UNIVERSITY
OF BORÅS

Assessing the Transformational Impact of DevOps in IT Companies and Startups: Analysis of Pre and Post-adoption.

Master's (one year) thesis in Informatics (15 credits)

Joseph Nishimwe

Spring 2024:2024MAGI04

List of tables

List of Figures

List of Abbreviations

DevOps: Development Operations.

IT: Information Technology.

DSDM: Dynamic Systems Development Method.

CI: Continuous Integration.

CD: Continuous Delivery.

AWS: Amazon Web Services.

QA: Quality Assurance.

VCS: Version Control System.

IEEE: Institute of Electrical and Electronics Engineers.

WNKY: Within keyword.

XM: eXtrem Manufacturing

TITLE-ABS-KEY: Title, abstract and keywords.

SLR: Systematic Literature Review

API: Application Programming Interface

SDLC: System Development Life Cycle

SMEs: Small-to- Medium Enterprises

DevSecOps: Development, Security and Operations

Title: Assessing the Transformational Impact of DevOps in IT Companies and Startups: Analysis of Pre-and Post-adoption.

Year: **2024**

Supervisor: **Peter Rittgen**

Examiner: **Rickard Lindgren**

Author: **Joseph NISHIMWE**

Abstract

In the contemporary landscape, software companies face significant pressure to deliver new features and updates to customers rapidly. This urgency is driven by several factors, including the fast pace of technological advancements, evolving customer needs, and intense competition. To achieve swift and frequent software delivery, companies must exhibit flexibility and efficiency throughout the entire software development lifecycle. This is the point where DevOps becomes indispensable. DevOps is a methodology that integrates development and operations teams, facilitating collaboration and dismantling silos to accelerate product delivery. The adoption of DevOps has surged in recent years as companies seek to leverage its benefits. However, implementing DevOps can present challenges. It is essential to comprehend how DevOps functions within an organization prior to its adoption.

This study provides to the researchers and software developers a deep understanding of how DevOps operates within organizations before and after adoption. Key lessons learned include conceptual insights such as the critical role of team collaboration, the impact of cultural shifts on successful DevOps integration, and the importance of aligning DevOps practices with organizational goals. Practically, the research highlights strategies to overcome adoption challenges, such as implementing effective communication frameworks, investing in automated tools, and fostering a learning culture within teams. The study identifies the advantages of DevOps for companies that already utilize the agile model, pinpoints the challenges faced by companies when adopting DevOps, and explores strategies to overcome these obstacles. It also determines the most prevalent DevOps practices and common issues encountered by DevOps teams during various stages of software development, such as integration, testing and deployment.

To accomplish these goals, the researcher employs a combination of qualitative and quantitative research methods. Qualitative research involves collecting non-numerical data through interviews and textual documents, while quantitative research entails gathering numerical data via surveys and experiments. The research process begins with a systematic literature review to assess existing studies on the benefits, challenges, and practices of DevOps. Following this, the researcher conducts interviews with DevOps team members to gather detailed insights and strategies for overcoming adoption challenges. Subsequently, the researcher analyzes the aggregated data from all sources to derive comprehensive insights into DevOps practices and challenges, providing actionable recommendations for organizations seeking to adopt and optimize DevOps methodologies.

Keywords: DevOps, Impact, Development, IT, Startups, Adoption, Challenges, Software Development.

Table of Contents

1. INTRODUCTION

This part includes the background of the thesis, a short description, related research, problem discussion, the purpose, and the aim of this research study, including the scope and research contributions.

In today's ever-changing industry of information technology, many organizations are turning to DevOps as a crucial strategy. They aim to improve their software development and IT operations, making the delivery of software quicker and more efficient [1]. This study delves into the effects of DevOps, looking at how things change in both well-established IT companies and startups when they start using DevOps principles. I am trying to understand the impact, comparing how things were before they adopted DevOps and how they have evolved since. In simple terms, DevOps, which stands for development operations, was created in the world of software engineering to connect development and operations teams. Over the years, a new way of developing software (like agile methodologies) became popular. Traditionally, developers and operations teams worked separately. After developers finished writing codes, they would hand it over to the operations team [2].

Consequently, many enterprises are facing with shared challenges, like the need for a more quick and frequent software delivery [3]. The era where customers patiently waited months for a product release, has given way to a demand for quicker responses to offer feedback afterward. Continuous software releases are now crucial to enable customers to provide ongoing feedback [3]. Interestingly companies like Meta regularly deploy software to their customers [4]. Although continuous software delivery provides a competitive edge, achieving it involves overcoming several challenges, both technical and non-technical [4].To surmount these challenges, companies must maintain an agile approach throughout the entire software development lifecycle [5]. Agile software development has gained popularity in many enterprises due to its numerous benefits, including reduced development time, increased project success rates, lower development costs, and heightened customer satisfaction. Agile methods enhance responsiveness and ensure that stakeholders' needs are met promptly. These methods foster collaboration among development teams. A few examples of agile methodologies include ScrumBan, DSDM, Lean, Crystal, XM, Nexus, and many others [6].

DevOps addresses drawbacks in agile methodologies by providing tools and practices for a smooth development life cycle [7]. Its implementation aims to eliminate silos among IT departments, fostering collaboration and reliable workflows. Breaking down silos enables quicker production releases and facilitates problem detection. However, achieving transparency requires organizational and cultural changes. Challenges include selecting the right tools, as the development potentially impacts efficiency. Low trust in the automation process poses additional difficulties for efficient implementation. Challenges arise when implementing various aspects of the software development life cycle. Adopting DevOps is not easy and might not be smooth due to the changes needed to be practical in the core of the followed methodologies. This thesis aims to describe DevOps practices and tools, the benefits of implementing its methodologies by addressing and identifying the current state of DevOps in the IT industry. This, in turn, minimizes time spent and helps avoid possible challenges. The primary objective of this thesis is to explore the methodologies and tools essential for implementing DevOps, while reinforcing its core principles to advance development, drawing insights from the historical performance data of various IT organizations.

1.1. Related Research

The increasing adoption of DevOps practices has motivated researchers to explore its various facets, from critical implementation challenges to the integration of security practices. Below is a summary of key studies and their contributions to understanding and advancing DevOps. In their Master's thesis, Arvedahl and Åkersten et al. [11], explored the criticality and impact of DevOps practices through thematic coding and surveys. Their two iterations of thematic coding grouped codes into broader categories such as "DevOps" "practices" and were verified through a survey sent to practitioners and achieved a 100% responses rate. This ensured a robust result reflecting practitioners' insight into the challenges and benefits of DevOps. Similarly, Kuvaja et al. [22],investigated the relationship between DevOps and agile, lean, and continuous development practices, their research highlighted how DevOps extends the agile philosophy, offering better alignment between development and operations and addressing gaps in the software development life cycle (SDLC) [21].

Škurla et al. [12] conducted a systematic review to identify methods and practices for integrating security into DevOps. This research focused on addressing security concerns within the continuous delivery pipeline and provided a comprehensive review of challenges, best practices, and solutions in DevOps security. Their work emphasized the need for secured software delivery pipelines and informed future research directions in the field of DevSecOps. Building on this, Sohrab & Bao et al. [15], explored a workflow for integrating security into the DevOps model. Their work focused on automating security activities within a continuous delivery pipeline by integrating security tools into GitHub Actions. Their proposed solution aimed to enhance security practices in software delivery and reduce vulnerabilities. Similarly, Jabbari et al.[1], provided a foundational understanding of DevOps, emphasizing how security practices can effectively woven into the DevOps framework.

Researchers Pelmas and Lejon et al.[13], investigated the challenges of adopting DevOps from a Scrum perspective. Through a multiple case study design, their research identified patterns in technical challenges such as technical debt, microservice management, and the interface between development and operations teams. By contrasting and comparing different cases they contributed to understanding how organizations implement DevOps in practice and the technical hurdles they face. Nybom, Porres and Smeds[25], added to this discussion by identifying perceived impediments to DevOps adoption. Their study outlined the barriers organizations face when transitioning to DevOps and offered strategies for overcoming these challenges. Similarly, Sharma[26], proposed a guide to adopting DevOps in multi-speed IT enterprise, focusing on aligning organizational culture and technical practices to achieve successful DevOps implementation.

Amaradri & Nutalapati et al. [14], identified the benefits, challenges, and practices associated with DevOps adoption in organizations which already practicing agile methodologies. Their study provided a comprehensive list of practices for successful DevOps implementation, such as continuous integration, testing, and deployment, and highlighted the challenges faced during deployment and testing phases. Their work served as a guide for researchers and practitioners aiming to understand DevOps functioning. Virmani[5], explored the transition from continuous integration to continuous delivery in DevOps environments. Their findings underscored the benefits of DevOps in achieving faster delivery times, improved collaboration, and higher software quality. Laukkanen and Itkonen[37], also examined the causes and solutions when adopting continuous delivery, providing practical insights into how organizations can overcome challenges during this transition.

Hüttermann et al. [16], provided a foundational perspective on DevOps, emphasizing its roots in combining development (Dev) and IT operations (Ops) to streamline software creation and delivery. They highlighted how Agile practices play a critical role in breaking down silos and fostering collaboration between teams. Similarly, Randall B. et al. [17], traced the origins of software practices, linking the evolution of DevOps to the software crisis that demanded better reliability and methodologies. Their study connected historical challenges with modern DevOps solutions emphasizing the importance of iterative and collaborative approaches. Kim [29] explored how organizations can achieve world-class agility, reliability, and security by adopting DevOps practices. Their book outlined strategies for overcoming traditional operational challenges and achieving a balance between innovation and stability.

Researchers Brian and Stol[63], proposed a roadmap for continuous software engineering, addressing the critical role of DevOps in enabling continuous integration, deployment, and delivery, their work emphasized how DevOps facilitates a seamless transition from development to operations, ensuring high-quality software delivery. Floris et al. [58] and Bjørnar and Jon [60], discussed the cooperation between development and operations teams, emphasizing the cultural and technical changes required to implement DevOps successfully. Their findings pointed on the importance of fostering collaboration and aligning team objectives to achieve DevOps success. Kristian and Smeds [59] discussed the implications of mixing responsibilities between development and operations, highlighting the benefits and risks associated with this practice. Their work reinforced the importance of clarity in roles and responsibilities within DevOps environments. Matt and Alexandra[57],emphasized the importance of making it easy for teams to “do the right thing” in DevOps environments. Their work highlighted how organizational policies and tools can be designed to promote good practices effortlessly.

The integration of DevOps into information systems (IS) has highlighted its role in bridging the gap between organizational objectives and technological execution. Markus and Keil [64], explored the interplay between technical systems and organizational alignment, offering frameworks that can be extended to DevOps practices. Their work emphasized that successful adoption of IT practices (including todays DevOps practices) depends on aligning them with business strategies to create values. Another pivotal contribution comes from Orlikowski[65], who introduced the concept of “technology-in-practice,” emphasizing how the enactment of technology is shaped by human agency and organizational structures. In the context of DevOps, this framework suggests that the success of DevOps practices is highly contingent on how they are implemented and adapted with specific IS organizational contexts.

The information systems (IS) literature has consistently highlighted agility as a key driver of competitive advantage. Sambamurthy et al [66], proposed the concept of digital options, emphasizing how IT enables organizations to respond to rapidly changing environments. This theoretical lens aligns closely with DevOps principles, which aim to enable rapid delivery of software through iterative development and deployment processes. The work of Fitzgerald and Stol[67],on continuous software engineering provides another bridge between DevOps and IS. They conceptualized continuous delivery as part of a broader ecosystem that integrates with information systems to enable continuous value delivery to customers.

1.2. Problem Discussion

The rapid evolution of software development practices has driven organizations to seek more efficient and collaborative approaches to deliver high-quality software, leading to the emergence of DevOps as a transformative methodology. DevOps integrates development (Dev) and operations (Ops) to improve collaboration, automate workflows, and enable continuous delivery. Despite its potential, many organizations struggle with inconsistent adoption and fail to realize its promised benefits due to various socio-technical and organizational barriers. Traditional practices have often been siloed, causing inefficiencies and delays that DevOps seeks to overcome by fostering a culture of share responsibility and integrating tools for automation. However, a growing disconnect exists between theoretical models and real-world implementation, as organizations face challenges such as a lack of clarity on best practices, culture resistance, and the difficulty of integrating security into DevOps pipelines, particularly with the rise of DevSecOps. Furthermore, existing research predominantly focuses on large organizations, leaving gaps in understanding DevOps adoption in small-to-medium enterprises (SMEs), multi-speed IT environments, and regulated industries.

These challenges are compounded by limited studies analyzing which practices are critical in different contexts, fragmented research on integrating security practices, and insufficient attention to the socio-technical dimensions of DevOps. Addressing these issues is essential for business aiming to leverage DevOps for digital transformation, as failure to do so leads to increased costs, reduced efficiency, and compromised software quality. This thesis seeks to bridge these gaps by exploring the criticality of DevOps practices investigating socio-technical barriers and their possible solutions before and after its adoption, and examining security integration. Through a comprehensive approach, this study aims to provide actionable insights for practitioners and advance academic understanding, guiding IT organizations and startups in overcoming adoption challenges and achieving sustainable and high-quality software delivery.

1.2.1. Benefits of DevOps Implementation

DevOps offers numerous benefits to organizations, including faster release cycles, improved team collaboration, and enhanced software quality, for established IT companies, DevOps enables streamlined workflows by dismantling silos between development and operations teams, fostering innovation, and facilitating rapid responses to market demands. These benefits are supported by studies that emphasize the alignment of team objectives with organizational goals and the acceleration of software delivery timelines[8]. Startups, despite their resource constraints, stand to gain from the agility and scalability offered by DevOps practices. This is particularly advantageous in a dynamic and a fast-paced environment[10],where adaptability is crucial.

1.2.2. Challenges in Adopting DevOps

Resistance to change is one of the most pervasive challenges faced by established IT companies during DevOps adoption. Employees adapted to traditional methods often perceive the shift to DevOps as disruptive, leading to a doubtful mindset about the value of new DevOps practices. Startups face a different set of challenges, primarily revolving around resource constraints and the need for rapid implementation. Limited budgets and team sizes can hinder access to essential tools and technologies required for DevOps. Furthermore, the fluid nature of startups, with frequently shifting priorities, complicates the establishment of consistent processes [10]. However, startups can leverage their inherit flexibility to adopt DevOps incrementally, ensuring that each stage aligns with their immediate goals.

1.2.3. Strategies for Overcoming Challenges

To address resistance to change in established organizations, effective communication is critical. Leadership must engage employees at all levels, providing clear explanations of the benefits and offering training programs to ease the transition [9]. Establishing pilot projects to demonstrate quick wins can also help build confidence in DevOps practices. Additionally, fostering a culture of continuous learning and feedback ensures that teams remain adaptable and committed to the transformation.

1.2.4. Selection of Tools and Technologies

Choosing the right tools is a critical factor in the success of DevOps adoption. Poor decisions in tooling can lead to inefficiencies and resistance among team members, undermining the potential benefits of DevOps. Organizations must evaluate tools based on their compatibility with existing technology stacks, team expertise, and budget constraints[7]. Conducting pilot tests and soliciting feedback from team members can ensure that tools are effectively integrated into workflows enhancing productivity and minimizing resistance.

1.2.5. Implications for Future Strategies

By identifying and addressing the challenges associated with DevOps adoption, this thesis contributes valuable insights to organizations seeking to embrace DevOps. The findings emphasize the importance of aligning cultural transformation with technical implementation, equipping organizations with practical strategies for navigating the complexities of DevOps integration. Ultimately, these insights enable organizations to unlock the full potential of DevOps, driving innovation and competitiveness in an ever-evolving technological landscap

1.3. Purpose and Research questions

1.3.1. Purpose

The purpose of this study is to comprehensively asses the transformational impact of DevOps in both IT companies and startups through an analysis of before and after its adoption. This research aims to contribute valuable insights into the specific changes, challenges, and benefits that organizations experience during the DevOps adoption journey. By understanding the nuances of this transformation, the study seeks to provide practical knowledge that can guide future implementations and inform decision-making process in IT environments.

1.3.2. Research questions:

The aim of this thesis is to contribute to the scientific research gap by responding to the following research questions:

RQ1: What are the conceptual and practical benefits of integrating DevOps in organizations?

This question explores both the theoretical value and the tangible advantages of implementing DevOps within organizations. It seeks to uncover how organizations can enhance productivity, reduce lead times, and foster collaboration as documented in studies and reflected in real-world practices.

RQ2: What are the most significant challenges organizations faces when adopting DevOps

This question examines the key obstacles hindering DevOps adoption, ranging from cultural resistance and skills gaps to technical integration difficulties. By identifying these barriers, the research can inform both academic discussions and practical implementation strategies.

RQ3: How can organizations effectively overcome the challenges associated with DevOps adoption?

This question focuses on actionable solutions and strategies, such as investing in training, developing clear transition plans, and leveraging tools to address challenges identified in RQ2.It aims to provide a roadmap for organizations embarking on DevOps adoption.

RQ4: What practices and principles should organizations adopt to ensure the success of DevOps integration?

This question identifies the essential practices and principles for organizations to follow during and after DevOps adoption. It aims to establish a framework of best practices, informed by existing literature and empirical findings, for sustainable and effective DevOps integration.

1.4. Scope and limitations

The research is limited to Swedish IT software development organizations within different branches, investigating employees' thoughts and opinions within DevOps adoption and integration in their daily activities. A crucial aspect of this research is the analysis, which aims to assess how the impact of DevOps adoption differs before and after being established in IT companies and startups. The study will also take a long-term perspective, examining how the transformative impact of DevOps evolves over time. However, the research does have some limitations. Time constraints may limit the depth of understanding, especially when looking at the long-term effects of DevOps adoption. Additionally, resources availability, particularly in startups, may pose challenges for conducting through survey investigations and interviews. It's important to note that findings from this research might not be applicable to the entire IT industry due to specific organizational contexts. The dynamic nature of technology and external factors like economic conditions may influence the outcomes observed in the organizations under study. Despite these challenges, the research seeks to provide valuable insights into the complex challenges, benefits, and strategies associated with the adoption of DevOps across diverse organizational contexts.

1.5. Research Contributions

This thesis examines the transformational impact of DevOps practices on IT companies and startups, focusing on the significant shifts observed before and after adoption. It offers a comprehensive framework to evaluate the criticality and influence of key DevOps practices, such as continuous integration, delivery, and deployment, allowing organizations to identify and prioritize strategies that enhance collaboration, automation, and software quality. The research highlights socio-technical challenges, including cultural resistance, skills deficiencies, and organization silos, and proposes actionable recommendations to address these issues, fostering a culture of shared responsibility and improved collaboration.

The study reveals how DevOps serves as a catalyst for innovation, enabling faster software products delivery to the market while enhancing team collaboration and communication across development and operations. A critical evaluation of DevOps' role in improving code quality and reducing system downtime reveals its profound influence on customer satisfaction and overall organizational productivity. By investigating the intersection of DevOps and modern trends, such as cloud computing and microservices architecture, the research underscores the significance of technology alignment with business objectives. Moreover, the thesis addresses emerging issues like security vulnerabilities and compliance challenges, proposing frameworks for continuous risk assessment and governance. It also considers the specific needs of startups, exploring how limited resources and high uncertainty can affect DevOps implementation strategies and success rates.

By analyzing DevOps adoption in diverse organizational contexts ranging from small and medium-sized enterprises (SMEs) to startups and regulated industries, the thesis offers insights into the adaptability and scalability of DevOps methodologies, and suggests future research directions to explore these aspects further. Additionally, the study provides practical guidelines for IT companies and startups to navigate pre-adoption challenges and optimize their post-adoption strategies, ensuring sustainable improvements in software delivery and organizational efficiency. These contributions aim to enrich academic discourse, guide DevOps experts, and inspire ongoing advancements in the dynamic field of DevOps.

1.6. Thesis Structure

This master's thesis will be structured in the following chapters:

Part 2: Theoretical concepts and terminologies related to DevOps are discussed and explained.

Part 3: Introduce the selected research method relevant to the study, qualitative method and quantitative method.

Part 4: This part of this research involves data analysis and the results

Part 5: The last part of this research concludes the research and exposes further implication of the study.

This part of thesis consists of introducing the theoretical background of DevOps, a short description of core principles and different terminologies used in software development by DevOps experts.

2.THEORY

The theoretical foundation of this thesis is rooted in understanding the principles, frameworks, and the methodologies underlying DevOps and its integration into software development and operations. By situating DevOps within a broader theoretical framework, this section examines its fundamental concepts, the cultural and technical dimensions required for its successful adoption, and its relationship with organizational transformation, particularly in IT companies and startups.

2.1. Defining DevOps and its core principles

DevOps, a combination of "Development" and "Operations" represents a cultural and technical approach aimed at improving collaboration between traditionally siloed teams[22]. At its core, DevOps emphasizes several key principles:

- Collaboration and Communication: Breaking down silos between development, operations, and other stakeholders to unsure seamless workflows and collective ownership of outcomes [9].
- Automation: Leveraging tools to automate repetitive processes, such as testing, integration, and deployment, to accelerate delivery cycles and reduce human errors[37].
- Continuous Delivery (CD) and Continuous Integration (CI): Practices that focus on delivery incremental updates to software in reliable and efficient manner[37].
- Feedback Loops: Implementing mechanisms to gather and incorporate feedback continuously, enabling iterative improvement and alignment with user needs
-

These principles underpin the operational goals of DevOps, including faster software delivery, improved reliability, and greater adaptability to market demands. The theoretical emphasis on collaboration and automation highlights DevOps role as both a cultural shift and a technical methodology.

2.2. The role of culture in DevOps Adoption

Cultural transformation is a cornerstone of DevOps implementation. DevOps challenges the hierarchical, compartmentalized structures prevalent in traditional IT organizations by promoting shared responsibility and open communication[9]. A successful DevOps culture fosters trust, accountability, and a willingness to experiment and learn from failures. However, resistance to cultural change is a well-documented barrier, particularly in established IT companies. Employees accustomed to conventional process may view DevOps as disruptive, leading to skepticism and reluctance to adopt new practices. Theories on organizational change, suggest that leadership plays a pivotal role in overcoming resistance. By articulating a clear vision, involving employees in the transformation process, and celebrating early wins, leaders can build momentum for cultural change. This aligns with existing research emphasizing the importance of leadership in driving successful DevOps adoption.

2.3. Technical Dimensions: Tooling and Automation

The technical implementation of DevOps is typically based on the integration of tools and automation to streamline workflows. Tooling choices are critical for achieving DevOps objectives, particularly in areas such as CI/CD pipelines, infrastructure as code (IaC), and monitoring. However, selecting appropriate tools is a complex process influenced by factors like the existing technology stack, team expertise, and budget constraints[37].For example, startups often gravitate toward cost-effective and open-source tools to mitigate resource limitations. The theory of socio-technical systems provides a useful lens to understand interplay between technical tools and human workflows in DevOps. This theory posits that technological solutions must align with organizational structures and cultural practices to achieve optimal performance. In the context of DevOps, this means that tooling decisions should be informed by the unique needs and capabilities of the organization, ensuring a harmonious integration of technology and teamwork.

2.4. DevOps and Organizational Transformation

Adopting DevOps is not merely a process change but a fundamental organizational transformation. This transformation can be conceptualized through the lens of change management theories, which emphasize the need for aligning people, processes, and technology. In established IT companies, the shift to DevOps requires rethinking traditional workflows and fostering a culture of experimentation and learning. This aligns with theories of learning which organizations advocate for continuous improvement and adaptability as key drivers of organizational success. Conversely, startups, due to their dynamic and resource-constrained nature, must adopt an agile approach to DevOps implementation, prioritizing incremental changes that align with their immediate goals.

2.5. Challenges and Opportunities in DevOps implementation

The theoretical framework for understanding the challenges of DevOps adoption is grounded in organization behavior and innovation diffusion theories. Resistance to change, as noted in the literature, often stems from uncertainty about the benefits of new practices and fear of disrupting established routines[9]. On the other hand, the opportunities presented by DevOps are substantial. By enabling faster delivery cycles, improving product quality, and fostering a culture of collaboration, DevOps aligns with the principles of agile development and lean thinking. These methodologies emphasize iterative progress, customer-centricity, and waste reduction, reinforcing the theoretical synergy between DevOps and modern software development practices.

2.6. Theoretical Contributions of this thesis

This thesis builds on the existing theoretical foundation by exploring the intersection of cultural and technical dimensions in DevOps adoption. It provides a nuanced understanding of the challenges faced by IT companies and startups. Contributing to the broader discourse on the organization transformation and innovation.

By integrating insights from change management, socio-technical systems theory, and resource-based perspectives, the research offers a comprehensive framework for navigating the complexities of DevOps implementation. In conclusion, the theoretical framework established in this thesis highlights the multifaceted nature of DevOps adoption, encompassing cultural, technical, and organizational dimensions. By grounding the research in well-established theories and aligning with empirical insights, this section lays the foundation for addressing the practical challenges and opportunities explored in subsequent sections of the thesis.

2.7. DevOps terminologies

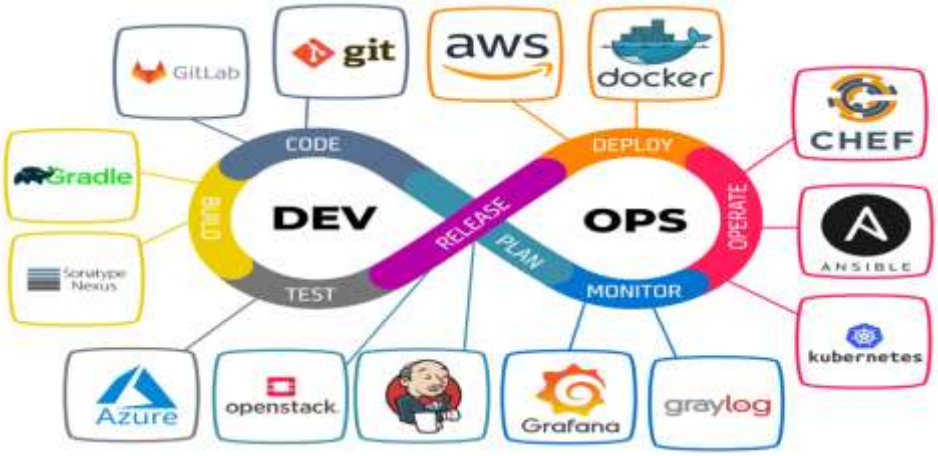


Figure 1: DevOps steps accessed on *[24]*

DevOps is a way to bridge the gap between development and operations, making them work together smoothly. Even though the goal of DevOps is clear, it has been a bit tricky to fully define it [25]. DevOps is a rather new idea, based on agile and lean principles, but it's distinct from them. Lean principles aim to cut down on waste and make the delivery time for new software shorter. Reducing waste might involve doing tests earlier in the development process, a concept known as "shifting left" in the toolchain. It's also suggested to use value stream mapping to see the whole development process and find ways to reduce waste [26]. On the other hand, agile principles focus on bringing the development team closer to the customer, while DevOps looks for cooperation between operations and development. DevOps provides a systematic and complete view of the entire development process where as agile software development might concentrate on specific parts of the DevOps chain. Despite these differences, both agile and DevOps share the goal of rapidly developing software, collecting information for frequent releases. The success of adopting DevOps often requires incorporating agile principles [22]. Before the era of agile, development and deployment were scheduled at a specific date, and operations knew when the software would be ready for testing. Agile software development has increased the speed of creating software. However, the pace of testing remained the same. Causing a blockage in the system due to sub-optimization. While continuous integration became possible and was sent daily to the Operations team, they couldn't keep up because continuous deployment was lacking and testing was not automated [26].

These DevOps terminologies form the basis for effective collaboration, automation, and the implementation of DevOps practices in software development and IT operations, understanding these terms is important for professionals working in a DevOps environment. Proficiency in DevOps terms is essential for integrating and implementing automation tools seamlessly, enabling efficient problem-solving, and aligning with DevOps culture and best practices. Additionally, a solid knowledge of DevOps terminologies supports continuous learning, and informed decision-making, and enhances the onboarding process for new team members. In essence, a shared understanding of DevOps terminologies is fundamental for creating a collaborative, efficient, and adaptable working environment that aligns with the principles and goals of DevOps.

2.7. Agile Software Development

The agile development story began in 2001 when a group of software consultants agreed to the manifesto for Agile software development [31]. Traditional software development methods, such as the waterfall method, didn't consider the unpredictable nature of the development environment [32]. These methods focused more step-by-step work and preparing requirements for each part of the design before moving to the next. Testing teams only got involved in the final phase, and problems remained hidden until then. This approach wasn't adaptable to the market and business changes. Agile is not a methodology but a mindset. It represents a way of working that incorporates the principles of collaboration, systems thinking [33]. In contrast, Agile was introduced and became a leading approach. Agile prioritizes during development. Agile reduces software development cycles by allowing finished products to be shipped faster.

This is beneficial for companies as products reach users quickly, and feedback can collected to refine the development process based on user needs. Recent methodologies like Kanban and scrum introduced in recent years, can be seen as extensions of Agile. Kanban focuses on maximizing efficiency and reducing work in progress based on each team's capabilities. The scrum method aims to create short development runs called sprints, evaluating and adjusting the software at each iteration. All these methods have been utilized by the industry for years, forming the foundation of software delivery life cycles and playing a significant role in the development process of software companies. Additionally, agile is an iterative and incremental approach to software development and project management that emphasizes flexibility, collaboration, and customer satisfaction. The agile methodology was introduced as a response to the limitations of traditional methodologies such as the waterfall model, which followed a linear and sequential process.

2.8. Scrum

In 1993, the first scrum team came into existence when Jeff Sutherland took an innovative approach to critical software development. The idea gained traction, and by 1995, Jeff Sutherland and ken Schwaber formalized the scrum framework in a conference paper, making it public. Interestingly, the members of that initial scrum, along with Sunderland and Schwaber, played a pivotal role in creating the Agile Manifesto. To understand Scrum, you need to delve into its values, the roles within a Scrum team, and the events that shape its workflow [34]. So, what exactly is Scrum? It's a method designed to facilitate Agile development, providing a framework with steps to guide the software and product development process. The primary goals are to enhance development speed, support the creation of shareholder value, and foster effective communication at all levels [35]. In the scrum framework, a team is self-managing, holding authority and responsibility for planning, scheduling, and decision-making related to their work.

2.9. CI/CD Pipeline

The CI/CD pipeline, consisting of Continuous Integration (CI) and Continuous Delivery (CD) is essentially a pipeline based on Agile principles designed to optimize the software development Life Cycle (SDLC). It's like a well-structured process ensuring that different stages of software development can happen continuously. Lately, this CI/CD pipeline has gained significance in software development, making the SDLC more adaptable, efficient, and quicker. Notably, with the emergence of cloud solutions and major cloud providers introducing command-line tools for deploying applications, continuous deployment has become the final step in the pipeline. Let's delve into the details of each phase.

2.10. Continuous Integration (CI)

It's a process that empowers software developers to integrate new code into the original repository and share it across the workflow. The automation in CI also detects errors early in the process, allowing immediate resolution in CI also detects errors(commit)early in the process, allowing immediate resolution when an issue arises [37]. When new codes get merged with the existing repository, a new version is activated. After the build is complete, automated tests are run against it to ensure everything works smoothly. The integration is continuous, signified by the "C" in CI. Every time developers push changes to the repository, the build automatically verifies the code, enabling development teams to sport and address issues early, top used tools are: Jenkins, AWS Code Pipeline, CircleCI and many more.

2.11. Continuous Delivery (CD)

Continuous Delivery (CD) draws inspiration from the concepts of distributors and deliveries. It's a software engineering approach centered on producing software in short cycles, allowing publishers to easily test, build, and deploy on a regular basis. This is not only facilitates agility but also reduces costs and risks associated with changes. CD is viewed as an extension of Continuous Integration (CI), and involves regular code upgrades to ensure quality assurance (QA) [38]. The CD phase is responsible for the automatic distribution of Integrated code from the development stage to the production stage[37]. It not only automates sending the integrated code but also ensures error-free and timely deliveries. This phase enables developers to consistently incorporate new code into the main branch, checking code quality and ensuring that a functional build can be released into the production environment, the most used tools are , GitLab, Azure DevOps, Maven and many more.

2.12. Continuous Deployment

Continuous Deployment is another aspect of the CI/CD process, it revolutionizes codebase management, making significant changes manageable and feasible during regular working hours. Continuous deployment goes beyond continuous delivery [39]. By aiming to accelerate the delivery of changes to end-users. This involves automating the deployment process, sometimes achieving hundreds of deploys in a day. Prominent tech giant like Meta (Facebook) and Flickr have embraced this method, leveraging Software as a Service solutions and API-Driven [40].

2.13. DevSecOps

In the conventional software development approach, ensuring the security of a product typically falls under the range of a separate security team. DevOps, driven by the need for speed and collaboration between development and operations, challenges the effectiveness of keeping security efforts isolated. Recognizing this gap, the concept of DevSecOps emerged as an enhancement to DevOps, aiming to integrate additional security measures into its properties [41]. The essence of DevSecOps lies in the demand for secure outputs from the DevOps processes poses challenges, Firstly, the availability of various toolsets for DevSecOps introduces inconsistency across different organizational implementations. Operations teams may be proficient in different programming languages, leading to a lack of standardized DevSecOps implementation. This diversity results in differing opinions between operation teams and other parts of the organization. Additionally, the introduction of new DevOps security tools necessitates thorough testing to ensure the overall security of the organization [42].

2.14. Version Control

In DevOps, version control is a critical practice ensuring systematic management and tracking of changes throughout the software development lifecycle. It plays a role in collaborative development, allowing multiple developers to work concurrently without conflicts, thanks to version control systems (VCS). A notable VCS is Git, extensively covered in “Pro Git “by Scott Chacon and Ben Straub et al.,[43], a comprehensive guide delving into Git’s concepts and practical implementations. Version control maintains a detailed history of changes, aiding in bug tracking and providing the ability to return to previous versions, as emphasized in “version control with Git” by John Loeliger and Matthew McCullough et al.,[44].The ability to create branches for independent work and seamlessly merge changes back into the main codebase is a core feature of version control, discussed in these references. For a pragmatic understanding of Git in real-world scenarios, “Pragmatic Version Control Using Git” by Travis Swicegood et al., [45], serves as a valuable guide. These references collectively provide a solid foundation for understanding version control in DevOps, emphasizing widely-used systems like Git and practical implementation scenarios.

2.15. Containerization

Containerization in DevOps refers to the encapsulation of an application and its dependencies into a lightweight, isolated unit called a container. Containers provide consistency across different environments and facilitate efficient deployment and scaling. The most widely used containerization platform is Docker, which simplifies the packaging and distribution of applications [46].

The methodology part is initiated by introducing the selected research method relevant to the study. The choice of this method is grounded in its appropriateness for investigating the research area in alignment with the study's purpose and research questions. The chapter then outlines the procedures employed for data collection and analysis. A profile of the selected respondents, SLR, and Interviewees is presented, accompanied by an exploration of the ethical considerations in the research. This part concludes with a reflective analysis of the chosen method. To fulfill its purpose, the research adopts a mix of qualitative and quantitative approaches.

3. METHOD

A mixed methods approach has been used for this research, which includes a qualitative, and quantitative method[51]. The main difference between qualitative and quantitative methods is that qualitative methods use words and open-ended questions, while quantitative methods use numbers and closed-ended questions. Qualitative methods are flexible, involving open-ended questions, with data collected in text or audio-visual formats and analyzed based on themes and patterns. Examples include grounded theory analysis, interviews, and literature reviews. Quantitative methods, on the other hand, are more structured, using closed-ended questions, with numeric data that is statistically analyzed such as in experiments and surveys [51].

The mixed methods approach combines both qualitative and quantitative data, making it useful for generalizing findings to a population while also providing detailed insights into a concept. This approach can be time-consuming because it requires the collection and analysis of both types of data. The reason for choosing this method is that it yields more reliable and valid results than using either qualitative or quantitative methods alone [51]. In this research, The researcher used a Systematic Literature Review (SLR) and Interviews for qualitative data, and an online questionnaire survey for quantitative data. Research questions (RQs) 1, 2,3 and 4 are addressed through SLR. On the other hand, all research questions (RQs 1,2,3, and 4) are addressed using interviews and surveys.

3.1. Reasons for using a mixed research method

This research aims to identify all potential challenges organizations might face when adopting DevOps, both at the beginning and during later stages, and to provide solutions for overcoming these challenges. This research also seeks to identify the practices organizations include as part of the DevOps process, the practical benefits they achieve from adopting DevOps, and the issues faced by DevOps teams during various activities. Given that DevOps is a relatively new concept with limited studies available, few quality studies are providing a solid understanding of it. Therefore, there is a need to offer comprehensive and reliable results, which is why I have used a mixed methods approach.

I employed an exploratory sequential mixed methods design. In this process, I first collected qualitative data, analyzed it, and then used the results in the quantitative phase. The goal of this approach is to determine if the findings from a small sample can be generalized to a larger population. Initially, I conducted a Systematic Literature Review, followed by interviews. Based on the qualitative findings, I have then designed a survey questionnaire to collect quantitative data. By using both qualitative and quantitative methods, I can triangulate the data, thereby validating the research.

3.2. Qualitative Research Methods: SLR and Interviews

3.2.1. Systematic Literature Review (SLR)

I have performed an exhaustive review of existing research to determine the impact of DevOps before and after its adoption in IT companies, the difficulties encountered during its adoption, and different associated methods and principles by following guidelines provided by Kitchenham [52]. According to kitchenham et al, SLR is a method for identifying, evaluating, and interpreting all research pertinent to a specific question or area of interest. This type of review is useful for summarizing all evidence related to a topic, finding gaps in current studies, and laying the groundwork for future research activities. It offers an unbiased comprehensive summary of available evidence. [52]. Steps of SLR typically include defining the topic, setting search parameters finding and analyzing evidence, and integrating findings.

I have chosen this method to collect reliable data. Findings helped to identify the advantages of DevOps implementation, adoption challenges, and practices. Results are validated by further interviews and surveys. SLR answers RQ1, RQ2, RQ3 and RQ4. Alternative qualitative methods like systematic mapping studies and traditional reviews exist and share similarities with the systematic review method, but they differ in data analysis approaches and do not require quality assessments. Mapping studies are suitable for broad topics and the results gained from systematic review are more reliable as they aim to gather all relevant evidence. Traditional reviews often suffer from researcher bias and typically lack quality assessments or data analysis, Consequently, systematic reviews provide more reliable and unbiased information [52]. That is the reason for choosing the SLR method for the qualitative research method.

3.2.2. Interviews

The interview is selected as a second qualitative method. Typically, these involve direct, remote or telephonic conversations. Interviews can be either structured or semi-structured **[53].** Structured ones are closed-ended with fixed questions, often analyzed statistically. Semi-structured interviews are open-ended, promoting interaction and allowing new questions as information is gathered. This data is analyzed qualitatively. Such interviews encourage interaction and can yield unexpected insights. High-quality responses are more likely if a good rapport exists between interviewer and respondent **[53].** An interview protocol is used for recording, which can include audio-visuals or notes. In our study, semi-structured interviews were conducted, recorded, and transcribed. Convenience sampling and a non-probabilistic method, was used for selecting respondents. I have interviewed 20 individuals from 15 software development companies via Zoom, Teams and google meet.

All interviewee have agreed to audio-visuals recording, respondents were DevOps experienced practitioners, most interviews lasted 30-35 minutes. I have used thematic analysis for examining data and followed steps like: transcribing, organizing, reviewing, coding, identifying themes and even interpreting them. All research questions (RQ1, RQ2, RQ3 and RQ4) were addressed using interview data. I identified benefits of adopting DevOps from literature and confirmed actual benefits through practitioner interviews. The aim was to discover additional challenges and verify if those identified are applicable to different firms in our study. Finding strategies to address these challenges by discussing measures practitioners use is also another aim for interview. Additionally, I examined various DevOps practices in development process and identify difficulties teams face during integration, deployment and testing through interviews.

3.2.3. SLR Process.

I carried out a systematic literature review (SLR) by adhering to the guidelines outlined by Kitchenham et al. [52]. The process began with:

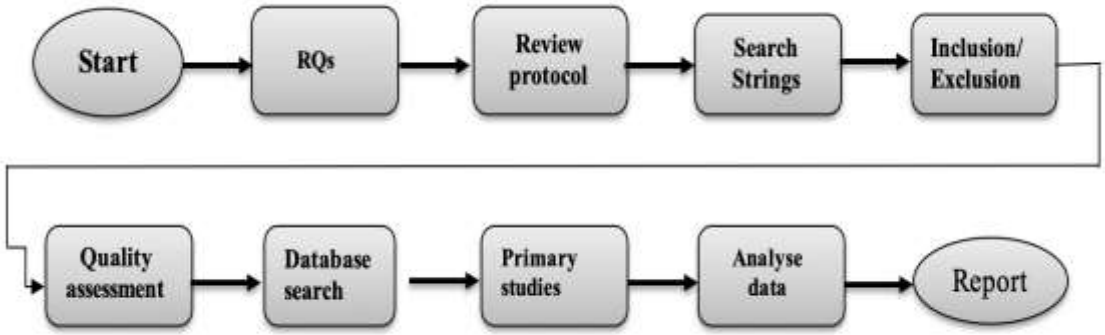


Figure 2: SLR process

3.2.4. Research Strategy

A comprehensive gathering of data from diverse scientific databases was undertaken to procure primary studies and pertinent literature. Keywords were meticulously identified with respect to the research inquiry, and search strings were formulated by employing Boolean operators such as "AND" and "OR" to combine these keywords effectively. Additionally, acronyms corresponding to the keywords were integrated into the search strings to ensure the retrieval of all relevant studies, research papers, scientific articles, and associated documentation. The keywords for this research study are: DevOps, Impact, Development, IT, Startups, Adoption, Challenges, Software Development. The scientific databases and repositories below were chosen for retrieving relevant studies for this research: ScienceDirect, IEE Xplore, Inspec, Scopus, SpringerLink.

Those different databases were selected to ensure a related source of data for covering this study, inspec as a bibliographic database that primarily focuses on scientific and technical literature in the field of physics, electronics, and computer science covered articles extracted from IEE Xplore and ScienceDirect databases. Scopus which is an abstract and citation database that covers a broad range of academic disciplines, is used by researchers, academics, and professionals to access scholarly literature and track citation metrics by indexing a vast collection of journals. Scopus has covered most of the articles from ScienceDirect, IEE Xplore, and SpringerLink. Therefore, there was not enough literature found on DevOps, the reason of all those mentioned databases were used individually by running search strings on each database window for ensuring a satisfactory package of related studies. The search strings used on each database are listed in Table 1

Table 1: Search strings

Scientific Databases	Search strings and metadata
ScienceDirect	("DevOps" OR "Development Operations" OR "Development and Operations") AND ("software engineering" OR "IT operations" OR "continuous integration" OR "continuous deployment" OR "agile methodology" OR "automation" OR "collaborative development" OR "deployment automation" OR "infrastructure as code" OR "containerization" OR "microservices" OR "cloud computing" OR "continuous delivery" OR "release management" OR "configuration management").
IEE Xplore	((DevOps) AND (adoption OR IT OR challenges OR startups OR development OR "software development" OR impact)) AND (pub year:([start year] TO [end year])).
Inspec	(((DevOps) AND (adoption OR startups OR challenges OR impact OR development OR software development OR IT OR impact)) WN KY).
Scopus	(TITLE-ABS-KEY(DevOps) AND TITLE-ABS-KEY (adoption OR impact OR challenges OR startups OR software development OR software OR IT OR development)).
SpringerLink	DevOps AND "DevOps" AND (adoption OR startups OR impact OR software development OR challenges OR software OR IT OR development).

3.2.5. Applying Inclusion and exclusion criteria to the extracted data

I have established specific criteria to narrow down and refine the selection of studies from the database search. This process aimed to eliminate irrelevant and duplicated research papers, articles and journals which contains elements that disqualify the research study target.

Used Criteria:

- These available and written in the English language.
- Studies available preferably in full text for the reason of easy filtering.
- Journal articles and conference papers were taken into consideration.
- Research studies focusing mainly on DevOps organizational aspects.
- Papers that address the benefits and challenges of DevOps adoption in organizations.
- Research studies that concentrate on the principles and practices of DevOps.

Rejected Criteria:

- Research studies focusing on other discipline than DevOps aspects.
- Duplicated research studies.
- Research studies that contain weak information on DevOps.
- Studies not written in English

Quality Assessment of Studies:

The following quality criteria have been established to evaluate the selected studies. Kitchenham et al. [52]. Emphasize that quality assessment is essential to gauge the reliability of the conclusions drawn from primary studies. These criteria are applied to the final set of primary studies identified through database searches.

- Are the research aims and objectives clearly defined?
- Is the research discuss the benefits of implementing DevOps in software development organizations?
- Is the research investigate the obstacles encountered by software organizations in the process of adopting DevOps practices?
- Does the study explain the various principles and practices of DevOps?
- Is the research methodology clearly presented?
- Are the research results clearly presented?
- Are the research conclusions explicitly stated?

3.2.6. Data extraction strategy

The protocol outlines the essential information needed from the primary studies and the methods for obtaining it. It should include an appropriate validation process for any data manipulation or assumptions required. For this thesis, the data needed for subsequent analyses is detailed in Table 2, which shows that 21 primary studies were selected for the systematic literature review. A data extraction form, designed according to guidelines by Kitchenham et al. [52]., was created to record data from these studies. This form was specifically designed to extract data relevant to research questions **(RQs 1,2,3,4).**

Table 2:Required information for data extraction

Paper/ Article Title	Article name specification
Author(s) of studies	Author's name(s)
Type of Article	Article type specification
Year of Publication	Publication year specification
Used research Methodology	Method specification
Research Questions	RQs specification
Is the research specify the benefits of adopting DevOps?	Yes/ No. more details if Yes
Is the research specify the challenges faced while adopting DevOps?	Yes/ No. more details if Yes
Does the study specify the practices of DevOps?	Yes/ No. more details if Yes
Study Conclusions	Specify conclusions

3.2.7. SLR Results

Since DevOps is a relatively a new concept, the publication years for studies should be restricted to between 2008 and 2021.The vast majority of selected studies on DevOps were published after 2015. Figure 6 illustrates the number of selected papers published each year.

Figure 3: Number of selected papers per year

3.2.8. Prisma diagram for Identified studies

The flowcharts in systematic reviews facilitate a quick understanding of review process and the attribution of records. The prisma flow diagram, first published in 2009 and updated in 2020 outlines the sources, numbers and records. These diagrams enhance transparency and communication by detailing the inclusion and exclusion of records at each stage and providing reasons for exclusions [54].

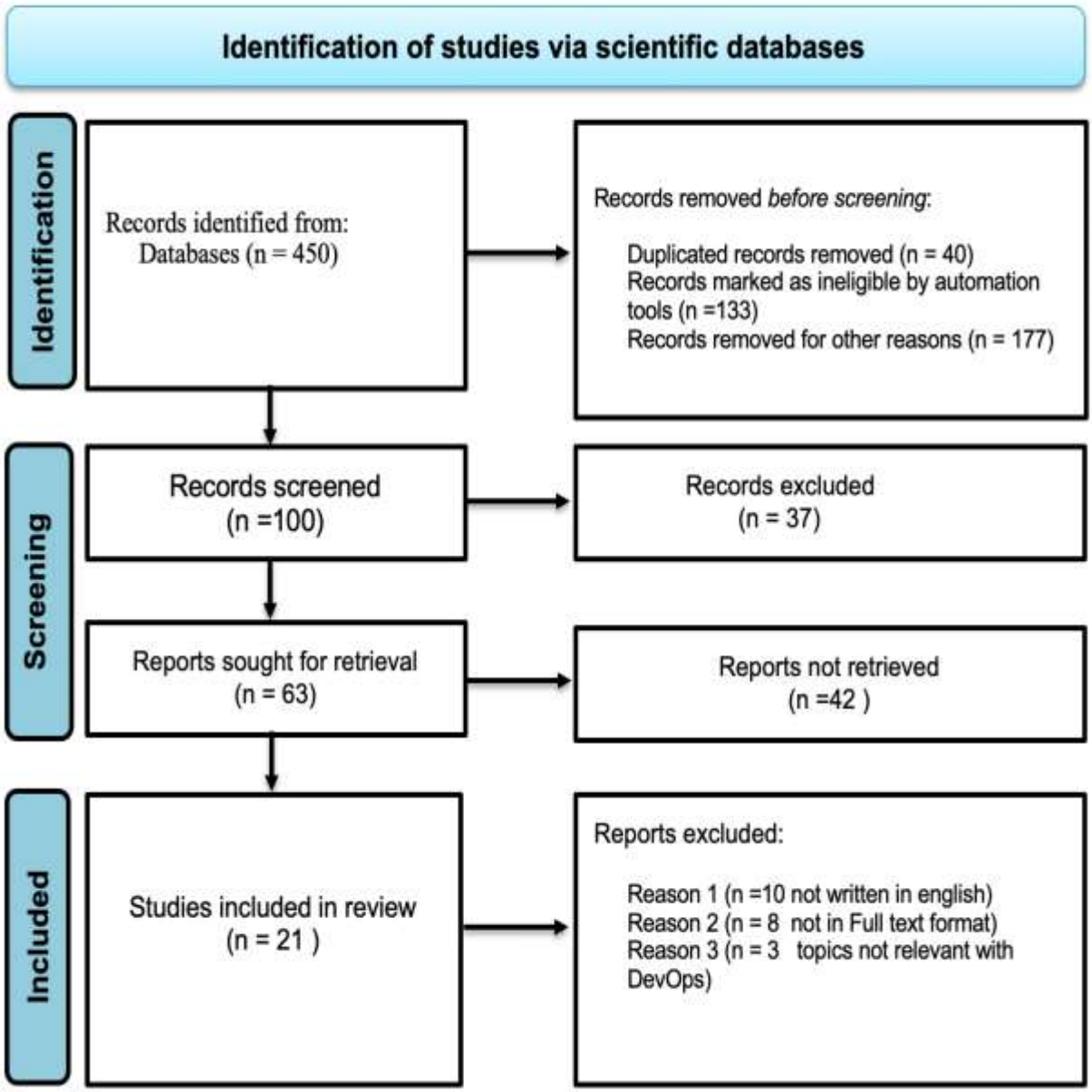


Figure 4: SLR Prisma diagram

3.2.9. Papers analysis on the benefits of DevOps post-adoption in IT companies

This section includes analyzes of primary studies see Appendix_2, on the benefits of adopting DevOps in organizations, various studies ([std2],[std3], [std4], [std5],[std6],[std7], [std8], [std11], [std12], [std13], [std14] [std15] [std16] [std17] [std21]), have identified perceived benefits of DevOps. Mark. S and Jose G **[55]. [**std3**]** observed qualitative improvements in team operations and faster project milestones post-DevOps implementation. Manish et al [56]. [std5] reported similar benefits in a case study at IBM. Matt and Alexandra **[57].** [std10] noted a significant reduction in software release cycle time at Wotif company. Floris [58], et al. [std8**]** found that DevOps positively impacts development operations and quality assurance teams, as well as web service development. Kristian N [59], et al. [std6] identified increased trust, collaboration, and improved workflow through shared responsibilities between development and operations. Bjørnar.T and Jon.I et al **[60] .**[std29] emphasized the importance of cooperation between these teams for successful software deployment and operation, even before the term DevOps was established.

From the extracted primary studies (check Appendix_2)the following benefits have been found:

- Improved software quality and deployment [std2] [std8].
- Efficient team collaboration and easy communication [std2] [std6] [std12] [std14].
- Improved business needs responsiveness[std2].
- Increase software reliability [std2].
- Increased code quality [std2].
- Easy Customer requirements implementation using software development process [std2] [std4].
- Time to time updates between development team and operations are openly provided [std4].
- Structures and standards support Dev Ops and facilitate the development team and operations to accomplish their tasks easily [std4] [std7].
- Software are released frequently [std2] [std4] [std15] [std16] [std21].
- The gap between development and operations teams are overcomes [std11] [std13] [std4].
- Improvement of the overall efficiency of the organization[std8] [std3 [std4] [std12].
- The time required for operational activities such as deployment and change control will be reduced [std5], [std7], [std4], [std8].
- Continuous improvement will be promoted, aiding in the automation of tools and processes and adapting swiftly to changes in the software environment [std13] [std14] [std14].
- It will lead to increased revenue for the organization [std4] [std13] [std5].
- Costs will be lowered by minimizing the need for rework [std11].
- Problems will be identified earlier and resolved more efficiently [std13].
- The time needed to release software will be shortened [std14].
- The average release cycle time will be reduced from weeks to hours [std21].
- Products will be delivered quickly [std16].
- Customer satisfaction will be enhanced [std15].
- DevOps quantifies aspects of the development process, focusing on metrics to improve product development [std14].
- DevOps facilitates the delivery of higher quality services more efficiently[std2].
- Workflows will become smoother and faster [std3] [std7] [std5].

3.2.10. Analysis of studies on the challenges of adopting DevOps in organizations.

This part includes the analysis of primary studies see Appendix_2 , that discuss the challenges organizations face when adopting DevOps. One of the goals of this research is to identify these challenges. The studies referenced as [std1], [std9], [std10], and [std18] highlight various obstacles to DevOps adoption. There are only a few studies that specifically address these challenges in organizations. Kristian N et al. [59], identified several obstacles to adopting DevOps in cloud services through interviews conducted in an IT organization with over 1000 employees. They pinpointed 11 challenges in the early stages of the adoption process. Stephen [61]. [std18] conducted a case study in a small to medium-sized enterprise (SME) with over 200 employees, identifying challenges as the organization transitioned from a legacy system to a new one. Lucy [62]. [std20] explored the challenges of adopting DevOps in the embedded systems domain through a multi-case study involving four companies.

The obstacles encountered while implementing DevOps in organizations include:

- ♣ Insufficient management structure within the organization can impede the adoption process [std1].
- ♣ Many organizations lack adequate training for DevOps, leading to the hiring of DevOps engineers based on their experience as system administrators and software developers [std9].
- ♣ The goals and definitions of DevOps adoption are often unclear [std10].
- ♣ The organizational structure itself can affect the adoption of DevOps [std1].
- ♣ Some customers may prefer different development approaches with strict deployment procedures, which makes DevOps unsuitable for product development in those cases [std18].
- ♣ The regional separation of development and Operations teams can be a barrier to DevOps adoption [std9].
- ♣ DevOps might be perceived as a trendy term, leading to resistance from people who see it negatively [std18].
- ♣ Software developers might fear that DevOps will overload their tasks with operational duties which reduces their work efficiency [std9].
- ♣ DevOps requires all team members to have both development and operations skills, which might be met with resistance due to concerns over lacking deep expertise in both areas[std18].
- ♣ Individuals may focus only on their specialized area and show little interest in other teams' activities [std1].
- ♣ An inflexible system architecture can obstruct DevOps practices such as continuous integration, testing, and deployment[std1].
- ♣ Discrepancies between development, testing, and production environments can complicate collaboration and continuous delivery and deployment [std18].
- ♣ Multiple production environments can introduce complexity, hindering the use of common tools and processes, and affecting continuous delivery [std18].
- ♣ Transitioning to DevOps is more challenging with traditional software [std10] [std18].

- ♣ Defining software architectures that are compatible with DevOps practices has become a challenge to developers [std9].
- ♣ Lack of support from senior management can be a significant challenge [std18].
- ♣ Resistance from employees can also impede the adoption process [std1].
- ♣ External Devices and physical peripherals are incompatible with updated software programs[std10]
- ♣ Complicated visibility of final released software to customers when running tests of requirements [std20].
- ♣ Embedded systems field Lacks automated tools for deploying new features [std18].
- ♣ If the system performance data collected lacks feature usage data, it can hinder DevOps adoption [std1].

Kristian N [59], et al.[std1] recognized numerous obstacles after interviewing software practitioners within a single company. They observed that if teams lacked a clear understanding of DevOps. It could lead to confusion about its objectives and the required actions. Organizational setup was another noted issue. Furthermore, there were mismatches between DevOps methodologies and the client's necessary processes. The geographic dispersion of teams led to communication and procedural problems, especially across different time zones. Some staff viewed DevOps merely as a buzzword which leads to skepticism. Due to the vague definition of DevOps, many personnel did not have confidence in the concepts, feeling that the activities under DevOps were similar to prior practices. This perception could obstruct change adoption. Developers were also concerned about increased workload and new duties affecting their concentration and efficiency. A lack of interest in cross-team functions was noted, with personnel focusing only on their specialization areas. Monolithic architecture posed a significant challenge, affecting creation, testing, and impending continuous builds and deployments.

Discrepancies between development, testing, and production environments were problematic, risking improper software validation before production deployment. This affected both continuous delivery and collaboration since developers were unfamiliar with production environments. Multiple production environments added complexity, hindering automation and the use of common tools and practices. Stephen et al. [61]. [std11], found that an inadequate management framework hindered DevOps adoption. In their case study, adoption was led bottom up by a software development manager who had to persuade senior management of DevOps benefits. Operations teams resisted, believing coding was solely a developer's responsibility. Maintenance of legacy systems was another issue, disrupting the development of new systems and impeding the learning of new technologies. Identified challenges included management framework, senior management buy-in, legacy systems, and employee resistance.

Lucy et al. [62]. [std10] conducted a multi-case study in four companies developing embedded systems and identified hardware dependency and software compatibility issues as primary challenges. Hardware dependencies created development silos, learning to longer development cycles and delayed feature releases. Limited visibility of customer production environments complicated testing environment configurations. The lack of automation tools for continuous deployment in embedded systems was another issue. Companies collected system performance data but lacked feature usage data, which is crucial for continuous product improvement. Thus, the absence of feature usage data is classified as a DevOps adoption challenge.

3.3. Analysis of studies with DevOps practices and principles

One of the research aims is to identify the practices organizations should adopt during the DevOps implementation process. Multiple studies have enumerated different DevOps practices **see Appendix_2** [std18], [std10] Adopting a new methodology requires significant effort from an organization. It involves changing the infrastructure, organizational structure, and development strategy, and ensuring continuous involvement of the operations team. Organizations must incorporate certain practices into their development process as part of the DevOps adoption. Successful DevOps adoption demands considerable internal collaboration, given that it necessitates an overhaul of the organization. The central aspect of DevOps adoption is the continuous integration between software development and operational deployment[63]. To put this change in action, organizations must adhere to specific guidelines and practices.

DevOps aims to dismantle the barriers between development and IT operations, fostering a collaborative environment focused on common goals throughout the process. It incorporates principles and practices that synergize the development and operations departments. Enhancing communication and coordination between these departments leads to better overall performance. Continuous collaboration within DevOps teams necessitates activities like continuous planning, integration, testing, and deployment, which strengthen continuous release of solutions.

3.3.1. DevOps Principles and Practices

The principles and practices of DevOps are attached to aspects like culture, automation, measurement, and sharing[63]**.**

Culture: DevOps merges development and operations, assigning joint responsibility to both departments for delivering high-quality software to end users. Thus, eliminating the “throw over the wall” mentality of codes.

Automation: To ensure quicker delivery, prompt feedback, and short lead times, DevOps relies on full automation of the build, deployment, and testing processes.

Measurement: Understanding current delivery capabilities and setting improvement goals requires comprehensive measurement, from business metrics to test coverage and deployment times.

Sharing: Sharing knowledge, tools culture, infrastructure, and successes strengthens the relationship between development and operations teams.

Kristian N et al. [59],[std19] defines DevOps as “a set of engineering process capabilities supported by certain cultural and technological enablers,” identifying automation, sharing, and communication as key enablers. These principles support technical practices that organizations should adopt.

3.3.2. Identified DevOps Practices by Selected Studies

The literature identifies several DevOps practices [std19], [std20].

Continuous Planning: Agile business plans are maintained through a product backlog and continuous feedback from customers, enabling frequent planning adjustments to cope with rapid business changes.

Continuous Integration: This practice requires early integration, sharing changes promptly, and continuous code validation through automation. Visible artifacts help address integration failures promptly.

Continuous Deployment: Software is automatically deployed to an environment optimizing overall software delivery. Automation and the principle of infrastructure as Code (IaC) are crucial.

Continuous Delivery: Valid software builds are automatically released to customers continuous delivery enables deployment whenever needed.

Continuous Testing: All test cases are automated and executed on every software build aiming to reduce error detection time and root causes.

Continuous Monitoring and Feedback: Infrastructure and user behavior are continuously monitored to provide feedback loops for improving service performance.

3.4. Summary of SLR

With 21 studies selected from different scientific databases see Appendix_2 15 are related to the benefits of DevOps, 4 are discussing the challenges and 2 are addressing DevOps principles and practices. Although DevOps is a relatively new topic with limited high-quality literature, its popularity is growing. Further extensive research is necessary to explore DevOps comprehensively. The study's objectives are to identify the benefits and challenges of DevOps and the practices organizations should include. Applying a systematic literature review (SLR) helped answer research questions RQs 1,2,3 and 4, the perceived benefits of DevOps adoption include increased software release frequency, improved quality, reduced time to market, and increased customer satisfaction. However, organizations face challenges like a lack of proper management structure, employee resistance, fear of change, and legacy software. Adopting DevOps requires adhering to principles of culture, automation, sharing, and communication. The identified technical practices, continuous planning, integration, deployment, delivery, testing, and monitoring support, these principles enable successful DevOps adoption. Further validation and detailed explanations of these practices will be clearly explained and will follow interviews with DevOps practitioners.

3.5. Reliability

During the process of conducting a systematic literature review (SLR), certain threats to validity may arise. It's crucial to address these threats to enhance the reliability of the data. Ignoring these threats could lead to incorrect interpretations. Researchers identify and narrow down these threats using validity categories such as:

3.6. Internal Validity

In this research, I have conducted a database search to gather literature in the field of DevOps, during this process, internal validity emerged as a significant concern. To address this threat, I followed specific procedures, including formulating search strings. The search string has been applied to different scientific database windows and after inclusion and exclusion criteria on obtained literature was taken into action, a typical example of exclusion is studies unrelated to the search area topics. After several iterations quality assessment criteria were performed to assess the retrieved studies from both database searches. This approach helped to mitigate the risk of publication bias. Finally, a finalized list of primary studies based on the quality assessment was taken into consideration.

3.7. Constructive Validity

In the context of constructive validity, I have addressed potential confounding factors and ensured that the study aligns with the intended aims and objectives. To mitigate threats, I carefully structured sections based on research questions. One significant concern was the search string used to retrieve relevant studies. Another challenge was the risk of overlooking critical information while documenting findings. To counter this, I have meticulously maintained an Excel sheet with essential data from primary studies. Data extraction forms facilitated an organized listing of information. Lastly, domain applicability posed a threat due to limited literature availability. To address this, the search string was tailored using domain-specific keywords, and quality assessment ensured the exclusion of irrelevant or flawed studies.

3.8. Interviews

3.8.1. Analysis of Interview Responses

This part of the thesis contains a detailed analysis and results derived from the conducted interviews.

3.8.2.Target Population and Sampling Method

The interviews targeted software engineering professionals who have experience with DevOps. A sampling method was utilized to select respondents, drawing from personal contacts, e-mails, and LinkedIn connections. The sampling involves collecting responses from readily available and willing participants, the choice of sampling method was driven by the specific and limited nature of the target population. Since DevOps is a relatively recent concept and not yet widely adopted by organizations, a random sample was deemed impractical.

3.8.3. Interview Process and Questionnaire Design

The interview format see **Appendix 3**, was open-ended questions Initially based on findings from a Systematic Literature Review (SLR), the questionnaire was refined, revised, and well-prepared. The interview includes questions about the benefits and challenges of adopting DevOps, strategies to overcome these challenges, various DevOps practices, and issues faced during solutions release like continuous integration, deployment and testing mostly done by operations teams.

3.8.4. Interview Execution

Ten interviews were conducted with software developers, data analysts, and designers from different 10 IT organizations and startups. Each participant was asked the prepared same questions on DevOps and some other additional spontaneous questions as needed. The interviews were conducted by using different audio-visuals conferences platforms such as Zoom, google meets and MS teams depending on the choice and familiar platforms currently used by interviewees, with confidentiality assured to encourage honest and detailed responses. All interviews were audio recorded with the participants' consents.

3.8.5.Interview data analysis methodology

The collected data was analyzed using the thematic analysis of interviews conducted with software engineering practitioners expected in DevOps sector:

The following steps was followed for interview data analysis:

- Transcription

All interview recordings were transcribed verbatim into separate word documents. This meticulous transcription ensured that no details were omitted. Reducing potential bias.

- Data organization

Each transcript was assigned a unique identifier to maintain organization and facilitate efficient data analysis.

- Familiarization

The transcripts were thoroughly read to gain an in-depth understanding of the data, allowing the researchers to immerse themselves in the content.

- Coding

Manual examination of the transcripts led to the identification of various codes. Each code was highlighted with a distinct color corresponding to the specific question asked.

- Themes Identification

The identified codes were grouped into themes based on their similarities and relevance. The inclusion criteria for these themes were defined by the frequency and relevance of the responses. Frequency refers to topics mentioned by multiple respondents, while relevance pertains to both the literature and the specific focus of this research.

- Interpretation

The themes were then interpreted to uncover the underlying meanings and insights related to the adoption and implementation of DevOps practices.

3.9. Thematic Analysis and Results for RQs.

3.9.1. Thematic analysis for RQ1

Table 3 provides a detailed overview of the codes and themes identified in relation to research question 1 (RQ1). From the analysis of 10 interview transcripts see **Appendix 3**, I have identified a total of 22 distinct benefits associated with DevOps post adoption in organizations. These benefits were selected based on their pertinence to the study and how regularly they were mentioned. The identified benefits were grouped under the overarching themes of "Benefits."

Table 3: Thematic analysis for RQ1

Codes/Titles	Included Criteria	Interviewees No	Identified Themes
Improved collaboration between Dev and Ops	Pertinent, Regular	2,3,5,6	Dev Ops Benefits
Increased deployment frequency	Pertinent	1,7	Dev Ops Benefits
Enhanced software quality and reliability	Pertinent, Regular	4	Dev Ops Benefits
Faster issue resolution and reduced complexity	Pertinent, Regular	9	Dev Ops Benefits
Greater automation of repetitive tasks	Pertinent, Regular	10	Dev Ops Benefits
Improved transparency and accountability	Pertinent, Regular	2,5	Dev Ops Benefits
Better resource utilization and efficiency	Pertinent, Regular	7,4,9	Dev Ops Benefits
Enhanced customer satisfaction	Pertinent, Regular	1,9,3	Dev Ops Benefits
Reduced time spent on unplanned work	Pertinent, Regular	4,8	Dev Ops Benefits
Increased innovation and experimentation	Pertinent, Regular	10,5,2,8	Dev Ops Benefits
Better scalability and availability	Pertinent, Regular	6,1	Dev Ops Benefits
Higher flexibility in response	Pertinent, Regular	6	Dev Ops Benefits
Improved quality of codes	Pertinent, Regular	1,3,6,7,9,10	Dev Ops Benefits
Increased confidence in deployment process	Pertinent, Regular	4	Dev Ops Benefits
Reduction in manual errors and risks	Pertinent, Regular	7	Dev Ops Benefits
More effective feedback loops between teams	Pertinent, Regular	2	Dev Ops Benefits
Greater job satisfaction and team morale	Pertinent, Regular	8	Dev Ops Benefits
Better management of technical debt	Pertinent, Regular	6,9	Dev Ops Benefits
Enhanced performance and stability	Pertinent, Regular	5	Dev Ops Benefits
Improved alignment with business goals	Pertinent, Regular	8	Dev Ops Benefits
Reduced separated silos	Pertinent, Regular	1,6,10	Dev Ops Benefits
Evolution of agile methodology	Pertinent, Regular	9	Dev Ops Benefits

3.9.2. Interview responses related to RQ1

This part focuses on the assessment of DevOps post-adoption benefits in organizations, from 10 interviews conducted using an interview questionnaire see **Appendix 3**, I have identified almost 22 reliable benefits categorized under the" Benefits" theme. A complete number of interview participants highlighted the advantages of integrating DevOps in their organizations, regardless of traditional methodology already used by companies like the waterfall model, and agile as an advanced model, DevOps become a new evolution as is currently solving different difficulties that software developers were facing before it's adoption.

Completed 4 interviewees 2,3,5,6, emphasized that DevOps improved collaboration between the development team and the operations team, interview 2 said that when it was time to communicate and exchange ideas on the ongoing software projects the development team struggled by explaining how codes work to the operations teams if they are in need, but with the help of different version controls platforms classified as DevOps practices it's easier to mention and share every single part(codes) of the software with the rest of the team include operation teams by using code documentation, updating changes(Push and Pull), comments and highlighting crucial areas of codes blocks. Interviewee 3 mentioned that collaboration is the backbone of better success and progress, and interviewees 5 and 6 have almost approached the ideas delivered by 2 and 3.

Interviewees 1 and 7 stated that DevOps increased deployment frequency as another key benefit of its adoption, interviewee 1 said again that deployment and release time was incredibly slow using the waterfall cascading stages gates, where it was difficult when a client needed to add some new requirements in the project, at that times the whole adding new requirements caused a loss of time and cost by restarting the model from the beginning, but with the rise of agile model and DevOps helped developers and clients who interact with clients directly on each sprint with the exchange of feedback during the scrum meeting and that increased a quick software deployment faster than before. Additionally, interviewee 7 also answered that containerization software like docker and others becomes best nowadays when it's time to deploy the project setup package which is easier to access remotely differently to the way developers used to burn setups on CDs, F-disks and deliver it manually, which sometimes was exposed to different damages.

Enhancing software quality and reliability as another benefit of adopting DevOps, interviewee 4 assumed that a strong collaboration throughout DevOps practices contributed to a strong review of code scripts, by using a determined time setting for reaching the sprint goals of the agile model, which leads to reliable solutions and quality software, additionally, interviewee 8 said the ability of continuous integration and testing ensures that software is consistently high-quality and dependable, interviewee 9 said that faster issue resolution and reduced complexity as another benefit and explained that DevOps streamlined processes and automation helped to quickly identify and fix issues by reducing overall system complexity.

Interviewee 10 mentioned how automation of repetitive tasks has become one of the major key factors of quick software delivery and remarkable time keeping, interviewees 2 and 5 revealed the same ideas for how DevOps improved transparency and accountability by providing clear visibility into the entire development and operation process throughout real-time monitoring and comprehensive documentation. Better resource utilization and efficiency is another benefit mentioned by interviewees 7, 4, and 9 in their responses they explained how automating repetitive tasks and streamlining workflows reduces the need for manual intervention and allows teams to focus on more strategic activities.

Enhancing customer satisfaction is a benefit that interviewees 1, 9, and 3 mentioned in their responses by enabling more frequent and reliable software releases, and ensuring that new features, updates, and bug fixes reach customers faster, this continuous delivery of improvements keeps the product aligned with customer needs and expectations. Interviewees 4,8 talked about reducing the time spent on unplanned work as a DevOps benefit, by automating routine tasks and implementing proactive monitoring and alerting systems helped teams to identify and address potential issues before they become major problems. Interviewees 10,5,2 and 8 stated how the increase in innovation and experimentation caused by DevOps is a typical benefit of creating an environment where continuous integration and continuous delivery (CI/CD) streamline the development and deployment process.

A benefit mentioned by interviewees 6 and 1 is that DevOps enhances scalability and availability by leveraging automation and infrastructure as code (IaC) to manage and deploy resources efficiently. This approach allows systems to dynamically adjust to varying workloads, ensuring optimal performance during usage times. Higher flexibility in response is another mentioned DevOps benefit which allows the changes of requirements and market conditions through its practices of continuous integration and continuous delivery (CI/CD) interviewee 6 said again that, this allows teams to quickly implement and deploy changes without significant downtime. Another benefit that has been pointed out by the majority of interviewees 1,3,6,7,9 and 10 is that DevOps enhances code quality by incorporating practices such as continuous integration and continuous testing, which ensure that code changes are regularly integrated and rigorously tested.

Interviewee 4 also stated that DevOps increased confidence in the deployment process through automation and standardized workflows. Continuous integration and continuous delivery (CI/CD) pipelines ensure that code changes are consistently tested and validated before deployment. Interviewee 7 in his responses mentioned that DevOps minimizes manual errors and risks by leveraging automation for repetitive and complex tasks. Automated testing, integration, and deployment processes ensure that tasks are performed consistently and accurately eliminating the variability and mistakes often associated with manual interventions. More effective feedback loops between teams a benefit mentioned by interviewee 2 it facilitates more feedback loops through practices such as continuous integration (CI) and continuous delivery (CD), which ensure that feedback is received and addressed promptly.

Greater job satisfaction and team morale it’s a benefits promoted by DevOps by fostering a collaborative and supportive work environment These benefits have been discussed by Interviewee 8, and interviewees 6,9 in their responses to the interview described how DevOps helped in better management of technical debts, DevOps practices offer effective solutions By implementing a shift-left testing, prioritizing major issues, fostering collaboration, emphasizing automation and promptly addressing problems by minimizing debt accumulation. Proactive management of technical debt leads to faster delivery times and improved product quality. Interviewee 5 in his responses said that DevOps enhanced performance and stability for the reason that automated monitoring and alerting systems provide real-time insights into system performance, allowing teams to quickly identify and resolve issues before they impact users. DevOps improved alignment with business goals is a benefit that has been discussed by interviewee 8 he explained how it enhances alignment with business goals by fostering close collaboration between development, operations, and business teams. One of the major benefits of DevOps is to reduce separated silos, interviewees 1,6, and 10 separately mentioned how integrated tools and processes facilitate seamless communication and information and information sharing, breaking down barriers that traditionally isolated these teams. Finally, interviewee 9 said how DevOps becomes an evolution of agile methodology by extending its principles of collaboration, iterative development, and responsiveness to include operations.

3.9.3. Thematic analysis for RQ2

Table 4 contains code titles and theme identifications related to research question 2 RQ2. From the analysis responses of 10 interview transcripts, several 20 different obstacles linked to the adoption of DevOps in IT startups and organizations. The selection of those obstacles or challenges on their frequency or regularity and their relevance or pertinence, and identified challenges were classified as DevOps obstacle’s themes.

Table 4: Thematic analysis for RQ2

Codes /Titles	Included Criteria	Interviewees No	Identified Themes
Cultural resistance to change	Pertinent, Regular	1,6,7	Dev Ops Obstacles
Lack of necessary skills and training	Pertinent	3,5	Dev Ops Obstacles
Integration of new tools and process with existing systems	Pertinent, Regular	6,3,7	Dev Ops Obstacles
Siloed teams and lack of communication	Pertinent, Regular	2,5	Dev Ops Obstacles
Resistance from management or stakeholders	Pertinent	6,8	Dev Ops Obstacles
Inconsistent practices across different teams	Pertinent, Regular	1,2	Dev Ops Obstacles
Insufficient budget for tools and training	Pertinent, Regular	1,2,4	Dev Ops Obstacles
Legacy systems and technical debt	Pertinent	3,5,6	Dev Ops Obstacles
Security and compliance concerns	Pertinent, Regular	7,5,9	Dev Ops Obstacles
Difficulty in measuring success and ROI	Pertinent	3,7,8	Dev Ops Obstacles
Fear of job loss or changes in job roles	Pertinent, Regular	9,5,7	Dev Ops Obstacles
Inadequate infrastructure and resources	Pertinent, Regular	7,8	Dev Ops Obstacles
Complexity in managing hybrid environments	Pertinent, Regular	4,6	Dev Ops Obstacles
Misalignment between IT and business objectives	Pertinent, Regular	5,6	Dev Ops Obstacles
Lack of a clear DevOps strategy or roadmap	Pertinent, Regular	7,4	Dev Ops Obstacles
Fragmented toolchain and processes	Pertinent	8,3	Dev Ops Obstacles
Difficulty in maintaining continuous improvement	Pertinent, Regular	4,8	Dev Ops Obstacles
Challenges in scaling DevOps practices	Pertinent	6,7	Dev Ops Obstacles
Conflicting priorities and deadlines	Pertinent, Regular	9,10	Dev Ops Obstacles
Insufficient support from leadership	Pertinent	1,9,5	Dev Ops Obstacles

3.9.4. Interview Responses for RQ2

With the provided responses from 10 interviewees about the challenges or obstacles faced by DevOps practitioners after its adoption (see the questionnaire **Appendix 3**). Interviewees 1,6,7 in their respective responses which targeted on culture resistance to change they provided different reasons like mindset and attitude shifts, fear of the unknown, Resistance to process changes, loss of control, communication and collaboration barriers and many more. Interviewees 3,5 said that the lack of necessary skills and training is one of DevOps challenges, the skill gaps of employees in automation of CI/CD and the lack of proper training requirements have been included in their responses. Interviewees 6,3,7 declared that integration of new tools and process with existing systems becomes a challenge as many organizations rely on legacy systems that may not be easily with new DevOps tools and process, integrating modern automation tools with these older systems can be complex and time consuming, and even migrating data between old and new systems can also present significant risks and challenges such as data loss, corruption, and breaches. Interviewees 2 and 5 said that siloed teams and lack of communication are major barriers to successfully implementing DevOps in an organization. The development, operations, QA and other teams often work in isolation with minimal interaction. Each team focuses on its specific goals and tasks, leading to misalignment with overall business objectives. From their responses interviewees 6 and 8 mentioned the resistance from management or stakeholders as another obstacles of DevOps adoption, the management and stakeholders often have a fear of change, especially when current process appear to be working well, they may be reluctant to invest in new practices that seem risky or disruptive. Interviewees 1 and 2 from their approximate responses discussed on inconsistent practices across different teams as a DevOps challenge, by showing how teams might use different tools for version control, CI/CD pipelines, testing and monitoring and this fragmentation can cause

integration issues and complicate collaboration. Within responses from interviewees 1,2 and 4 they have approximately mentioned how insufficient budget for tools and training is a common challenge in adopting DevOps practices, implementing DevOps often requires significant initial investments in tools, infrastructure, and training programs, limited budgets can make it difficult to justify these expenditures.

Interviewees 7,5 and 9 in a combination of their answers to the interview questions they said that security and compliance concerns is a significant challenge of DevOps adoption and ensuring that DevOps process are secure and compliant with regulatory requirements is critical to protect sensitive data and maintain trust, DevOps emphasizes rapid development and deployment cycles, which can sometimes lead to security being overlooked or inadequately addressed. Different industries are subject to various regulatory requirements (e.g. GDPR, HIPAA, PCI-DSS). Ensuring that all aspects of the DevOps pipeline comply with these regulations can be challenging. Interviewees 3,7 and 8 said that measuring success and ROI (Return on Investment) in DevOps can be challenging due the complexity and multi-faced nature of DevOps transformations which impact various aspects of the software development lifecycle, including development speed, quality, collaboration and customer satisfaction. Measuring all these dimensions comprehensively can be complex, many benefits of DevOps, such as improved team moral, better collaboration and enhanced customer satisfaction are intangible and difficult to quantify directly.

Interviewees 9,5 and 7 have pointed the fear of job loss or changes in job roles as another challenge in the adoption of DevOps, this fear can lead to resistance from employees, impacting the overall success of DevOps transformation and even employees may fear that automation and new tools will make their roles redundant, and employees may worry about their ability to adapt and their future relevance within the organization. Interviewees 7 and 8 in their respective answers to the interview questions they mentioned inadequate infrastructure and resources as a significant challenge to the successful implementation of DevOps practices, without the necessary infrastructure, organization struggle to achieve the automation, integration, and scalability that DevOps requires.

Respectively interviewees 4 and 6 said that managing hybrid environments, which include both on premises and Claud-based infrastructure, introduces significant complexity in DevOps practices. This complexity arises from the need to integrate, secure and maintain consistent performance across diverse platforms. Misalignment between IT and business objectives is one of other challenges which interviewees 5 and 6 have mentioned in their responses, they said that it can significantly hinder the success of DevOps initiatives, when IT and business goals are not aligned, it can lead to misunderstandings and lack of support for crucial projects, misalignment can lead to conflicts over resource allocation, with business units and IT competing for budget and manpower.

The lack of clear DevOps strategy or roadmap is essential for the successful implementation and sustainability of DevOps practices within organization. Without it, teams may lack direction, face inconsistent processes and struggle with achieving meaningful outcomes, this is taken as interview answer for interviewees 7 and 4. Without a clear strategy, the vision for DevOps implementation maybe vague, leading to misalignment and confusion among teams. Teams may struggle to prioritize tasks and initiatives, resulting in fragmented efforts and suboptimal outcomes.

Interviewees 8 and 3 in a summary of their responses said a fragmented toolchain and processes are not well integrated, which can lead to inconsistencies, inefficiencies, and communication barriers between teams, and manual handoffs between tools can lead to errors and slow down processes. Interviewees 4 and 8, revealed that difficulty in maintaining continuous improvement becomes a challenge, contrary maintaining continuous improvement is essential for the success of DevOps, as it enables organizations to adapt, innovate, and optimize their processes over time, however achieving and sustaining this requires overcoming various challenges. Interviewees 6 and 7, showed how scaling DevOps practices across an organization can be complex and challenging due to a variety of factors. Successfully scaling DevOps requires addressing these challenges strategically to ensure consistency, efficiency, and effectiveness across diverse teams and projects.

Interviewees 9 and 10, said that conflicting priorities and deadlines pose significant challenges in DevOps environments, where the need for rapid development and deployment often clashes with other organizational demands, balancing these competing priorities requires strategic planning, effective communication, and robust processes. Business units may prioritize market delivery dates and features, while IT teams focus on technical debt reduction and system stability. Finally, interviewees 1,9 and 5, said that the insufficient budget for tools and training is a common challenge in adopting DevOps practices. Implementing DevOps often requires significant initial investments in tools, infrastructure, and training programs. Limited budgets can make it difficult to justify these expenditures.

3.9.5. Thematic analysis for RQ3

From the 10 interview transcripts asked to DevOps practitioners using the interview questionnaire see **Appendix 3** , 20 distinct codes addressing how organizations can overcome challenges after adopting DevOps were identified. These codes were chosen based on their pertinence to the study and how regular they appeared. The identified codes were grouped under the themes “DevOps solutions”. Table 5 provides a detailed list of these codes and themes related to the Research question 3(RQ3). The process of identifying these codes involved an analysis of the interview transcripts, ensuring that each code was directly linked to overcoming specific challenges in DevOps adoption. The inclusion criteria focused on both the relevance to the research objectives and the recurrence of the themes during the interviews.

This rigorous approach ensured that the solutions identified were both practical widely applicable. To further validate the findings, the codes were reviewed and cross-checked, enhancing the reliability of the results. The categorization into the “DevOps solution” theme helps in clearly delineating the practical steps organizations can take to address DevOps challenges. Table 4 is structured to not only present the codes but also to offer a brief explanation of each, providing context and deeper insight into how each solution can be implemented. This table serves as a valuable resource for practitioners and researchers alike, offering concrete guidance derived from real world experiences. By systematically categorizing the codes, the study provides a clear framework that organizations can follow to improve their DevOps practices.

Table 5:Thematic analysis for RQ3

Codes /Titles	Included Criteria	Interviewees No	Identified Themes
Providing comprehensive training and education	Regular	3,7,8	Dev Ops solutions
Promoting a culture of collaboration	Regular	9,5	Dev Ops solutions
Implementing DevOps practices to ease transition	Regular	1,3,7	Dev Ops solutions
Encouraging leadership buy-in and support	Regular	2,5	Dev Ops solutions
Establish clear goals and metrics for DevOps success	Regular	6,8	Dev Ops solutions
Creating cross-functional teams with shared responsibility	Regular	6,2	Dev Ops solutions
Leveraging automation to reduce manual tasks	Regular	1,2,4	Dev Ops solutions
Incrementally integrating new tools and process	Regular	3,5,6	Dev Ops solutions
Conducting regular assessments and feedback loops	Regular	9,10	Dev Ops solutions
Aligning DevOps initiatives with business objectives	Regular	1,6,7	Dev Ops solutions
Fostering an environment of experimentation	Pertinent, Regular	9,5,7	Dev Ops solutions
Developing a strong change management plan	Pertinent, Regular	7,8	Dev Ops solutions
Encouraging open communication and knowledge sharing	Pertinent, Regular	2,6	Dev Ops solutions
Building a community of practice for DevOps	Regular	4,6	Dev Ops solutions
Recognizing and rewarding DevOps achievements	Regular	1,4	Dev Ops solutions
Providing resources and support for continuous learning	Pertinent	8,3	Dev Ops solutions
Emphasizing the importance of security and compliance	Regular	7,8	Dev Ops solutions
Collaborating with external experts and consultants	Regular	6,7	Dev Ops solutions
Creating a flexible and adaptive DevOps framework	Regular	7,5,9	Dev Ops solutions
Prioritizing the modernization of legacy systems	Regular	1,9,5	Dev Ops solutions

3.9.6. Interview responses for RQ3

The study identified several key solutions to address the challenges faced by organizations during DevOps adoption, derived from the insights of various interviewees, each contributing unique perspective." Providing comprehensive training and education" was highlighted by interviewees 3,7 and 8 emphasizing the importance of equipping teams with the necessary skills and knowledge. "Promoting a culture of collaboration" was noted by interviewees 9 and 5, underscoring the need for teamwork and shared goals. "Implementing DevOps practices to ease transition" was recommended by interviewees 1,3 and 7, focusing on the adoption of best practices to facilitate smoother transitions. "Encouraging leadership buy-in and support" was suggested by interviewees 2 and 5, stressing the role of leadership in driving DevOps initiatives.

"Establishing clear goals and metrics for DevOps success" was pointed out by interviewees 6 and 8, highlighting the need for measurable objectives to track progress." Creating cross functional teams with shared responsibility" was advocated by interviewees 6 and 2, promoting collaboration across different functions within the organization. "Leveraging automation to reduce manual tasks" was emphasized by interviewees 1,2 and 4, focusing on the efficiency gains from automating repetitive tasks. "Incrementally integrating new tools and processes" was suggested by interviewees 3,5 and 6, recommending a gradual approach to adopting new technologies.

"Conducting regular assessments and feedback loops" was recommended by interviewees 9 and 10, stressing the importance of continuous improvement through regular evaluations." Aligning DevOps initiatives with business objectives was highlighted by interviewees 1,6 and 7, ensuring that DevOps efforts support the broader goals of the organization." Fostering an environment of experimentation" was suggested by interviewees 9,5 and 7, encouraging a culture of innovation and trial. "Developing a strong change management plan" was recommended by interviewees 7 and 8, focusing on structured approaches to managing change." Encouraging open communication and knowledge sharing" was emphasized by interviewees 2 and 6, highlighting the importance of transparent communication channels. "Building a community of practice for DevOps" was pointed out by interviewees 4 and 6, fostering a support network for sharing best practices. "Recognizing and rewarding DevOps achievements" was noted by interviewees 1 and 4, motivating teams through acknowledgment of their successes." Providing resources and support for continuous learning" was suggested by interviewees 8 and 3, promoting ongoing professional development.

"Emphasizing the importance of security and compliance" was highlighted by interviewees 7 and 8, ensuring that DevOps practices meet regulatory requirements. "Collaborating with external experts and consultants" was recommended by interviewees 6 and 7, leveraging external expertise to enhance internal capabilities." Creating a flexible and adaptive DevOps framework" was suggested by interviewees 7,5 and 9, promoting a framework that can evolve with changing needs. "Prioritizing the modernization of legacy systems" was emphasized by interviewees 1,9 and 5, addressing the challenges of outdated technologies. These insights, drawn from diverse interviewees, provide a comprehensive roadmap for organizations seeking to overcome challenges in their DevOps journey.

3.9.7. Thematic Analysis for RQ4

A total of 20 codes representing the practices an organization should adopt for successful DevOps implementation were identified from 10 interview transcripts (see the questionnaire **Appendix 3**). These codes were chosen based on their pertinence to the study and their appearance regularity. The identified codes were categorized under the theme "Practices". Table 6 presents the codes and themes identified for Research question 4(RQ4). The process of identifying these codes involved a meticulous analysis of the interview transcripts, ensuring that each code was directly linked to essential DevOps practices. Inclusion criteria focused on both the relevance to the research objectives and the recurrence of the themes during the interviews. This through approach ensured that the identified practices were practical and widely applicable across various organizational contexts.

Table 6 is designed not only to list the codes but also to offer concise explanations for each, providing context and insight into how these practices can be implemented. The structured presentation in Table 6 facilitates easier reference and application of these practices, helping organizations to prioritize and integrate them into their existing workflows. This comprehensive approach aims to support a smoother transition to DevOps, fostering a culture of continuous improvement and innovation. Through this detailed analysis, the study contributes valuable insights into the practical steps necessary for successful DevOps adoption, ultimately aiding organizations in achieving their DevOps goals.

Table 6: Thematic analysis for RQ4

Codes /Titles	Included Criteria	Interviewees No	Identified Themes
Continuous integration and Continuous deployment (CI/CD)	Pertinent, Regular	4,6,8	DevOps practices
Infrastructure as Code (Iac)	Pertinent, Regular	10,5	DevOps practices
Automated testing and monitoring	Pertinent, Regular	1,5,7	DevOps practices
Configuration management tools	Pertinent, Regular	7,5	DevOps practices
Version control systems	Pertinent, Regular	1,4	DevOps practices
Containerization and orchestration (Docker, Kubernetes)	Pertinent, Regular	8,3	DevOps practices
Microservices architecture	Pertinent, Regular	7,2,4	DevOps practices
Logging and centralized monitoring solutions	Pertinent, Regular	1,5,6	DevOps practices
Incident management and response tools	Pertinent, Regular	2,4	DevOps practices
Collaboration and communication platforms	Pertinent, Regular	8,6,9	DevOps practices
Security as code and automated security testing	Pertinent, Regular	2,5,4	DevOps practices
Performance monitoring and optimization tools	Pertinent, Regular	9,8	DevOps practices
Dependency management and build automation tools	Pertinent, Regular	3,8	DevOps practices
Service mesh for managing microservices	Pertinent, Regular	1,6	DevOps practices
Artifact repositories and package management	Pertinent, Regular	6,4	DevOps practices
Release management and deployment automation	Pertinent, Regular	9,3	DevOps practices
Feature flagging and canary deployments	Pertinent, Regular	8,4	DevOps practices
Blue-green and rolling deployments	Pertinent, Regular	5,2	DevOps practices
Chaos engineering for resilience testing	Pertinent, Regular	8,5,	DevOps practices
Continuous feedback and improvement mechanisms	Pertinent, Regular	9,5,1	DevOps practices

3.9.8 Interview Responses for RQ4

Based on the analysis of 10 interview transcripts, 20 distinct codes representing essential practices for successful DevOps adoption were identified. These codes were selected based on their pertinence to the study and frequency of regularity. They were categorized under the theme" DevOps practices". Table 6 presents these codes and themes related to research question 4(RQ4)." Continuous Integration and Continuous Deployment (CI/CD)" was emphasized by interviewees 4,6 and 8, highlighting its significance in automating and streamlining the development process." Infrastructure as Code (IaC)" noted by interviewees 10 and 5, underscores the importance of managing and provisioning computing infrastructure through machine-readable definition files. "Automated Testing and Monitoring" highlighted by interviewees 1,5 and 7, focuses on ensuring code quality and system performance through continuous testing and real-time monitoring.

"Configuration Management Tools" were pointed out by interviewees 7 and 5, stressing the need for consistent system configurations across different environments. "Version Control Systems" mentioned by interviewees 1 and 4, are crucial for tracking changes in code and facilitating collaborative development. "Containerization and Orchestration like Docker, Kubernetes" was emphasized by interviewees 8 and 3 showcasing the need for efficient management of containers.

"Microservices Architecture" highlighted by interviewees 7,2 and 4, promotes building applications as a suite of small, independently deployable services. "Logging and Centralized Monitoring Solutions" noted by interviewees 1,5 and 6 are essential for maintaining visibility and diagnosing issues in a distributed system. "Incident Management and Response Tools" mentioned by interviewees 2 and 4, focus on handling and resolving incidents efficiently to minimize downtime. "Collaboration and Communication Platforms", emphasized by interviewees 8,6, and 9 facilitate better teamwork and information sharing. "Security as Code and Automated Security Testing ", highlighted by interviewees 2,5 and 4 integrate security practices into the DevOps workflow. "Performance Monitoring and

Optimization Tools" noted by interviewees 9 and 8, are crucial for ensuring optimal system performance and user experience.

"Dependency Management and Build Automation Tools", mentioned by interviewees 3 and 8, help manage project dependencies and automate the build process. "Service Mesh for Managing Microservices", highlighted by interviewees 1 and 6, focuses on managing service-to-service communication within microservices architectures. "Artifact Repositories and Package Management" noted by interviewees 6 and 4, are essential for storing and managing binary and dependencies. "Release Management and Deployment Automation" mentioned by interviewees 9 and 3, streamline the process of deploying new software releases. "Feature Flagging Canary Deployments", emphasized by interviewees 8 and 4, enable testing new features in production with minimal risk. "Blue Green and Rolling Deployments", noted by gradually shifting traffic between different environments. "Chaos Engineering for Resilience Testing", highlighted by interviewees 8 and 5, focuses on testing the system's ability to withstand unexpected disruptions. "Continuous Feedback and Improvement Mechanisms", mentioned by interviewees 9,5 and 1, emphasize the importance of iterative improvements based on regular feedback. These insights, categorized under "DevOps practices," provide a comprehensive guide for organizations aiming to adopt DevOps practices effectively.

3.10. Identifying SLR and Interviews findings.

The benefits of implementing DevOps practices in software development and operations found in Table 7 are numerous, as evidenced by the synthesis of Systematic Literature Reviews (SLR) and interviews. Improved software quality and deployment (BFI) is a significant benefit, supported by SLR findings. Additionally, efficient team collaboration and easy communication (BF2) are facilitated through DevOps, further enhancing responsiveness to business needs (BF3). The increased reliability of software (BF4) is another key benefit highlighted by SLR. Increased code quality (BF5) is noted, with evidence from both SLR and Interviews. The implementation of customer requirements is streamlined through the software development process (BF6), according to SLR findings. Regular updates between development teams and Operations (BF7) are openly provided, ensuring transparency and efficiency. Structures and standards supporting DevOps facilitate seamless task accomplishment for development teams and operations (BF8). Frequent software release (BF9) are a direct result of adopting DevOps practices, bridging the gap between development and operations teams (BF10), as supported by both SLR and interviews.

The overall efficiency of organizations improves (BF11), with SLR indicating reduced time for operational activities such as deployment and change control (BF12). Continuous improvement is promoted, aiding the automation of tools and process and allowing swift adaptation to changes in the software environment (BF13), supported by SLR and Interviews.

DevOps implementation leads to increased revenue for organizations (BF14), as interviews suggest. Additionally, costs are lowered by minimizing the need for rework (BF15), and problems are identified and resolved more efficiently (BF16), both confirmed by SLR and interviews. The time needed to release software is shortened (BF17), and the average release cycle time is reduced from weeks to hours (BF18), further supported by both SLR and interviews. Products are delivered quickly (BF19), and DevOps facilitates the delivery of higher quality services more efficiently (BF20), with SLR and interview data confirming these benefits. Overall, DevOps practices bring substantial improvements across various aspects of software development and operations, leading to enhanced efficiency, quality, and responsiveness in the software development lifecycle.

Table 7:Identifying SLR and Interviews (DevOps benefits).

ID	Description of DevOps benefits	Sources Type
BF1	Improved software quality and deployment	SLR
BF2	Efficient team collaboration and easy communication	SLR
BF3	Improved business needs responsiveness	SLR
BF4	Increase software reliability	SLR
BF5	Increased code quality	SLR, Interviews
BF6	Easy Customer requirements implementation using software development process.	SLR
BF7	Time to time updates between development team and operations are openly provided.	SLR
BF8	Structures and standards support DevOps and facilitate the development team and operations to accomplish their tasks easily.	SLR
BF9	Software are released frequently.	SLR
BF10	The gap between development and operations teams are overcomes.	SLR, Interviews
BF11	Improvement of the overall efficiency of the organization.	SLR
BF12	The time required for operational activities such as deployment and change control will be reduced.	SLR, Interviews
BF13	Continuous improvement will be promoted, aiding in the automation of tools and processes and adapting swiftly to changes in the software environment.	SLR, Interviews
BF14	It will lead to increased revenue for the organization.	Interviews
BF15	Costs will be lowered by minimizing the need for rework.	SLR, Interviews
BF16	Problems will be identified earlier and resolved more efficiently.	SLR
BF17	The time needed to release software will be shortened.	SLR, Interviews
BF18	The average release cycle time will be reduced from weeks to hours.	SLR, Interviews
BF19	Products will be delivered quickly.	SLR, Interviews
BF20	DevOps facilitates the delivery of higher quality services more efficiently.	SLR, Interviews

Adopting DevOps practices in organizations faces numerous challenges, as highlighted by both Systematic Literature Reviews (SLR) and interviews. Insufficient management structures within organizations (CH1) and inadequate training for DevOps (CH2) are significant impediments, often leading to the hiring of engineers with only system administration or software development experience. The goals and definitions of DevOps adoption are frequently unclear (CH3), and the organizational structure itself can hinder adoption (CH4).

Customer preferences for strict deployment procedures can make DevOps unsuitable for some product development scenarios (CH5). The regional separation of development and operations teams is another barrier (CH6), often exacerbated by the perception of DevOps as a trendy term, leading to resistance (CH7). Software developers may fear an overload of operational duties, reducing their work efficiency (CH8), while the requirement for team members to possess both development and operations skills (CH9) can lead to resistance due to concerns over expertise.

Individuals might focus solely on their specialized areas, showing little interest in other team activities (CH10). An inflexible system architecture can obstruct DevOps practices such as continuous integration, testing, and deployment (CH11). Discrepancies between development, testing, and production environments complicate collaboration and continuous delivery (CH12), while multiple production environments introduce complexity, hindering the use of common tools and processes (CH13). Transitioning to DevOps is particularly challenging with traditional software (CH14).

Defining software architectures compatible with DevOps practices poses a significant challenge (CH15), and the lack of support from senior management can be a critical obstacle (CH16). Employee resistance to change is another common challenge (CH17). External devices and physical peripherals often become incompatible with updated software programs (CH18). Additionally, the visibility of the final released software to customers during requirement tests is complicated (CH19), and the field of embedded systems lacks automated tools for deploying new features (CH20).

Moreover, organizations may struggle with aligning DevOps practices with existing company cultures, which are often resistant to the collaborative and iterative nature of DevOps methodologies. This cultural mismatch can lead to significant pushback from teams accustomed to traditional hierarchical structures and processes, further complicating the adoption of DevOps practices.

These challenges underline the complexities of adopting DevOps practices and the need for comprehensive strategies to address them effectively within organizations.

Table 8:Identifying SLR and Interviews (DevOps challenges).

ID	Description of DevOps challenges	Sources Type
CH1	Insufficient management structure within the organization can impede the adoption process.	SLR, Interviews
CH2	Many organizations lack adequate training for DevOps, leading to the hiring of DevOps engineers based on their experience as system administrators and software developers.	SLR, Interviews
CH3	The goals and definitions of DevOps adoption are often unclear	SLR
CH4	The organizational structure itself can affect the adoption of DevOps.	SLR
CH5	Some customers may prefer different development approaches with strict deployment procedures, which makes DevOps unsuitable for product development in those cases.	SLR
CH6	The regional separation of development and Operations teams can be a barrier to DevOps adoption.	SLR, Interviews
CH7	DevOps might be perceived as a trendy term, leading to resistance from people who see it negatively.	SLR, Interviews
CH8	Software developers might fear that DevOps will overload their tasks with operational duties which reduces their work efficiency.	SLR
CH9	DevOps requires all team members to have both development and operations skills, which might be met with resistance due to concerns over lacking deep expertise in both areas.	SLR, Interviews
CH10	Individuals may focus only on their specialized area and show little interest in other teams activities.	SLR
CH11	An inflexible system architecture can obstruct DevOps practices such as continuous integration, testing, and deployment.	SLR
CH12	Discrepancies between development, testing, and production environments can complicate collaboration and continuous delivery and deployment.	SLR, Interviews
CH13	Multiple production environments can introduce complexity, hindering the use of common tools and processes, affecting continuous delivery.	SLR
CH14	Transitioning to DevOps is more challenging with traditional software.	SLR, Interviews
CH15	Defining software architectures which are compatible with DevOps practices have becomes a challenge to developers.	SLR, Interviews
CH16	Lack of support from senior management can be a significant challenge.	SLR, Interviews
CH17	The resistance to change from employees can also impede DevOps adoption process.	SLR, Interviews
CH18	External Devices and physical peripherals are incompatible with updated software programs.	SLR, Interviews
CH19	Complicated visibility of final released software to customers when running tests of requirements.	SLR
CH20	Embedded systems field Lacks automated tools for deploying new features.	SLR

The table 9 below outlines 20 distinct DevOps solutions, each associated with specific IDs and source types, derived from System Literature Review (SLR) and interview findings. Starting by SL1 suggests that developing a comprehensive management structure to support DevOps adoption is a DevOps solution. This solution is supported by both SLR and interviews, the need for strong leadership and strategic planning. SL2 involves implementing extensive training programs to equip employees with necessary DevOps skills, also confirmed by both SLR and interview data, highlighting the importance of skill development and continuous learning.SL3 focuses on clearly defining the goals and definitions of DevOps adoption within the organization. This is crucial for aligning the entire team's understanding and this solution is supported by both SLR and interviews.

SLR4 recommends restructuring the organization to support DevOps practices, based solely on SLR. Organizational change is necessary to break down silos and foster a DevOps culture. SLR5 aligns customer requirements with DevOps practices to ensure compatibility, with support from both SLR and interviews, stressing the need for integrating customer feedback into development processes. SLR6 emphasizes facilitating closer collaboration between development and operations teams regardless of geographic location. Both SLR and interview data suggest that such collaboration is key to successful DevOps implementation. SL7 advises education employees on the long-term benefits of DevOps to reduce resistance, supported by both SLR and interviews, indicating that awareness and understanding can mitigate resistance to change. SL8 proposes allocating operational duties in a balanced manner to prevent overload on software developers, based on SLR findings this approach helps maintain productivity and prevent burnout.

SL9 recommends providing cross-training to develop both development and operations skills in team members, supported by both SLR and interviews. Cross-functional skills are essential for DevOps success. SL10 encourages inter-team collaboration and participation in various activities, also backed by both SLR and interviews, fostering a collaborative culture within the organization.SL11 suggests designing flexible system architectures that support continuous integration, testing and deployment, with evidence from both SLR and interviews. Flexibility in architecture is crucial for accommodating DevOps practices.SL12 involves standardizing development, testing, and production environments to facilitate collaboration and continuous delivery, supported by both SLR and interviews, ensuring consistency and reliability across environments.

SL13 advocates simplifying the production environment and using common tools to support continuous delivery, based on both SLR and interview data. Simplification can reduce complexity and improve efficiency. SL14 focuses on developing strategies to transition traditional software to DevOps compatible frameworks, supported by both SLR and interviews. Transition strategies are vital for integrating DevOps practices into existing workflows. SL15 involves defining software architectures that are compatible with DevOps practices, with support from both SLR and interviews, ensuring that the architectural foundation supports continuous and delivery. SL16 highlights the need to secure strong support from senior management for DevOps initiatives, based on both SLR and interview data. Management support is critical for the successful implementation of DevOps.

SL17 suggests fostering a culture of change and adaptability among employees to embrace DevOps practices, supported by both SLR and interviews, indicating that cultural change is necessary for DevOps adoption. SL18 involves ensuring compatibility of external devices and peripherals with updated software programs, based on both SLR and interviews, to maintain

seamless integration and functionality. SL19 recommends improving the visibility of final released software to customers during requirement tests, supported by both SLR and interviews, enhanced visibility can lead to better customer satisfaction and feedback. SL20 proposes developing automated tools for deploying new features in embedded systems, based on SLR findings. Automation tools are essential for efficiency and reducing manual errors.

These solutions derived from rigorous analysis, provide a comprehensive guide for organizations looking to adopt DevOps practices effectively. Each solution is backed by substantial evidence from both literature and real-world interviews, ensuring their applicability and relevance.

Table 9: Identifying SLR and Interviews (DevOps solutions)

ID	Description of DevOps solutions	Sources Type
SL1	Develop a comprehensive management structure to support DevOps adoption.	SLR, Interviews
SL2	Implement extensive training programs to equip employees with necessary DevOps skills.	SLR, Interviews
SL3	Clearly define the goals and definitions of DevOps adoption within the organization	SLR, Interviews
SL4	Restructure the organization to support DevOps practices	SLR
SL5	Align customer requirements with DevOps practices to ensure compatibility	SLR, Interviews
SL6	Facilitate closer collaboration between development and operations teams regardless of geographic location.	SLR, Interviews
SL7	Educate employees on the long-term benefits of DevOps to reduce resistance.	SLR, Interviews
SL8	Allocate operational duties in balanced manner to prevent overload on software developers	SLR
SL9	Provide cross-training to develop both development and operations skills in team members	SLR, Interviews
SL10	Encourage inter-team collaboration and participation in various activities	SLR, Interviews
SL11	Design flexible system architectures that support continuous integration, testing, and deployment.	SLR, Interviews
SL12	Standardize development, testing, and production environment to facilitate collaboration and continuous delivery.	SLR, Interviews
SL13	Simplify production environment and use common tools to support continuous delivery.	SLR, Interviews
SL14	Develop strategies to transition traditional software to DevOps compatible frameworks	SLR, Interviews
SL15	Define software architectures that are compatible with DevOps practices.	SLR, Interviews
SL16	Secure strong support from senior management for DevOps initiatives.	SLR, Interviews
SL17	Foster a culture of change and adaptability among employees to embrace DevOps practices	SLR, Interviews
SL18	Ensure compatibility of external devices and peripherals with updated software programs.	SLR, Interviews
SL19	Improve the visibility of final released software to customers during requirement tests.	SLR, Interviews
SL20	Develop automated tools for deploying new features in embedded systems	SLR

The table 10 below, identifies six core DevOps practices, each tagged with their corresponding IDs and supported by both Systematic Literature Review (SLR) and interview data. PR1, which focuses on Continuous Planning, emphasizes the importance of ongoing planning throughout the development cycle. Both SLR and interview sources suggest that continuous planning helps in aligning development activities with business goals and customer needs, ensuring that the project remains on track and adaptable to changes.

PR2 pertains to Continuous Integration, a practice where code changes are frequently integrated into a shared repository. SLR and interviews highlight that continuous integration helps in early detection of integration issues, facilitates regular testing, and improves code quality by encouraging incremental development. PR3 is about Continuous Deployment, the practice of automatically deploying code changes to production. Supported by SLR and interview data, this practice ensures that new features and updates are delivered to users quickly and efficiently, reducing time-to-market and enabling rapid feedback cycles.

PR4 refers to Continuous Testing, which involves the ongoing testing of code throughout the development process. Both SLR and interview findings stress the importance of continuous testing in maintaining code quality, identifying defects early, and ensuring that the software meets the required standards and specifications. PR5, or Continuous Delivery, extends the concept of Continuous Delivery, extends the concept of continuous integration by ensuring that code is always in a deployable state. According to both SLR and interviews, continuous delivery enables organizations to release new features and updates to production with minimal manual intervention, enhancing the reliability and consistency of releases.

PR6 focuses on Continuous Monitoring, the practice of continuously monitoring the performance and health of applications in production. Supported by SLR and interviews, continuous monitoring provides real-time insights into system performance, helps in early detection of issues, and ensures that the system meets performance and availability requirements. These six practices form the backbone of effective DevOps implementation. By integrating continuous planning, integration, deployment, testing, delivery, and monitoring, organizations can achieve greater agility, improve collaboration between development and operations teams, and ensure high-quality software delivery. Each practice is supported by substantial evidence from both literature and real-world interviews, underscoring their importance and applicability in modern DevOps environments.

Table 10: Identifying SLR and Interviews (DevOps Practices)

ID	Description of DevOps Practices	Sources Type
PR1	Continuous Planning	SLR, Interviews
PR2	Continuous Integration	SLR, Interviews
PR3	Continuous Deployment	SLR, Interviews
PR4	Continuous Testing	SLR, Interviews
PR5	Continuous Delivery	SLR, Interviews
PR6	Continuous Monitoring	SLR, Interviews

3.11. Challenges to Validity

The challenges to validity for the interviews are presented in this section:

3.12. Internal Validity

The internal validity challenges to the interview data could be related to selection bias, interview bias and coding reliability.

- Selection Bias: The sample of interviewees and the selection criteria for the codes pose significant risks. If the interview participants were chosen based on convenience or their organization's inclination towards DevOps success, the results might not represent a broader population. Additionally, the inclusion criteria for codes, which focused on pertinence and regularity, could have led to the exclusion of less frequent yet significant practices. To mitigate this threat, the target population was defined as software engineering practitioners who have already experience of working with DevOps. Convenience sampling was used to select the interviewees, by inviting them using personal contacts and their social medias (LinkedIn, e-mails).

- Interview Bias: Participant responses might have been influenced by the desire to provide socially desirable answers, especially if they were aware of the study's objectives. The manner in which interviewers framed questions or reacted to responses could also have caused participants' answers, introducing bias. To mitigate this threat, semi-structured interview questionnaire with open-ended and closed-ended questions, was designed based on SLR findings. All interviewees were asked the same set of questions, and new questions framed and asked spontaneously when needed. Interviewees were promised complete confidentiality to obtain genuine and valid answers.

- Coding Reliability: The subjectivity inherent in qualitative data coding is a notable challenge. Different researchers might interpret interview transcripts differently, leading to inconsistencies. Ensuring high inter-coder is crucial to mitigate this risk and achieve consistent coding results. To mitigate this threat, audio-visuals recording were done in all interviews after obtaining permission from the interviewees. Data transcription was manually done without leaving out any sentence. Thematic coding was done to analyze the collected data, and all transcripts were thoroughly read to familiarize with the data.

3.13. Construct Validity

A challenge to the validity of the interview data arises from the interview questionnaire potentially not aligning with the research goals. To address this issue, the questionnaire was created based on findings from the systematic literature review (SLR) and was continuously refined. Additionally, after the interviews, participants were asked to confirm that the questions were clear and relevant to further ensure the questionnaire's effectiveness.

3.14. External Validity

The challenge to external validity concerns the generalizability of the findings. Since practices and solutions that work well in one organization or industry might not be relevant in another due to varying organizational cultures, structures, and industry requirements, this poses a problem. To address this, interviewees with diverse IT background, experiences and roles were selected, and the number of interviewees from any single organization was limited to two.

3.15. Validity of Data Sources

Both interviews and the systematic literature review (SLR) rely on self-reported data, which can be prone to inaccuracies or exaggerations. The reliability of the SLR also depends on the thoroughness of the literature search and the quality of the selected studies. To address this, a systematic approach was adopted in both literature review and the selection of interview participants.

3.16. Temporal Validity

DevOps practices and solutions are continuously evolving. What is effective today may not be applicable in the future, making it essential to regularly update the findings to maintain their relevance and applicability. Regular updates and reviews of the findings are necessary to ensure ongoing relevance.

3.17. Interpretation and Implementation

Misinterpretation of the coded data could result in inaccuracies. The practical feasibility of implementing some of the identified solutions in various organizational contexts might also pose challenges, despite their theoretical soundness. To address this, clear documentation of the coding process and decision-making was maintained to allow reproducibility and verification.

3.18. Quantitative Research Method

3.19. The Design of the Survey Questionnaire

In this study, an online survey was used as the quantitative research method to assess the transformational impact of DevOps pre and post-adoption in IT companies and startups. The decision to employ a survey was informed by its ability to reach a broad population of individuals, as highlighted by Shull et al. [47]. This approach allowed me to gather quantitative data from a larger population sample and compare the findings gained. The survey process involved several key steps, including defining the survey objectives, planning, designing, validating the questionnaire, collecting, and analyzing data [47]. The questionnaire was developed using Google Docs, the online word processor that contains Google Forms, and reviewed and validated by three DevOps practitioners before being distributed to the target population. To ensure a diverse respondent pool, the survey was disseminated through personal contacts, professional networking platforms like LinkedIn, and other social media as well as various online groups and forums dedicated to DevOps. Sampling was employed, resulting in a complete response of 45 received, which were considered for statistical analysis. The statistical analysis methods applied in our study included descriptive analysis [47]. Descriptive statistics were used to analyze the quantitative data gathered through the survey. Creswell et al.[48],emphasizes that the importance of ensuring the sample for the quantitative phase differs from other methods like the qualitative method.

The survey questionnaire used in this research was designed based on different objectives, with both open-ended questions and closed questions. The questions were composed based on the demographics of the survey respondents and the focused study. The responses from the survey were stored in an Excel sheet file and this favored data analysis practices. The survey includes questions about different aspects of organizations and their experiences with DevOps. Here is a breakdown of what each section of the questionnaire covers:

- Demographic questions: These questions ask about the organization's type location, size, the respondent's role, and their experience both overall and specifically with DevOps.

- Benefits of Implementing DevOps: This section asks about the benefits organizations

have experienced after adopting DevOps. Respondents can choose from a list of benefits identified from previous research and interviews. They also have the opportunity to add any additional benefits they've observed.

- Challenges Faced in DevOps Adoption: Here, respondents are asked about the challenge's organizations encounter when adopting DevOps. They can choose from a list of challenges identified from previous research and interviews. They can also mention any other challenges they've faced. The importance of each challenge is rated on a scale.

- Practices of DevOps: This section asks about the practices organizations should include when adopting DevOps. Respondents can choose from a list of practices identified from previous research and interviews. They can also suggest any other practices they believe are important.

  Problems Faced During, Deployment, and Testing: Finally, respondents are asked about the specific problems DevOps teams encounter during these processes. They can choose from a list or problems identified from interviews or mention any other issues they have encountered.

All questions related to demographics and the study are required to ensure complete responses. At the end of the survey, there's a text box for respondents to provide their names and email addresses if they wish to receive a compilation of the survey results.

3.20. Sampling of respondents

To ensure that this research accurately reflects the experiences of those familiar with DevOps, I first needed to define our target population, as noted by Shull et al[47]. This target population consists of software engineering practitioners who have experience with DevOps methodologies. It's crucial for a sample to be representative of this target population. Since the target population is specific and limited, relying on random sampling methods is challenging, instead I've employed a random sampling technique to select survey respondents. A convenience method were utilized as a method which involves gathering responses from individuals who are available and willing to participate in the research study. The convenience sampling is a widely used method in research, particularly when access to the target population is limited or when time and resources are restricted. In the context of DevOps research, convenience sampling offers an efficient way to gather insights from software engineering practitioners who have experience with DevOps methodologies. This research explores the explanation behind the use of convenience sampling, its advantages, limitations, and implications for the validity of our findings.

Convenience sampling offers several advantages in the context of DevOps research. Firstly, it allows for quick and cost-effective data collection, making it suitable for studies with limited time and resources. Secondly, convenience sampling facilitates access to a wide range of participants from various organizational backgrounds, geographic locations, and levels of experience in DevOps. This diversity enriches the data by capturing a broad spectrum of perspectives and insights, enhancing the depth and richness of the findings. Furthermore, a convenience sampling aligns with the practical realities of conducting research in a dynamic and a rapidly evolving fields such as DevOps. The flexibility and adaptability of convenience sampling enable researchers to respond promptly the changing circumstances and emerging opportunities for data collection. This agility is essential for capturing timely and relevant insights into contemporary DevOps practices and challenges.

While convenience sampling offers numerous benefits, it is not without limitations. One key concept is the potential for sampling bias, as participants may not be representative of the broader population of software engineering practitioners. Additionally, convenience sampling may overlook individuals who are less accessible or inclined to participate in research, leading to underrepresentation of certain perspectives or experiences. To mitigate these limitations, researchers must exercise caution in interpreting and generalizing the findings obtained through convenience sampling. It is essential to acknowledge the fundamental biases and limitations of sampling method and consider alternative perspectives or supplementary data sources to validate the findings. Transparency and reflexivity in reporting the sampling approach and its implications for the study's validity are critical for ensuring the credibility and trustworthiness of the research outcomes. Initially, the population sample was drawn from personal contacts and practitioners found on professional networking platforms such as LinkedIn, and direct emails contacts. The survey link was shared across various online forums and groups dedicated to DevOps, respondents were encouraged to forward the survey to their own contacts who can provide valuable insights**[47]**.

3.21. Data Collection

To gather the necessary data for this research, a quantitative questionnaire was utilized, by using the online tool Google Forms, the questionnaire was efficiently designed and disseminated via a shareable link. Initially a questionnaire was distributed to 3 test respondents to gather feedback and refine the final questionnaire. The final version consists of 24 questions and was estimated to take participants approximately 20 minutes to complete. The questionnaire was distributed via email and other media channels to all employees. Although participation was voluntary, efforts were made to achieve a target response rate of at least 30 respondents completed surveys. Initially, the response rate was low during the beginning few weeks. To encourage participation, I've sent an alert or reminder messages and emails several times prompting employees to take part. Over a period of 5 weeks, a total of 45 responses were collected. Subsequently, data collection ceased, and analysis commenced on the gathered information. This methodology aligns with the focus of the thesis, which examines the transformative impact of DevOps adoption in IT companies and startups through an analysis of DevOps pre-and post-adoption phases.

This part of the thesis includes data analysis of all survey respondents, their interpretations, and visualizations with the help of tables, charts, and graphics.

4. Data Analysis and Results

For analyzing the collected data using the survey questionnaire results, I have retrieved the most important questions or variables that align with the objectives of this research study and are relevant to answering the research question. Those questions results are analyzed and displayed in visualization charts below:

4.1. Analysis of the companies where respondents are employed

The survey requested respondents to indicate the type of companies they are working for. According to Appenddix_1, Question 1, a comprehensive breakdown reveals that the majority of participants, 43 in total (95.6%), are employed by IT consultancy companies. Similarly, a substantial portion, 42 respondents (93.3%), are engaged in roles within software development companies, with an equal number also employed in scientific software companies. Furthermore, 41 participants (91.1%) are associated with software backplane companies, which indicates a significant presence in this sector. The analysis continues to unveil the distribution of respondents across various sectors, with 39 individuals (86.7%) contributing to artificial intelligence startups, and 38 respondents (84.4%), involved in scientific creativity roles, emphasizing the synergy between technology and innovation. Moreover, 35 participants (77.8%) are employed by design and engineering software companies, reflecting the emphasis on design-driven solutions in the industry. Further examination reveals that 34 respondents (75.6%) work in information display and transaction entry companies, underlining the importance of user interface and experience. Additionally, 33 individuals (73.3%) are focused on consumer-oriented software, emphasizing the shift towards user-centric development approaches. The analysis also highlights the distribution of respondents across niche areas such as operations systems (27 participants, 60%) and information management and manipulation (26 participants 57%), indicating specialization within the industry. Furthermore, 24 respondents (53.3 %) are involved in server-related roles, while 22 individuals (48.9 %) contribute to artistic creativity aspects in their respective companies.

Additionally, the analysis showcases the involvement of respondents in critical areas such as device/peripherals drivers (20 participants, 44.4%) and business-oriented software and hardware control (18 participants, 40 %), indicating the diverse range of expertise within the surveyed population. Furthermore, 9 respondents (20 %) are engaged in networking/ communications and operations research, highlighting the intersection of technology and strategic planning. Moreover, a smaller cohort of respondents, 7 individuals (15.6%), are employed in support utilities, middleware, system components, and malware companies, indicating the importance of infrastructure and security. Additionally, 6 participants (13.1%) are involved in real-time control software, while 5 respondents (11.1 %) contribute to process control software, underlining the significance of precision and efficiency in industrial settings. Overall, the analysis provides a comprehensive overview of the diverse landscape of companies and startup represented by the respondents, the figure below displays all types of companies and their workers.

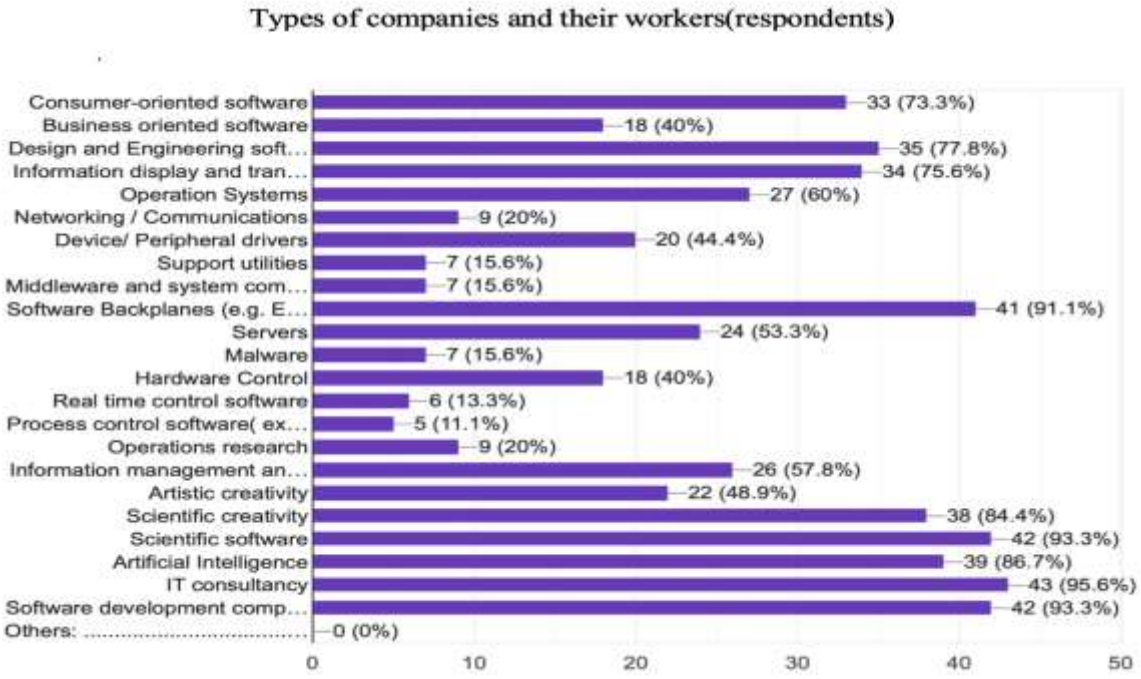


Figure 5: Types of companies and their workers

4.2. Size of companies of the survey respondents

The examination of the survey data see Appenddix_1, Question 3. reveals interesting insights into the sizes of companies represented by respondents. The majority of participants with a substantial of 86.7%, indicated that they work for organizations with a workforce ranging from 10 to 50 employees. This finding highlights the prevalence of small to medium-sized enterprises among the surveyed group. Furthermore, a significant portion of respondents, comprising 68.2%, reported working for companies with fewer than 10 employees. This suggests a notable presence of micro-enterprises or startups.

Interestingly, only one respondent identified himself as working for a large company, with an employee count ranging from 50 to 5000 individuals. While this constitutes a minority within the sample, it indicates the presence of larger corporation alongside the small and medium-sized enterprises. In summary, our analysis underscores the predominance of small to medium-sized enterprises among survey respondents, with a minority representing both micro-enterprises and larger corporations. This provides valuable insights into the diversity of company sizes within our surveyed population.

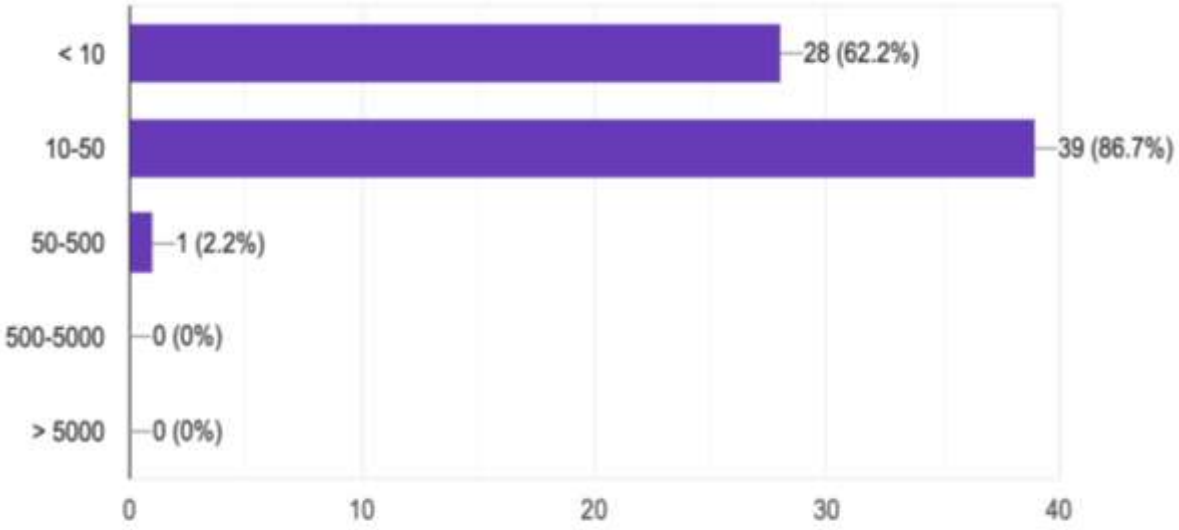


Figure 6: Size of companies of the survey respondents

4.3. Experience with DevOps practices

This research study focuses on understanding how DevOps and a set of practices in software development, can contribute to companies' development [22]. It's really important that respondents participating in this survey see Appenddix_1, Question 6, know a lot about DevOps because their experience is the key. So, they were asked how long they have been working with DevOps.

40% of the participants who responded said they have been using DevOps for about 3 years. This is the majority group and it shows that respondents have quite a bit of experience with DevOps. Then, about 28.9% of respondents said they have been using DevOps for around 2 years. This also is a significant number and shows that respondents are experienced and know exactly a lot about DevOps. Another group, about 20%, mentioned that they have been using DevOps for 4 years. This group represents a good part of the survey respondents too.

Some others, around 6.7%, mentioned they have used DevOps for 5 years, which is less but still notable. Then, there is only one respondent, about 2.2% said he or she used DevOps for a year, and another 2.3% said he/she used DevOps for almost 6 years. So, in short, the analysis shows that respondents who participated in this survey have different levels of experience with DevOps. This variety in experience is important to understand how DevOps is affecting different organizations. Throughout the visual chart below all employees experiences with DevOps practices are displayed.

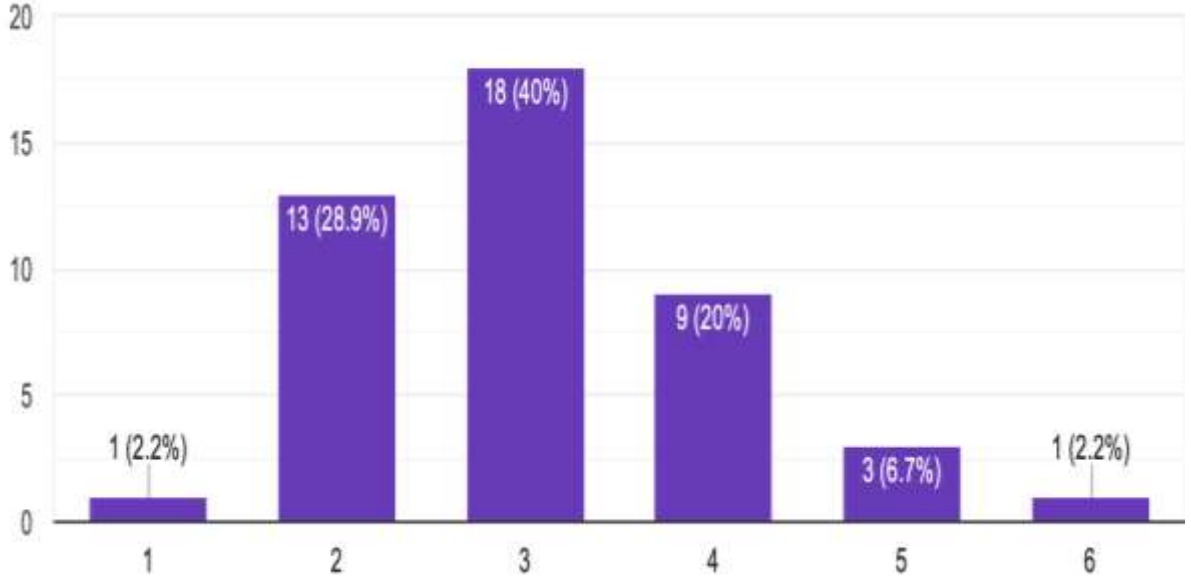


Figure 7: Experience with DevOps practices

4.4. Analysis of development methodologies companies used before adopting DevOps

The analysis reveals the development methodologies prevalent among respondents'organizations, prior to embracing DevOps practices. Notably, the majority of respondents see Appenddix_1, Question 7, comprising 41 individuals (91.1%), indicating a prior reliance on agile methodologies. This suggests a significant inclination towards agile principles before the adoption of DevOps. Additionally, a considerable proportion of respondents, numbering 34 (75.6 %), reported utilizing the waterfall model, signifying the substantial pressure of traditional, sequential development approaches in their organizations. Moreover, 24 respondents (53.3 %) acknowledged employing scrum methodologies prior to embracing DevOps. This finding underscores the widespread adoption of iterative and incremental development practices, often associated with scrum in the pre-DevOps landscape.

Interestingly, 23 respondents (51.1 %) declared employing a hybrid methodology, indicating a blend of different development approaches tailored to their organizational needs. This highlights a nuanced approach to software development, where organizations leverage a combination of methodologies to address specific project requirements and challenges. Overall, the analysis represent a diverse landscape of development methodologies prevalent in respondents' organizations before transitioning to DevOps. While agile methodologies emerge as the dominant choice, the presence of waterfall, scrum and hybrid approaches underscores the varied strategies employed by organizations to manage their software development processes. See below figure, is the visualization of used methodologies before adopting DevOps.

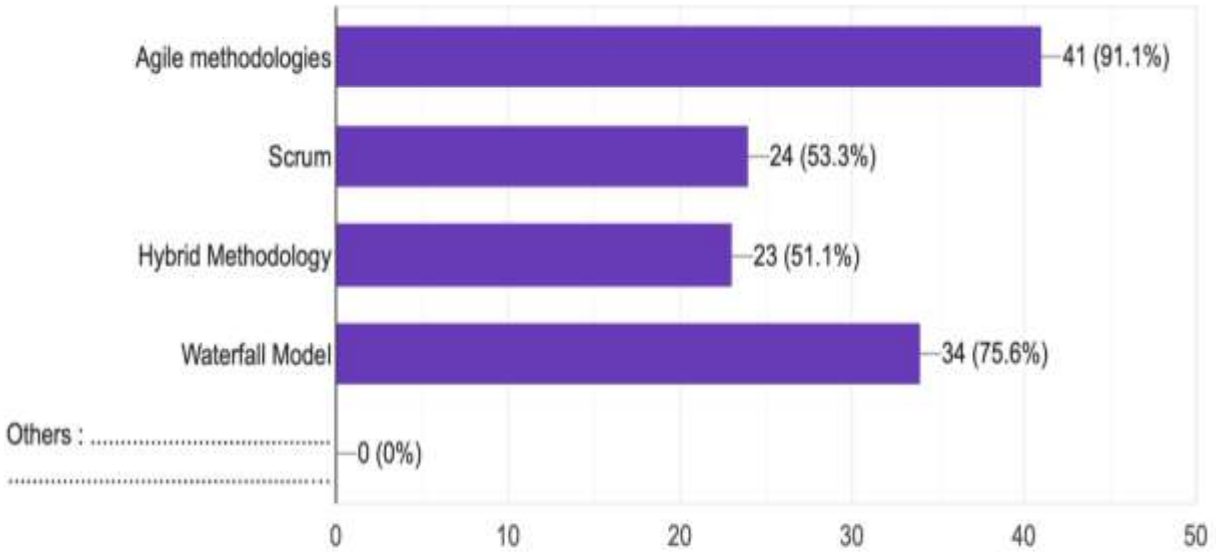


Figure 8: Analysis of development methodologies

4.5. Transformational Impact of DevOps after Adoption in IT companies (Post-Adoption)

Analyzing the transformational impact of DevOps in IT companies is one of the objectives of this research study and according to the survey results see Appenddix_1, Question 8, agile methodology is the most used practice before adopting DevOps. Therefore, respondents were asked to identify the benefits that they think should be achieved after implementing DevOps in their organizations. Additionally, they were asked to specify any additional benefits of adopting DevOps, which were not listed in the options. The overview of responses shows that 41 respondents with (91.1%) identified DevOps as a tool to facilitate software development automation, 38 respondents with (84.4%) declared that DevOps improved software quality. 37 respondents with (82.2%) revealed that DevOps benefits increased the rapid product delivery, 35 respondents with (77.8 %) declared that it increased revenue and efficiency, business responsiveness, and improved workflow efficiency.

33 respondents with (73.3%) have declared that DevOps elevated code quality, 32 respondents with (71.1%) said the adopting DevOps minimized manual intervention,26 respondents with (57.8%), 25 respondents with (55.6%) declared that DevOps enhanced cost-effective solutions, 24 respondents (53.3%), enhanced collaboration and accelerate software release cycles, 21 respondents with(46.7%) said that it enhanced tracking and failure detection, 20 respondents with (44.4%) revealed that it enhanced system stability,18 respondents with (40%) said that DevOps benefited in customer centric approach, and finally 17 respondents with (37.8%) enhanced customer satisfaction and expedited feedback loops.

The figure below visualizes expected benefits of adopting DevOps in IT organizations

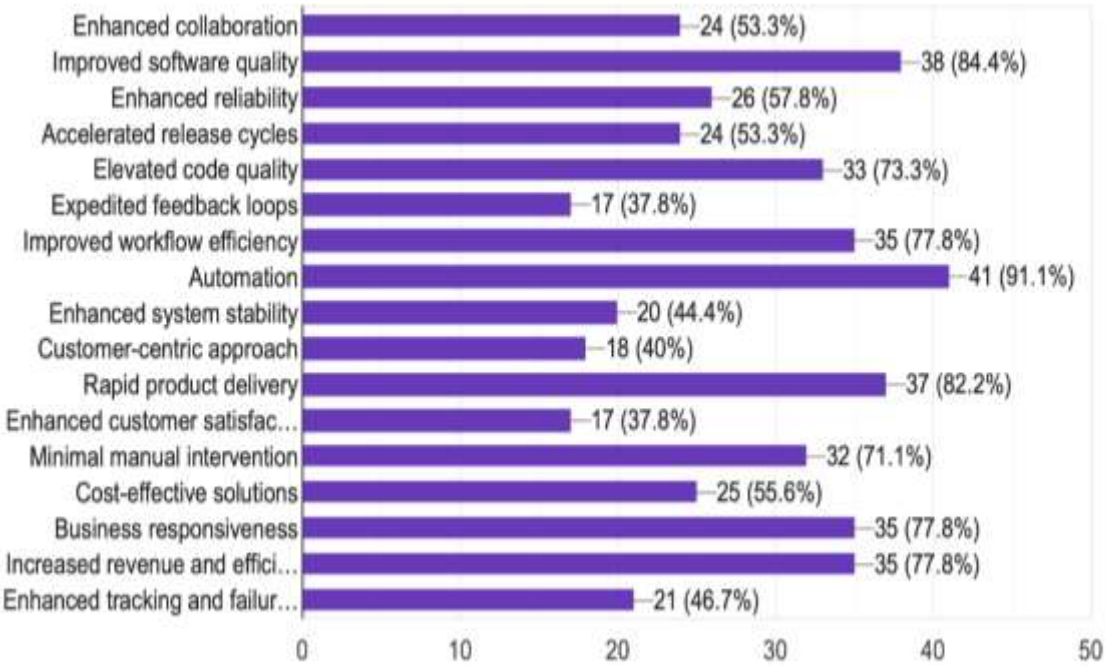


Figure 9: Expected benefits of adopting DevOps

4.6. Challenges faced by Organizations while adopting DevOps

To discern the obstacles encountered by organizations during the implementation of DevOps is another aim of this study. The survey participants were queried regarding the perceived challenges they faced when initiating DevOps within their respective organizations see Appenddix_1, Question 11. Responses were elicited through a rating system employing ordinal numbers, ranging from 1 up 5, to show the extent of difficulty associated with DevOps adoption. Furthermore, respondents were asked to enumerate the specific challenges encountered during this process. The identified challenges were presented as response options within the survey. Additionally, participants were requested to articulate the degree of difficulty associated with each challenge, utilizing an ordinal scale which have options such as “very challenging”, “challenging”, “somewhat challenging”, “not challenging” and cannot specify. The figure below showcases the challenges which faced by organizations while adopting DevOps, and the difficulty levels which were exposed by the survey respondents.

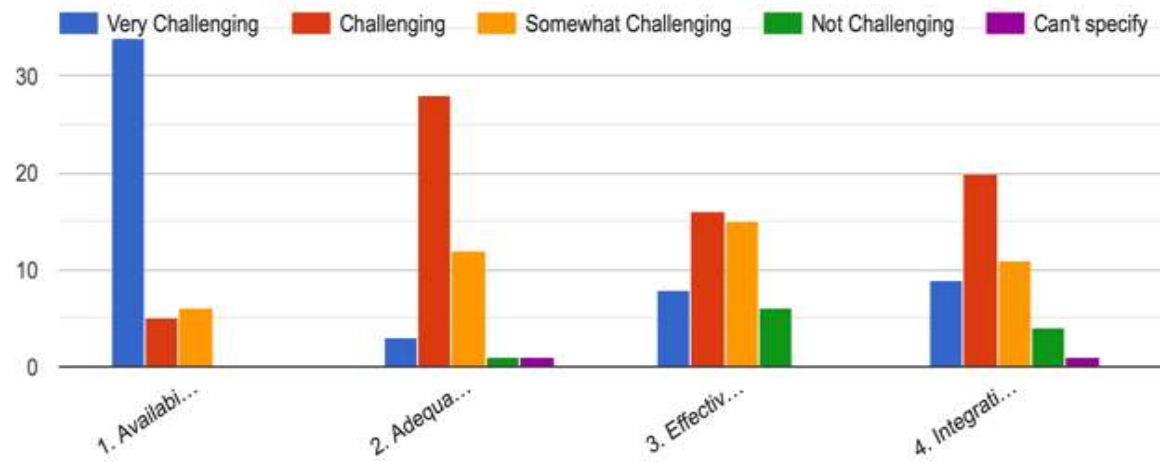


Figure 10: Challenges faced by organizations.

Aspects like availability of reliable resources for environment configuration, adequacy of the current management structure, integration of new tools with existing company tools, employee resistance towards DevOps implementation, dealing with legacy software, alignment of organizational structure with DevOps methodologies, understanding complexities and cost associated with DevOps, management of complexity due to multiple system, designing software architectures to align with DevOps principles, managing technical debt within the DevOps framework, perception of DevOps as a buzzwords, clarity of definitions and goals of DevOps, security clearances when handling sensitive data in DevOps process, support from customers towards DevOps practices, fear of change among employees during DevOps implementation, learning curve associated with DevOps adoption, employees fear of measurement in DevOps performance evaluation, availability of proper automation tools in the domain of embedded systems all those aspects are shown as challenging to the organizations after DevOps adoption.
Therefore, Managerial fear of failure in DevOps adoption, making everyone accustomed to a new workflow, Understanding of applications and environments, especially in outsourced projects, Familiarization with new workflows introduced by DevOps, Utilization of custom scripts for task automation in DevOps, Availability of resources for configuring environments in DevOps, visibility of production environments particularly in embedded systems seems, effective management of geographical distribution of Dev and Ops teams to be not challenging aspects.

Consequently, organizations must contend with managerial apprehensions regarding the potential failure of DevOps adoption, the necessity to familiarize every member with a new workflow, gaining a thorough understanding of applications and environments, particularly in outsourced projects, acclimatizing to the new workflows introduced by DevOps. Leveraging custom scripts for task automation, securing adequate resources for configuring DevOps environments, and ensuring visibility into production environments, particularly within embedded systems. These facets collectively represent significant difficulties for organizations navigating the landscape post-DevOps adoption.

4.7. Strategies that can be employed to mitigate the challenges of DevOps adoption

To ascertain the strategies conducive to ameliorating the challenges associated with DevOps adoption, respondents actively participated in survey inquiries as outlined in Appenddix 1, Question 13. The findings revealed the majority of participants, with 37 respondents, constituting 82.2% of the sample, advocating for the implementation of a flexible architecture as a viable approach to mitigating the difficulties which inherent in DevOps adoption. Furthermore, 35 respondents (77.8%) underscored the importance of recruiting individuals with a DevOps-oriented mindset and instituting robust documentation practices for errors and their corresponding resolutions.

Moreover, 34 respondents (75.6 %) emphasized the significance of promptly addressing emerging issues, while 32 respondents (71.1 %) recommended commencing DevOps initiatives with smaller-scale projects. Additionally, 29 respondents (64.4 %) highlighted the utility of establishing proof of concept (POC) frameworks and cultivating a structured management framework to navigate the challenges associated with DevOps integration.

Furthermore, 27 respondents (60%) emphasized the importance of conducting comprehensive cost and risk assessments prior to embarking on DevOps adoption initiatives. Similarly, 26 respondents (57.8%) emphasized the critical role of adequately training personnel to effectively confront post-DevOps implementation challenges, alongside the inclusion of quality assurance personnel in the process.

Additionally, 25 respondents (55.6%) advocated for the implementation of brief, regular meetings to foster inter-team communication, with 24 respondents (53.3%) proposed the delineation of project release criteria and the documentation of encountered challenges as effective strategies for minimizing post-DevOps adoption challenges within organizations. Finally, 22 respondents (48.9 %) identified that specifying release criteria for every project and document the challenges that popup.

The figure below demonstrates all strategies possible to mitigate the challenges of DevOps

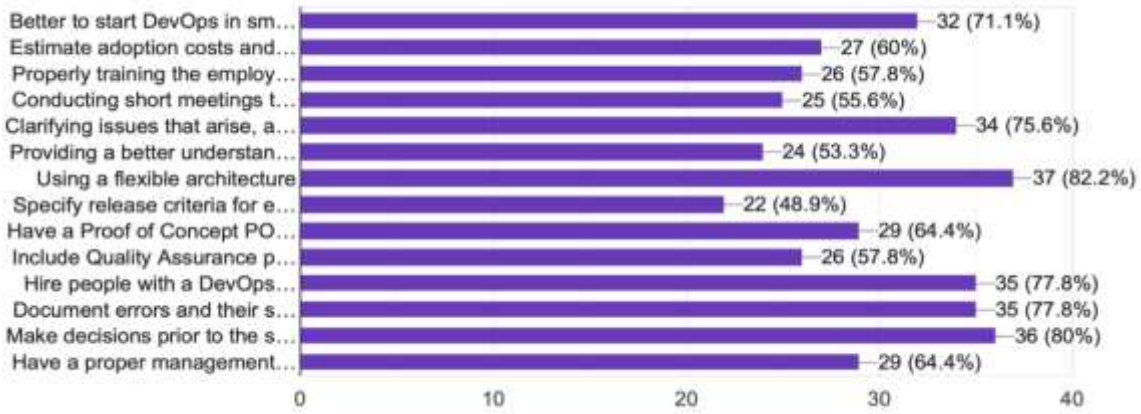


Figure 11: strategies possible to mitigate the challenges of DevOps

4.8. DevOps practices companies should incorporate into their development processes.

To identify various practices essential for DevOps adoption, survey respondents were tasked with providing a spectrum of practices suitable for their respective organizations. The practices identified by respondents are considered as potential options. Among the respondents, 45 individuals, constituting 100% of the total, acknowledged that Git and GitHub Collaboration, along with Continuous Delivery, are integral DevOps practices that can be implemented within their organizations. Furthermore, 44 respondents (97.8%) highlighted Continuous Integration as crucial DevOps practice. Azure Pipelines and Continuous Testing were identified by 43 respondents (95.6 %) as essential practices. Lastly, 37 respondents (82.2 %) expressed interest in incorporating Containerization as a DevOps practice with their organizations.

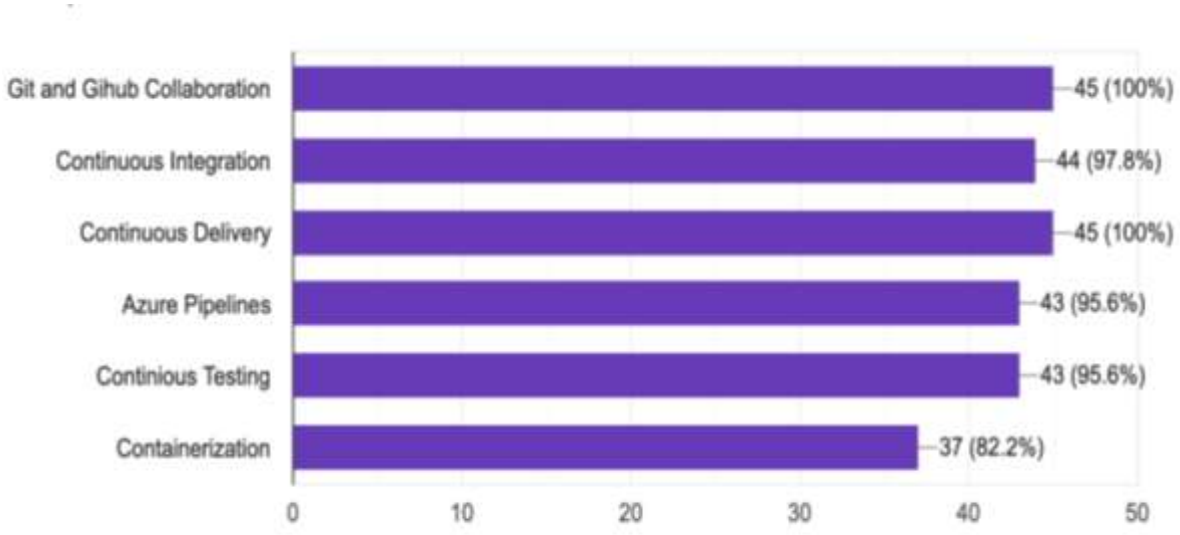


Figure 12: DevOps practices to be included in organizations

The conclusion and further research study section is the part which summarize the key findings and implications of the study. Additionally, suggestions for further research venues or areas for exploration are presented, and providing potential directions for future studies to build upon the current research findings.

5. Conclusion and Further Research Study

The research conducted delves into the impact of DevOps before and after its adoption, which has accumulated significant attention in recent years, particularly with major industry players like Netflix and Yahoo embracing its principles. However, the existing literature on DevOps remains limited and often of questionable quality. This research seeks to enhance our comprehension of DevOps implementation within organizational settings. The focus of this study has identified the challenges, solutions, practices, and problems associated with DevOps in organizations before and after its adoption. A mixed method approach has been employed using quantitative validation through an online survey and a qualitative method through interviews and SLR. Subsequently, the survey and interviews were conducted to validate these findings and uncover additional insights.

Moving forward, there is a clear need for more empirical research in the field of DevOps, particularly focusing on the contextual factors influencing its adoption and effectiveness. Case studies can help to validate findings across different organizational settings, while investigating the impact of cultural factors on DevOps adoption warrants further exploration. Additionally, future research should delve into the tools required to support DevOps practices and explore the organizational changes necessary for a successful transaction to DevOps. In conclusion, this study emphasizes the importance of informed decision-making regarding DevOps adoption, cautioning against, the associated risks and challenges. By shedding light on the benefits, challenges, solutions, practices, and problem of DevOps adoption, this research aims to provide valuable insights for both researchers and practitioners in the software industry.

5.1. Further Research study

Further research endeavors should concentrate on conducting empirical investigations to validate findings across varied organizational contexts and investigate, the causing factors influencing DevOps adoption. Additionally, there is a need to delve deeper into the tools supporting DevOps and organizational changes required for successful adoption. Further exploration of the impact of cultural factors on DevOps adoption and solutions to overcome adoption of the impact challenges is warranted. Additionally, investigating the necessary changes for a complete transition to a DevOps state and assessing the availability and functionality of DevOps tools are avenues for future research.

References:

[1]. Ramtin Jabbari, (2016) “What is DevOps?” In: Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP ’16 Workshops. New York, USA: ACM Press, pp. 1–11. isbn: 9781450341349. doi: 10.1145/2962695. 2 962 707. url: http://dl.acm.org/citation.cfm?doid=2962695.2962707.

[2]. Patrick Debois,(2011) “Devops: A Software Revolution in the Making?” In: Cutter IT Journal 24.8 , pp. 1–40. issn: 15227383. url: https: / / www.cutter.com/sites/default/files/itjournal/fulltext/2011/08/ itj1108.pdf.

[3]. 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.

[4]. 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.

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

[6]. 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.

[7] Gartner. (2022). DevOps. Retrieved from https://www.gartner.com/en/information-technology/glossary/devops

[8]. Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press.

[9] .Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Pearson.

[10]. Kim, G., Behr, K., & Spafford, G. (2013). The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution Press.

[11]. Arvedahl, M,& Åkersten,C.(2018). Exploring the Criticality and Impact of DevOps Practices. Master's thesis, Departement of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, Sweden.

[12]. Škurla, A. (2020). DevOps integration of Security Practices. Master's thesis, Faculty of Informatics, Masaryk University, Brno.

[13]. Pelmas, D., & Lejon, L. (2023). Exploring the interfaces and the challenges. Master's thesis, School of Engineering, Jönköping University, Jönköping, Sweden.

[14]. Amaradi, A. S., & Nutalapati, S. B. (2016). Continuous Integration, Deployment and Testing in DevOps Environment. Master's thesis, Faculty of Computing, Blekinge Institute of Technology, Karlskrona, Sweden.

[15]. Sohrab, C., & Bao, K. P. (2021). Automated Software security activities in a continuous delivery, Master's thesis, Facullty of Science and Technology, University of Stavanger, Norway.

[16]. M. Hüttermann. DevOps for Developers. Apress 2012, 1st edition, 2012.

[17]. Randell, B. (1986). The 1968/69 NATO software Engineering Reports. History of Software Engineering, Page 37.

[18]. Hamilton, M. and Zeldin, S.(1976). Higher order software a methodology for dening software. IEEE Transactions on software Engineering, SE-2(1):9_32.

[19]. Bell, T.E. and Thayer, T.A.(1976). Software Requirements: Are they really a problem? In Proceedings of the 2nd international conference on Software engineering, pages 61_68. IEEE.

[20]. Boehm, B. W.(1988). A spiral model of software development and enhancement. Computer, 21(5):61_72.

[21]. Sommerville, I. (1992). Software Engineering. Addison-Wesley, 4th edition. New York.

[22]. P. kuvaja. L.E. Lwakare and M. Oivo, "Relationship of DevOps to agile lean and continuous deployment.," in international conference on product-focused software process improvement. 2016.

[23]. L.E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo and C. Lassenius. "DevOps" in practice: A multiple case study of five companies," Information and software technology, Vol. 114. Pp. 217-230, 2019.

[24]. Anastasia ,k.(2019).What is DevOps and where is it applied, visited on: https://shalb.com/blog/what-is-devops-and-where-is-it-applied/

[25]. K. Nybom, I. Porres och J. Smeds,(2015) "DevOps: a definition and perceived adoption impendiments." Internaltional conference on agile software development.

[26]. S. Sharma, The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise, John Wiley & Sons, 2017.

[27]. Royce. W.(1970). "Managing the Development of Large Software Systems." Proceedings of IEEE WESCON.

[28]. Beck, k.(2001). "Manifesto for Agile Software Development." Agile Alliance.

[29]. Kim, G., (2016). "The DevOps Handbook: How to create World-Class Agility, Reliability, & Security in Technology Organizations." IT Revolution Press.

[30]. Juric, A. (2021-06-09). "Lean DevOps for Automotive OEMs." Master's Thesis (No. 2021:63). Department of Mechanics and Maritime Sciences, Chalmers University of Technology, SE-412 96 Göteborg, Sweden. Page 4.

[31]. Max Rehkpf, Manifesto for agile software development, 2001.
Accessed on: https://agilemanifesto.org/

[32]. Kuda Nageswara Rao, G kavita Nidu, and Praneeth Chakka. A study of the Agile software development methods, applicability and implications in industry. International Journal of Software Engineering and its applications, 5(2):35-45,2011.

[33]. Alan Z, (2024). Agile, Project Management, Effective Time Management for Project Teams.

[34]. Sims, C., & Johnson, H.L.(2011). The Elements of Scrum. Dymaxicon.

[35]. Srivastava, A., Bhardwaj, S., Saraswat, S. (2017). SCRUM model for agile methodology. International Conference on Computing, Communication and Automation (ICCCA).

[36]. Alamy,(2024).DevOps word cloud concept. Accessed on: https://www.alamy.com/devops-word-cloud-concept-on-white-background-image356735998.html.

[37]. E. Laukkanen, J.Itkonen, C. Lassenius,(2017) Problems, causes and solutions when adopting continuous delivery a systematic literature review, information and software Technology 82,55-79.

[38]. Lianping Chen.(2015) Continuous delivery: Huge benefits, but challenges too. IEEE Software, 32,03. doi: 10.1109/MS.

[39]. Moitaba Shahin, Muhammad Ali Babar, and Liming Zhu. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. IEEE Access, 5:3909-3943, 2017. Doi: 10.1109/ACCESS.2017.2685629.

[40]. Harish G, (2015). API Driven development, bridging the gap between providers and Consumers.

[41]. Håvard Myrbakken and Ricardo Colomo-Palacios. DevSecOps. DevSecOps: a multivocal literature review. In international Conference on Software Process Improvement and Capability Determination, pages 17-29. Springer, 2017.

[42]. Roshan. R., Mansooreh. Z., Haifeng, Shen, (2021). Challenges and Solutions when adopting DevOps: A systematic review.

[43]. Chacon, S., Straub, B. (2024). Pro Git. Retrieved from: https://git-scm.com/book/en/v2

[44]. Loeliger, J., McCullough, M. (2012). Version Control with Git. Retrieved from : https://www.oreilly.com/library/view/version-control-with/9781449345037/

[45]. Swicegood, T. (2008). Pragmatic Version Control Using Git. Retrieved from: https://pragprog.com/titles/tsgit/pragmatic-version-control-using-git/

[46]. Docker. (2024). Docker Documentation. Retrieved from: https://docs.docker.com/get-started/overview/#what-can-i-use-docker-for

[47]. Shull. F, Singer. J. (2007). Guide to Advanced Empirical Software Engineering. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[48]. Creswell, J. (2013). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publictions.

[49]. Nikolaos Pandis, (2016). Associate Editor of Statistics and Research Design, Bern, Switzerland, and Corfu, Greece.

[50] .The survey form accessed on: https://docs.google.com/forms/d/e/1FAIpQLSeqiSzjfH0PIXNoAYOXniFvs8T2RQEC1HPLrbBHh87Gc9GGzg/viewform?usp=pp_url

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

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

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

[54]. PRISMA (2020). PRISMA 2020 Explanation and Elaboration: Updated Guidances and Exemplars for reporting systematic reviews. [online] Available at: http:// www.prisma-statement.org/ [Accessed 29 May 2024].

[55]. Mark. S and Jose C. A DevOps approach to integration of software components in a research project. In Proceedings of the 1st International Workshop on Quality-Aware DevOps, QUDOS 2015, pages 1–6, New York, NY, USA, 2015. ACM.

[56]. Manish. V. 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.

[57]. Matt. C and Alexandra. S (2016) DevOps: Making it easy to do the right thing. IEEE Software, 33(3):53–59.

[58]. Floris. E, C. Amrit, and M. Daneva (2014). 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.ACM.

[59]. Kristian. N, J. Smeds, (2016). *On the Impact of Mixing* Responsibilities Between Devs *and Ops*, pages 131–143. Springer International Publishing, Cham.

[60]. Bjørnar. T and Jon. I. (2008.) Cooperation between developers and operations in soft- ware 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. ACM.

[61]. S. Jones, J. Noppen, and F. Lettice, (2016). 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, ACM.

[62]. Lucy. E. L, 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.

[63]. Brian. F and K. Stol. (2014). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*.

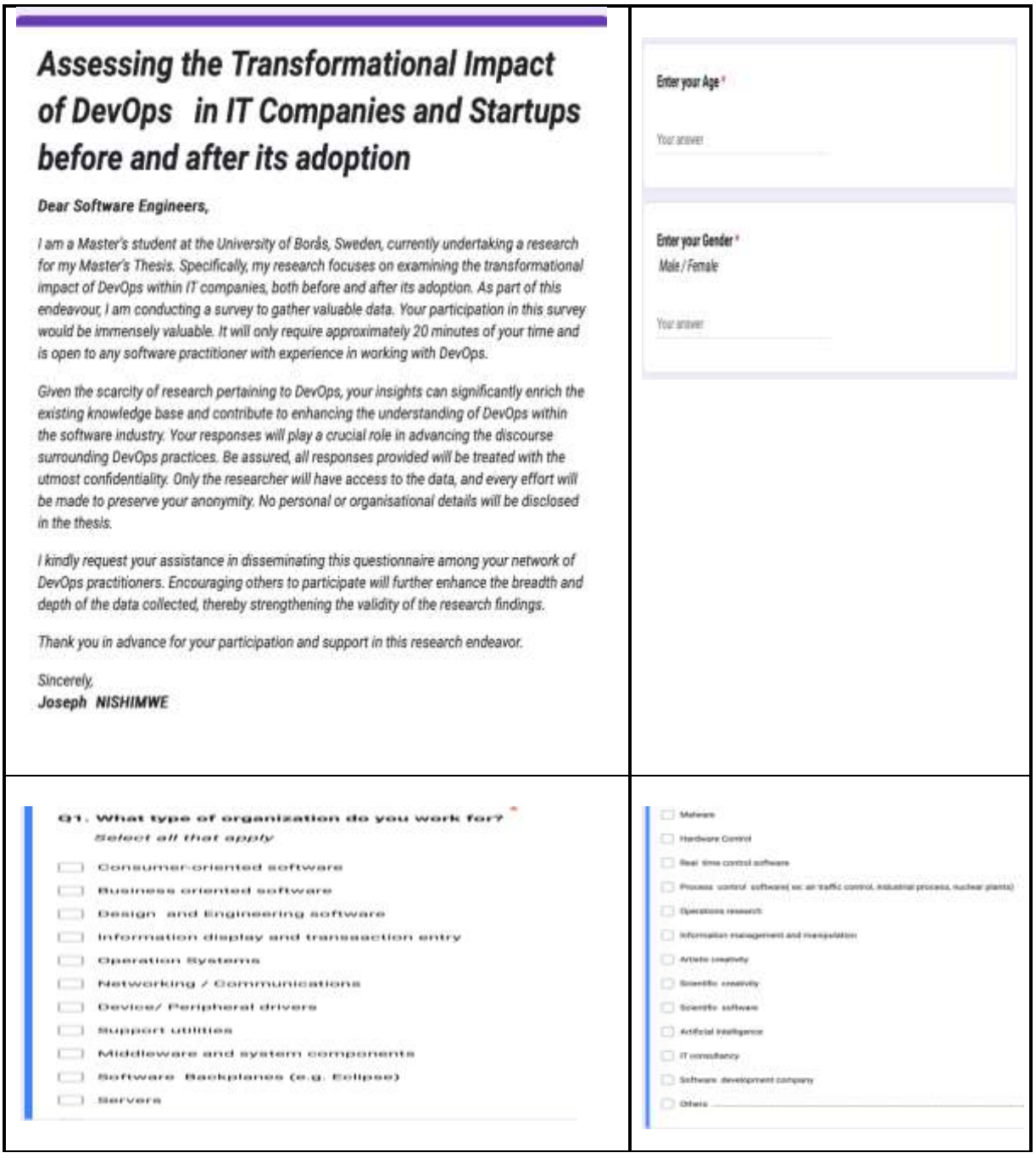
[64]. Markus, M.L., & Keil, M. (1994). If we build it, they will come: Designing information system that people want to use. MIT Sloan Management Review.

[65]. Orlikowski, W.J. (1992). The duality of technology: Rethinking the concept of technology in organizations. MIT Sloan School of Management.

[66]. Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping agility through digital options: Reconceptualizing the role of IT in contemporary firms. MIS Quarterly.

[67]. Fitzgerald, B., & Stol, K.-J.(2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software.

Appendix 1: The Survey questionnaire



Assessing the Transformational Impact of DevOps in IT Companies and Startups before and after its adoption

Dear Software Engineers,

I am a Master's student at the University of Borås, Sweden, currently undertaking a research for my Master's Thesis. Specifically, my research focuses on examining the transformational impact of DevOps within IT companies, both before and after its adoption. As part of this endeavour, I am conducting a survey to gather valuable data. Your participation in this survey would be immensely valuable. It will only require approximately 20 minutes of your time and is open to any software practitioner with experience in working with DevOps.

Given the scarcity of research pertaining to DevOps, your insights can significantly enrich the existing knowledge base and contribute to enhancing the understanding of DevOps within the software industry. Your responses will play a crucial role in advancing the discourse surrounding DevOps practices. Be assured, all responses provided will be treated with the utmost confidentiality. Only the researcher will have access to the data, and every effort will be made to preserve your anonymity. No personal or organisational details will be disclosed in the thesis.

I kindly request your assistance in disseminating this questionnaire among your network of DevOps practitioners. Encouraging others to participate will further enhance the breadth and depth of the data collected, thereby strengthening the validity of the research findings.

Thank you in advance for your participation and support in this research endeavor.

Sincerely,
Joseph NISHIMWE

Enter your Age *

Your answer

Enter your Gender *

Male / Female

Your answer

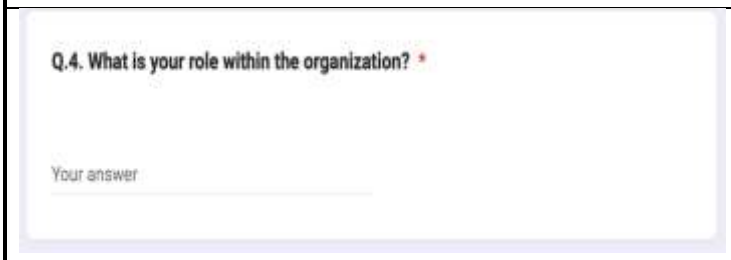
Q1. What type of organization do you work for? *

Select all that apply

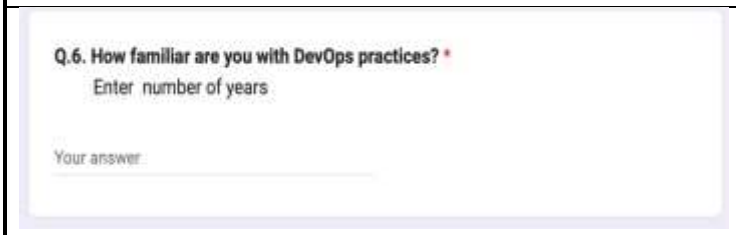
- Consumer-oriented software
- Business oriented software
- Design and Engineering software
- Information display and transaction entry
- Operation Systems
- Networking / Communications
- Device/ Peripheral drivers
- Support utilities
- Middleware and system components
- Software Backplanes (e.g. Eclipse)
- Servers
- Malware
- Hardware Control
- Real time control software
- Process control software(ex: air traffic control, industrial process, nuclear plants)
- Operations research
- Information management and manipulation
- Artistic creativity
- Scientific creativity
- Scientific software
- Artificial intelligence
- IT consultancy
- Software development company
- Others


Q.2.Where is your company located in Sweden? *
North
South
West
East


Q.3. How large is your company/organization? *
Tick only one box
< 10
10-50
50-500
500-5000
> 5000


Q.4. What is your role within the organization? *
Your answer


Q.5. What is your level of work experience? *
Please Enter the number of Years
Your answer


Q.6. How familiar are you with DevOps practices? *
Enter number of years
Your answer


Q.7. What development methodology did your company use before adopting DevOps? *
Check only one box
Agile methodologies
Scrum
Hybrid Methodology
Waterfall Model
Others :

Q.8. What are the expected benefits of adopting DevOps in your company? *
Select all that apply

- [ ] Enhanced collaboration
- [ ] Improved software quality
- [ ] Enhanced reliability
- [ ] Accelerated release cycles
- [ ] Elevated code quality
- [ ] Expedited feedback loops
- [ ] Improved workflow efficiency
- [ ] Automation
- [ ] Enhanced system stability
- [ ] Customer-centric approach
- [ ] Rapid product delivery
- [ ] Enhanced customer satisfaction
- [ ] Minimal manual intervention
- [ ] Cost-effective solutions
- [ ] Business responsiveness
- [ ] Increased revenue and efficiency
- [ ] Enhanced tracking and failure detection:

Q.9. Can you identify any additional benefits of adopting DevOps? *

Your answer

Q.10. How do you rate the challenges organizations face when initially adopting DevOps? *

1	2	3	4	5
○	○	○	○	○

The survey form can be found digitally here [50]

Q.11.What aspects do you find most challenging during DevOps adoption? *

Select one in each row

	Very Challenging	Challenging	Somewhat Challenging	Not Challenging	Can't specify
1. Availability of reliable resources for environment configuration	○	○	○	○	○
2. Adequacy of the current management structure	○	○	○	○	○
3. Effective management of geographical distribution of Dev and Ops teams	○	○	○	○	○
4. Integration of new tools with existing company tools	○	○	○	○	○
5. Dealing with legacy softwareDealing with legacy software	○	○	○	○	○
6. Employee resistance towards DevOps implementation	○	○	○	○	○

The survey form can be found digitally here [50]

7. Alignment of organizational structure with DevOps methodologies	○	○	○	○	○
8. Understanding hidden complexities and costs associated with DevOps	○	○	○	○	○
9.Management of complexity due to multiple systems	○	○	○	○	○
10. Designing software architectures to align with DevOps principles	○	○	○	○	○
11. Managing technical debt within the DevOps framework	○	○	○	○	○
12. Perception of DevOps as a buzzword	○	○	○	○	○
13.Clarity of definitions and goals of DevOps adoption	○	○	○	○	○
14. Unclear definition and goals of DevOps	○	○	○	○	○

The survey form can be found digitally here [50]

15. Security clearances when handling sensitive data in DevOps processes	○	○	○	○	○
16. Support from customers towards DevOps practices	○	○	○	○	○
17. Fear of change among employees during DevOps implementation	○	○	○	○	○
18. Managerial fear of failure in DevOps adoption	○	○	○	○	○
19. Making everyone accustomed to a new workflow	○	○	○	○	○
20. Understanding of applications and environments, especially in outsourced projects	○	○	○	○	○
21.Familiarization with new workflows introduced by DevOps	○	○	○	○	○

The survey form can be found digitally here [50]

22. Utilization of custom scripts for task automation in DevOps	○	○	○	○	○
23. Learning curve associated with DevOps adoption	○	○	○	○	○
24. Availability of resources for configuring environments in DevOps	○	○	○	○	○
25. Employee fear of measurement in DevOps performance evaluation	○	○	○	○	○
26. Visibility of production environments, particularly in embedded systems	○	○	○	○	○
27. Availability of proper automation tools in the domain of embedded systems	○	○	○	○	○

The survey form can be found digitally here [50]

Q.12.Are there any other challenges not mentioned previously that you think organizations face during DevOps adoption? *

Your answer

Q.13.What strategies can be employed to mitigate the challenges of DevOps adoption?Select all relevant options *

- [ ] 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
- [ ] Using a flexible architecture
- [ ] Specify release criteria for every project and document the challenges that popup
- [ ] 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

The survey form can be found digitally here [50]

Q.14. Can you suggest any additional strategies to address DevOps adoption challenges? *

Your answer

Q.15.To what extent do you agree with the characterization of DevOps as a framework of principles and practices rather than a specific process? *

	Strongly Disagree	Disagree	Neither agree nor Disagree	Agree	Strongly Agree
Mark only one option.	○	○	○	○	○

Q.16. Which DevOps practices do you believe organizations should incorporate into their development processes? *

Choose all that apply.

- ☐ Git and Gihub Collaboration
- ☐ Continuous Integration
- ☐ Continuous Delivery
- ☐ Azure Pipelines
- ☐ Continious Testing
- ☐ Containerization

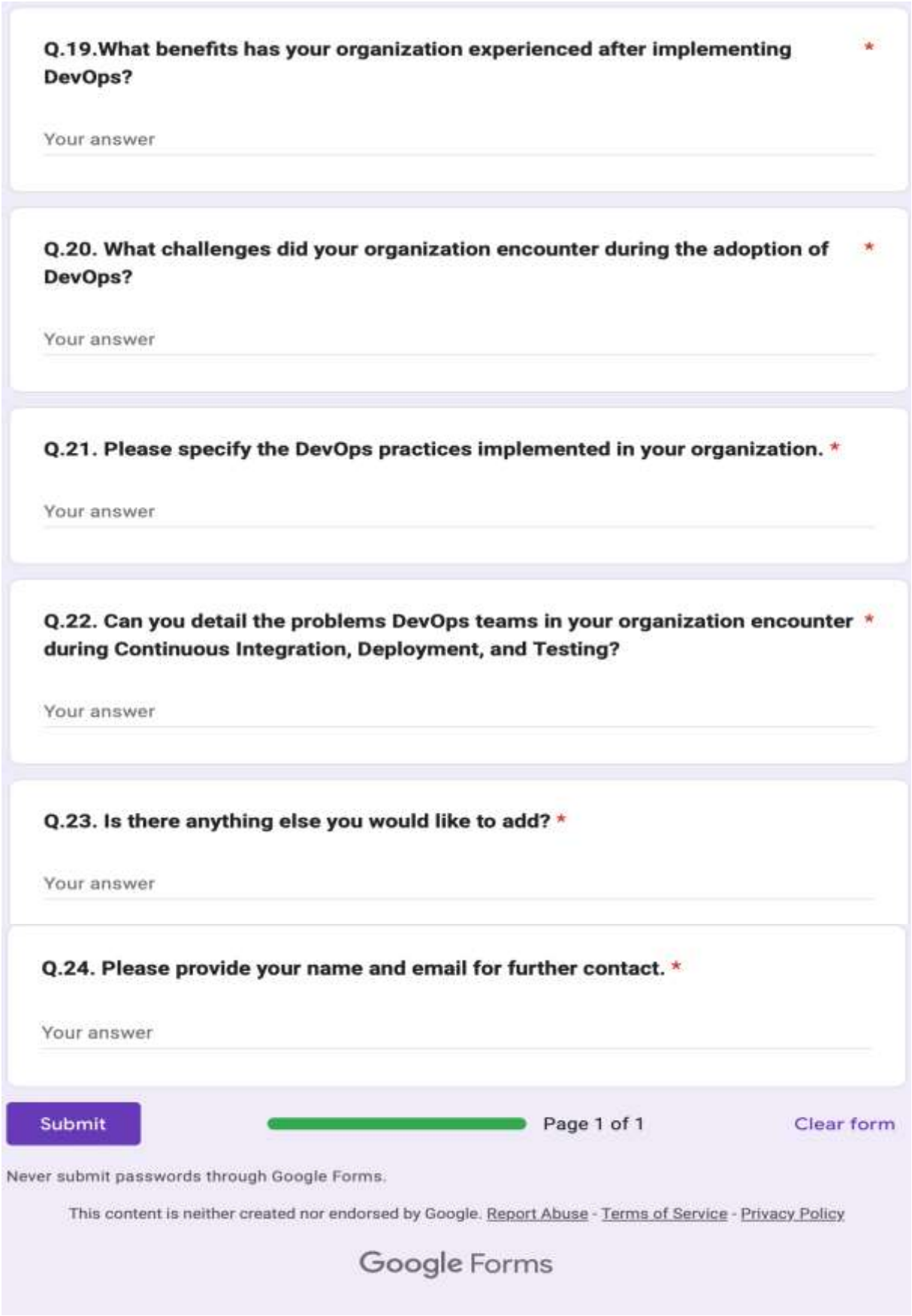
Q.17. Are there any other DevOps practices you would recommend that were not mentioned previously? *

Your answer

Q.18. What other problems do DevOps teams face that you consider significant? *

Your answer

The survey form can be found digitally here [50]



The survey form can be found digitally here [50]

Appendix 2: SLR Studies

Studies	Year	Author	Title	Publicatons Type	Reference
Std1	**2018**	Arvedahl, M,& Åkersten	Exploring the Criticality and Impact of DevOps Practices	Thesis	[59],[11]
Std2	**2011**	Patrick Debois	A Software Revolution in the Making?	Journal	[2]
Std3	**2012**	M. Hüttermann	DevOps for Developers	Conference	[16]
Std4	**2016**	P. kuvaja. L.E. Lwakare	Relationship of DevOps to agile lean and continuous deployment.	Conference	[22]
Std5	**2017**	S. Sharma	A Guide to Adopting DevOps in a Multi-Speed IT Enterprise.	Book	[26]
Std6	2020	Škurla, A.	DevOps integration of Security Practices	Thesis	[12]
Std7	**2015**	M. Virmani	Understanding devops bridging the gap from continuous integration to continuous delivery.	Conference	[5]
Std8	**2016**	Amaradi, A. S., & Nutalapati, S. B.	Continuous Integration, Deployment and Testing in DevOps Environment	Thesis	[14]
Std9	**2015**	Lianping C	Continuous delivery: Huge benefits, but challenges too.	Journal	[38]
Std10	2016	Matt. C and Alexandra	DevOps: Making it easy to do the right thing.	Journal	[62] **[57]**
Std11	**2016**	S. Jones, J. Noppen, and F. Lettice	Management challenges for DevOps adoption within uk smes	Conference	[61]
Std12	**2014**	Brian. F and K. Stol	Continuous software engineering: A roadmap and agenda.	Journal	**[63]**
Std13	**2021**	Roshan. R., Mansooreh. Z.	Challenges and Solutions when adopting DevOps.	Journal	[42]
Std14	**2016**	Kim, G	How to create World-Class Agility, Reliability, & Security in Technology Organizations.	Book	[29]
Std15	**2016**	Kristian. N, J. Smeds	On the Impact of Mixing Responsibilities Between Devs and Ops.	Conference	[59]
Std16	**2008**	Bjørnar. T and Jon. I.	Cooperation between developers and operations in software engineering projects.	Conference	**[60]**
Std17	**2014**	Floris. E, C. Amrit, and M. Daneva	Cooperation between information system development and operations.	Conference	[58]
Std18	**2017**	E. Laukkanen, J. Itkonen	Causes and solutions when adopting continuous delivery.	Journal	[37]
Std19	2021	Sohrab, C., & Bao	Automated Software security activities in a continuous delivery.	Thesis	[15]
Std20	**2016**	Ramtin Jabbari	What is DevOps.	Journal	[1]
Std21	**2015**	K. Nybom, I. Porres och J. Smeds	DevOps: a definition and perceived adoption impendiments.	Conference	[25]

Appendix 3: Interview Questionnaire

EMPLOYEES INTERVIEW QUESTIONS
INTERVIEWEE NAME:
DATE:
POSITION:
BACKGROUND AND EXPERIENCE QUESTIONS
▪ Can you describe the key benefits your company has experienced since adopting DevOps practices?
▪ Can you provide your age, occupation, and describe the initial challenges your company faced before adopting DevOps practices?
▪ What were the primary motivations for your company to transition to DevOps?
▪ How did the implementation of DevOps practices change your development and operations workflows?
CAREER GOALS AND MOTIVATION QUESTIONS
▪ What specific metrics or KPIs did you use to measure the impact of DevOps adoption on your company's performance?
▪ Can you provide examples of improvements in software delivery speed and quality since adopting DevOps?
▪ How has DevOps affected team collaboration and communication within your organization?
STRENGTHS AND WEAKNESSES QUESTIONS
▪ What tools and technologies were critical in your DevOps transformation, and why?
▪ Have there been any unexpected challenges or downsides since implementing DevOps practices? If so, what were they?
▪ In what ways has customer satisfaction been impacted by the adoption of DevOps in your company?
▪ What are recommended solutions to overcome these challenges?
WORK STYLE AND APPROACH QUESTIONS
▪ What advice would you give to other IT companies or startups considering adopting DevOps practices based on your experience?

University of Borås is a modern university in the city center. We give courses in business administration and informatics, library and information science, fashion and textiles, behavioral sciences and teacher education, engineering and health sciences.

In the **School of Business and IT (HIT),** we have focused on the students' future needs. Therefore, we have created programs in which employability is a key word. Subject integration and contextualization are other important concepts. The department has a closeness, both between students and teachers as well as between industry and education.

Our **courses in business administration** give students the opportunity to learn more about different businesses and governments and how governance and organization of these activities take place. They may also learn about society development and organizations' adaptation to the outside world. They have the opportunity to improve their ability to analyze, develop and control activities, whether they want to engage in auditing, management or marketing.

Among our **IT courses**, there's always something for those who want to design the future of IT-based communications, analyze the needs and demands on organizations' information to design their content structures, integrating IT and business development, developing their ability to analyze and design business processes or focus on programming and development of good use of IT in enterprises and organizations.

The **research** in the school is well recognized and oriented towards professionalism as well as design and development. The overall research profile is Business-IT-Services which combine knowledge and skills in informatics as well as in business administration. The research is profession-oriented, which is reflected in the research, in many cases conducted on action research-based grounds, with businesses and government organizations at local, national and international arenas. The research design and professional orientation is manifested also in InnovationLab, which is the department's and university's unit for research-supporting system development.

HÖGSKOLAN I BORÅS
VETENSKAP FÖR PROFESSION

VISITING ADDRESS: JÄRNVÄGSGATAN 5 · POSTAL ADDRESS: ALLÉGATAN 1, SE-501 90 BORÅS
PHONE: + 46 33 435 40 00 · E-MAIL: INST.HIT@HB.SE · WEB: WWW.HB.SE/HIT