous deployment. In *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*, ASWEC ' 15 Vol. II, pages 147–148, New York, NY, USA, 2015. ACM.

[41] F. Shull, J. Singer, and D. I. Sjøberg. *Guide to Advanced Empirical Software Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[42] J. Smeds, K. Nybom, and I. Porres. *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings*, chapter DevOps: A Definition and Perceived Adoption Impediments, pages 166–177. Springer International Publishing, Cham, 2015.

[43] D. Spinellis. Don't install software by hand. *IEEE Software*, 29(4):86–87, July 2012.

[44] D. Spinellis. Being a devops developer. *IEEE Software*, 33(3):4–5, May 2016.

[45] M. Stillwell and J. G. F. Coutinho. A devops approach to integration of software components in an eu research project. In *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, QUDOS 2015, pages 1–6, New York, NY, USA, 2015. ACM.

[46] B. Tessem and J. Iden. Cooperation between developers and operations in software engineering projects. In *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '08, pages 105–108, New York, NY, USA, 2008. ACM.

[47] M. Virmani. Understanding devops bridging the gap from continuous integration to continuous delivery. In *Innovative Computing Technology (INTECH), 2015 Fifth International Conference on*, pages 78–82, May 2015.

[48] A. Wahaballa, O. Wahballa, M. Abdellatief, H. Xiong, and Z. Qin. Toward unified devops model. In *Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on*, pages 211–214, Sept 2015.

[49] J. Waller, N. Ehmke, and W. Hasselbring. Including performance benchmarks into continuous integration to enable devops. *ACM SIGSOFT Software Engineering Notes*, 40(2):4 pp. –, 2015/03/.

[50] J. Wettinger, V. Andrikopoulos, and F. Leymann. Automated capturing and systematic usage of devops knowledge for cloud applications. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 60–65, March 2015.

[51] J. Wettinger, V. Andrikopoulos, and F. Leymann. *On the Move to Meaningful Internet Systems: OTM 2015 Conferences: Confederated International Conferences: CoopIS, ODBASE, and C&TC 2015, Rhodes, Greece, October 26-30, 2015. Proceedings*, chapter Enabling DevOps Collaboration and Continuous Delivery Using Diverse Application Environments, pages 348–358. Springer International Publishing, Cham, 2015.

[52] J. Wettinger, U. Breitenbücher, and F. Leymann. Standards-based devops automation and integration using tosca. In *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on*, pages 59–68, Dec 2014.

[53] J. Wettinger, U. Breitenbücher, and F. Leymann. *Service-Oriented and Cloud Computing: Third European Conference, ESOCC 2014, Manchester, UK, September 2-4, 2014. Proceedings*, chapter DevOpSlang – Bridging the Gap between Development and Operations, pages 108–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[54] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 38:1–38:10, New York, NY, USA, 2014. ACM.

[55] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[56] L. Zhu, L. Bass, and G. Champlin-Scharff. Devops and its practices. *IEEE Software*, 33(3):32–34, May 2016.

Appendix A- Primary Studies for SLR

S1 : 2014, "A Framework for Managing Mission Needs, Compliance and Trust in the DevOps Environment" [14]

S2 : 2015, "A DevOps Approach to Integration of Software Components in an EU Research Project" [45]

S3 : 2014, "A Mapping Study on the Cooperation between Information System Development and Operations" [13]

S4 : 2014, "DevOpSlang - Bridging the Gap between Development and Operations" [53]

S5 : 2015, "Dimensions of DevOps" [29]

S6 : 2014, "Standards-based DevOps Automation and Integration using TOSCA" [52]

S7 : 2014, "Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through Systems approach" [21]

S8 : 2015, "Toward Unified DevOps Model" [48]

S9 : 2015, "Understanding DevOps and Bridging the gap from Continuous Integration to Continuous Delivery" [47]

S10 : 2011, "The Business Case for DevOps: A Five-Year Retrospective" [18]

S11 : 2011, "DevOps: So You Say You Want a Revolution?" [18]

S12 : 2014, "Cooperation Between Software Development and Operations: A Literature Review" [12]

S13 : 2011, "DevOps: New Challenges, Proven Values" [26]

S14 : 2016, "DevOps Making It Easy to Do the Right Thing" [5]

S15 : 2012, "Don't Install Software By Hand" [43]

S16 : 2014, "Integrated Solution for Timely Delivery of Customer Change Requests: A Case Study of Using DevOps Approach" [27]

S17 : 2016, "CMMI Guided Process Improvement for DevOps Projects: An Exploratory Case Study" [39]

S18 : 2015, "On the journey to continuous deployment: Technical and Social challenges along the way" [6]

S19 : 2016, "Being a DevOps Developer" [44]

S20 : 2016, "On the Impact of Mixing Responsibilities Between Dev and Ops" [33]

S21 : 2015, "DevOps: A Definition and Perceived Adoption Impediments" [42]

S22 : 2016, "DevOps" [11]

S23 : 2016, "A Software Architecture Framework for Quality-Aware DevOps" [9]

S24 : 2016, "Management Challenges for DevOps Adoption within UK SMEs" [22]

S25 : 2016, "Towards DevOps in the Embedded Systems Domain: Why is It so Hard?" [28]

S26 : 2014, "Continuous software engineering: A roadmap and agenda" [16]

S27 : 2014, "Continuous Software Engineering and Beyond: Trends and Challenges" [17]

S28 : 2016, "DevOps and Its Practices" [56]

S29 : 2008, "Cooperation between Developers and Operations in Software Engineering Projects" [46]

S30 : 2011, "Why Enterprises Must Adopt DevOps to Enable Continuous Delivery" [20]

S31 : 2015, "Modeling DevOps Deployment Choices Using Process Architecture Design Dimensions" [1]

Appendix B- Interview Questionnaire

1. Can you tell us about your experience in working with DevOps?
2. Can you describe about the methodology your organization followed before adopting DevOps?
3. So, why did your organization decide to adopt DevOps?
4. From what we have learned from literature, there is no specific DevOps process. Rather, it is viewed as a framework and organizations must include the DevOps principles and practices in their processes. What can you say about this?
5. Can you tell us about the principles and practices of DevOps?
6. Let us say, an organization wants to adopt DevOps. What principles and practices do you think it must include in its development process for this purpose?
7. Can you tell us about which practices of DevOps your organization has included?
8. Can you tell us about the benefits of adopting DevOps in organizations?
9. What benefits did your organization achieve after adopting DevOps?
10. Do you think the DevOps adoption process can be challenging for organizations which are implementing it for the first time?
11. Can you describe about the possible challenges that an organization may face during the starting stages of the adoption process?
12. Now, let us say the organization has moved past the initial stages of adoption. What are the possible challenges that await? Can you describe about them in detail?
13. What do you think are the solutions to overcome these challenges? Can you describe about them in detail?
14. Can you tell us about the challenges that your organization came across while adopting DevOps in your experience, if any?

15. Can you describe about what strategies your organization used to overcome the challenges?
16. Also, the DevOps teams must work together for continuous integration, deployment and testing. Can you tell us about the problems the DevOps teams face during continuous integration, deployment and testing?
17. Can you give a few examples from your experience, about the problems that the DevOps teams usually face?
18. Do you have anything more to add?

Appendix C- Survey Questionnaire

Dear Software Engineering Practitioners,

We are Master's students at Blekinge Institute of Technology, Sweden. As part of our Master's Thesis, we are conducting a survey to learn about the adoption of DevOps in software organizations.

This survey won't take more than 15 minutes and can be completed by any software practitioner who has worked with DevOps. There is currently very little research related to DevOps. By answering this survey questionnaire, you can help in contributing to the research pool and in improving the perception of software organizations regarding DevOps.

Your response would be invaluable to our research, and contribute greatly to the DevOps research community. All your responses are strictly confidential, and will only be accessed by the two researchers. Your anonymity will be preserved, and no details about an individual person or organization will be included in the thesis. We would be really grateful if you can forward this questionnaire to your contacts who are also DevOps practitioners, and encourage them to answer.

Thank you in advance.

Best Regards,
Anand and Swetha

Adoption of DevOps in Software Organizations

This survey questionnaire is designed to identify the benefits of adopting DevOps in organizations, challenges faced during adoption, strategies to overcome these challenges, the DevOps practices, and the problems faced by DevOps teams during continuous integration, deployment and testing. Please participate by adding as much detail as you can.

*Required

1. **The type of organization in which you work** *

   Select all that apply
   Tick all that apply.

   - [ ] Consumer-oriented software
   - [ ] Business oriented software
   - [ ] Design and Engineering software
   - [ ] Information display and transaction entry
   - [ ] Operating Systems
   - [ ] Networking/Communications
   - [ ] Device/Peripheral drivers
   - [ ] Support utilities
   - [ ] Middleware and system components
   - [ ] Software Backplanes (e.g. Eclipse)
   - [ ] Servers
   - [ ] Malware
   - [ ] Hardware Control
   - [ ] Embedded Software
   - [ ] Real time control software
   - [ ] Process control software (i.e. air traffic control, industrial process, nuclear plants)
   - [ ] Operations research
   - [ ] Information management and manipulation
   - [ ] Artistic creativity
   - [ ] Scientific software
   - [ ] Artificial Intelligence
   - [ ] Other: .....

2. **Geographical location of your organization** *

   .....

3. **Size of your organization** *
   Mark only one oval.

   - ◯ < 10
   - ◯ 10-50
   - ◯ 50-500
   - ◯ 500-5000
   - ◯ > 5000

4. **Your role in the organization** *

   .....

5. **Work Experience** *
   Enter number of years

   .....

6. **Experience with DevOps** *
   Enter number of years

   .....

7. **Development Methodology in your organization before adopting DevOps** *
   Mark only one oval.

   - ◯ Plan-driven Methodologies
   - ◯ Agile Methodologies
   - ◯ Hybrid Methodology
   - ◯ Other: .....

8. **Identify which of these benefits can be achieved after implementing DevOps in organizations** *

Select all that apply
Tick all that apply.

- [ ] Increased collaboration between Dev and Ops team members
- [ ] Empathy between Dev and Ops teams
- [ ] Improved quality of software deployment
- [ ] Increased software reliability
- [ ] Frequent software releases
- [ ] Improved code quality
- [ ] Faster feedback
- [ ] Reduced communication overhead
- [ ] Reduced time to market
- [ ] Reduced release cycle time
- [ ] Reduced need for re-work
- [ ] Smooth and faster work flow
- [ ] Automation
- [ ] Increased system stability
- [ ] Easier implementation of customer requirements
- [ ] Increased predictability of releases
- [ ] Rapid delivery of products
- [ ] Increased customer satisfaction
- [ ] Minimal manual intervention
- [ ] Balance between cost and quality
- [ ] Increased responsiveness to business needs
- [ ] Increased revenue
- [ ] Increased overall organizational efficiency
- [ ] Breaks down silos
- [ ] Easier tracking
- [ ] Easier detection of failures

9. **Please specify any other benefits of implementing DevOps, which are not listed in the previous question, which you feel are achieved with DevOps** *

.....

.....

.....

10. **Do you think the DevOps adoption process can be challenging for organizations implementing it for the first time?** *

Mark only one oval per row.

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
	◯	◯	◯	◯	◯

11. **Which of the following can be more challenging for an organization while adopting DevOps?** *

Select one in each row
Mark only one oval per row.

	Very challenging	Challenging	Somewhat challenging	Not challenging	Can't specify
1. Lack of proper management structure					
2. No proper training					
3. Geographical distribution of Dev and Ops teams					
4. Integrating new tools with existing company tools					
5. Legacy software					
6. Resistance from employees					
7. Monolithic system architecture					
8. Organizational structure					
9. Hidden complexities and hidden costs					
10. Complexity due to multiple production environments					
11. Designing software architectures to make them work with DevOps					
12. Technical debt					
13. Buzzword perception					
14. Unclear definition and goals of DevOps adoption					
15. Security clearances (when handling sensitive data)					
16. Buy-in from senior management					
17. Buy-in from customer					
18. Fear of change					
19. Fear of failure in managers					
20. Insufficient knowledge on applications and environments (in outsourcing projects)					
21. Making everyone accustomed to a new workflow					
22. Writing custom scripts for automationg tasks					
23. Learning curve					
24. Lack of proper resources to configure environments					
25. Fear of measurement in employees					
26. Culture and mindset of longtimers					
27. Limited visibility of					

production environments in embedded systems					
28. Lack of proper automation tools in embedded systems domain					

12. **Specify any other challenges which are not listed in the previous question, that you think are faced while adopting DevOps** *

13. **Which of the following mitigation strategies can be used to address the DevOps adoption challenges?** *

Select all that apply
Tick all that apply.

- [ ] Better to start DevOps in small projects
- [ ] Estimate adoption costs and risks before adopting DevOps
- [ ] Properly training the employees
- [ ] Conducting short meetings to encourage communication between teams
- [ ] Clarifying issues that arise, as soon as possible
- [ ] Providing a better understanding of business requirements to every employee
- [ ] Use tools like CloudShell and Helion CloudSystem to bridge compatibility gap between legacy tools and new cloud based tools
- [ ] Minimize hand-offs using an open communication channel and good alignment of teams
- [ ] Using a flexible architecture
- [ ] Specify release criteria for every project and document the challenges that pop-up
- [ ] Have a Proof of Concept POC in place
- [ ] Include Quality Assurance people
- [ ] Hire people with a DevOps mindset
- [ ] Document errors and their solutions
- [ ] Make decisions prior to the start of projects
- [ ] Have a proper management structure

14. **Suggest any other mitigation strategies (that are not listed in the previous question) which you think can solve the DevOps adoption challenges** *

15. **What is your opinion on the description of DevOps as- not a specific process, but rather a framework of principles and practices that organizations must include into their development processes?** *

Mark only one oval per row.

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
	◯	◯	◯	◯	◯

16. **Which of the following practices of DevOps should an organization include in its development process?** *

Select all that apply
Tick all that apply.

- [ ] Continuous Planning
- [ ] Continuous Integration
- [ ] Continous Deployment
- [ ] Continuous Delivery
- [ ] Continuous Testing
- [ ] Continuous Monitoring
- [ ] Continuous Feedback

17. **Please mention any other practices of DevOps that are not listed in the previous question** *

.....

.....

.....

.....

18. **Which of these problems might DevOps teams face during Continuous Integration, Deployment and Testing?** *

Select all that apply
Tick all that apply.

- [ ] Lack of proper communication between employees may cause delay in deployments
- [ ] Tool problems (For e.g. Legacy tools won't be compatible with modern tools, finding the right tools)
- [ ] Time plans of the projects may overlap due to the lags in other projects
- [ ] Configuring infrastructure for the first time
- [ ] Learning new technologies
- [ ] Connectivity issues during integration
- [ ] Getting resources for solving the work flow problems
- [ ] Testing environment flakiness
- [ ] Reverting from database changes
- [ ] Production failures
- [ ] Accessibility
- [ ] Having to manually check for application logs in Jenkins
- [ ] Teams might not be collocated

19. **Mention any other problems faced by DevOps teams (that are not listed in the previous question) which you feel are important** *

20. **Can you specify the benefits achieved in your organization after implementing DevOps?** *

21. **Can you state any challenges faced during DevOps adoption in your organization?** *

22. **Please specify the DevOps practices that you have included in your organization** *

23. **Can you specify the problems DevOps teams in your organization face during Continuous Integration, Deployment and Testing?** *

24. **Do you have anything more to add?**

25. **Include your name and email**
    Your details are completely confidential