ADDIS ABABA UNIVERSITY

School of Information Science

Department of Information Systems

DevOps Implementation Practices and Challenges in the Ethiopian Software Development Industries

By

Mesfin Tafese Gashaw

June 2022

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

School of Information Science

Department of Information Systems

DevOps Implementation Practices and Challenges in the Ethiopian Software Development Industries

A Thesis Submitted to Addis Ababa University's School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Systems

By

Mesfin Tafese

June 2022

Addis Ababa, Ethiopia

DevOps Implementation Practices and Challenges in the Ethiopian Software Development Industries

BY

Mesfin Tafese

Name and signature of Members of the Examining Board

Name	Title	Signature	Date
TIBEBE BESHAH (PHD)	Chairperson		
TIBEBE BESHAH (PHD)	Advisor		
TEMTIM ASSEFA (PHD)	Examiner		10/1/2022
LEMMA LESSA (PHD)	Examiner		Sept 09/2022

DEDICATION

This work is dedicated to all of my beloved families who have inspired me throughout my life! And especially to my dear brother Tasew (Ph.D.) and my lovely two sisters Abenezer and Tsion, who supported me throughout this academic journey!

And above all, to the Almighty God!

ACKNOWLEDGEMENTS

First and foremost, I thank the Almighty God for providing me with the wisdom and strength to begin and complete this study.

I'd like to express my heartfelt gratitude to my advisor, Dr. Tibebe Beshah, for his helpful advice and support. He has shown me the correct research path and encouraged me to continue with the study. Throughout the journey of this thesis work, he was always available to answer my questions, provide feedback, advice, and quick responses to my emails. I admire his dedication to his academic and professional responsibilities. Dr. Tibebe, thank you so much!

I would also like to express my heartfelt gratitude to all of my friends who have assisted me in all of my duties and who have participated in the survey and interview by providing me with strong commitment and advice on how I should proceed, as well as for my valuable time during the survey and interview.

DECLARATION

I declare that this thesis is my original work and that it has not been submitted for a degree at any other public or private university or college.

_____

Date

This thesis has been submitted for examination with my approval as university advisor.

_____

Advisor

ABSTRACT

The objective of this study is to investigate the practices and challenges of DevOps implementation in Ethiopian software development companies. This study used a mixed research approach with an explanatory sequential design method to collect and analyze data quantitatively and qualitatively. Based on a thorough literature review and empirical data findings, a research guideline was created. The survey questionnaires and interview outlines were designed with the established research objectives in mind, with some adaptations from existing literature and new ones developed as needed. The quantitative component of the study included 107 respondents from ten Ethiopian software development companies that implement the DevOps process either partially or fully. Interviews were also conducted to gather information from senior staff, managers, directors, and DevOps implementation teams. The quantitative data sets were analyzed using descriptive methods. OpenCoding approaches were used to examine the qualitative data. The findings revealed challenges in transitioning from traditional development to a DevOps approach; building DevOps infrastructure and applications with complex processes; and configuring DevOps tools and technology. Furthermore, the researchers discovered that the companies in the study faced a variety of challenges when implementing DevOps in culture-related practices. Employee resistance to new culture and technology; a culture of team collaboration and communication; a misunderstanding of DevOps environments and culture; and a lack of organizational commitment to DevOps culture changes were the major challenges. Expert validation and descriptive approaches were used to assess the study's outcome (proposed guideline). Expert validation was obtained as a result of company mail and survey questionnaires. As a result, the research process and results of this study are thought to be valid, indicating the usability and applicability of the study's output. This study is expected to produce insights that will assist Ethiopian Software Development Companies' DevOps teams in better understanding and implementing DevOps. Finally, in order to sustain the DevOps implementation process, it is suggested that the proposed guidelines be implemented by prioritizing the team work theme as a priority goal.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION AND ACRONYMS

CALMS	Culture Automation Lean Measurement Sharing
Dev	Development
IT	Information Technology
Ops	Operations
QA	Quality Assurance
SDLC	Software Development Life Cycle
SLAs	Service-Level Agreements

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Human social and economic activities are heavily reliant on software applications. Furthermore, for the majority of their activities, organizations prefer to use software applications. As a result of its importance in having software-intensive products and services, it is always useful, secure, and reliable during operational use (Lwakatare, 2017). Traditionally, the software advance and function teams have operated with a separation of responsibilities (Bass, et al., 2015; and Hemon et al., 2019). As a result, people's and organizations' proclivity to use new software applications increased competition in the software industry market.

The traditional "waterfall" methodology, with its sequential design-develop-test-deploy activity, is usually too slow and inefficient for digital business needs. Tensions frequently arise as developers strive to meet customer demands by releasing new software applications, while operations strive to keep software in production stable and reliable (Humble and Molesky, 2011; Wiedemann, et al., 2019). Businesses are increasingly requiring more frequent releases of new software in order to reduce risks during the delivery process of any production in order to satisfy clients (Leite, et al., 2019). In all of these understandings, software companies have provided new descriptive technologies, methodologies, and systems to their users, resulting in a multibillion-dollar business. It also creates software architecture, powerful software development tools, robust programming languages, and frameworks to improve the processes, functions, and security of software applications. As a result, software companies make an effort to integrate the best technologies and apply them in order to satisfy customers. Organizations and individuals must use innovative software applications and access their services to be competitive and successful in the business world. Many software companies are currently developing new software applications and methods of implementation based on this basic need.

Many implementation approaches are used for developing software in today's software development industry. The Agile Methodology is one of the most popular approaches to

developing software applications today. Since the 1990s, the agile development model has addressed communication gaps between customers and developers. Agile is defined as an approach in which improvement requirements and solutions are delivered collaboratively by self-organizing and cross-functional teams and their customers. Unlike traditional rigid software development methods, Agile encourages and promotes adaptability by promoting adaptive planning, evolutionary development, early delivery, and continuous improvement. According to Hüttermann (2012), while agile methodology successfully addressed common obstacles between programmers, testers, quality assurance, and customers, conflicts between the Development and IT Operation Teams persisted.

DevOps emerged in the last decade to address gaps between software developers (Dev) and IT operations/infrastructure (Ops). DevOps is an end-to-end model that focuses on increasing efficiencies and automating processes throughout the software lifecycle, from planning and coding to testing and refinement to implementation and enhancement. As a result, the new DevOps technology is the best solution for combining those two teams while also extending agility practices within the organization.

DevOps is, thus, an evolution of the agile methodology movement that encourages closer collaboration between developers and IT operations. DevOps, according to the researchers, is a collaborative and multidisciplinary effort within an organization to automate the continuous delivery of new software versions while ensuring their correctness and reliability (Leonardo et al., 2019). In this context, DevOps is defined as a software engineering process, practice, tools, culture, and philosophy that employ cross-functional teams to build, test, and release software more quickly and reliably through automation.

DevOps, as defined by Macarthy and Bass (2020), aims to improve collaboration between development and operations teams in software development. To clarify, DevOps is a combination of development and operations. It improves application development and management cycles, brings agility, and increases the availability and security of IT infrastructure by using standardized development methodology, clear communication, and documented processes supported by a standards-based, proven middleware platform. As a result, DevOps is concerned with connecting people, products, and processes, as well as, ultimately, IT and

business. Thus, it is a new paradigm for software development and operations (Jabbari et al., 2016; Fitzgerald and Stol, 2017). Accordingly, the benefits of DevOps include increased speed, new paths to greater quality and reliability, increased security, and support for scaling as needed.

Developing countries have adopted DevOps to improve their services because it shortens the systems development life cycle and provides continuous delivery of high-quality software products. Thus, DevOps is being implemented or adopted by the majority of software companies around the world. Accordingly, some Ethiopian software companies are implementing DevOps in order to align with the software development industry. Thus, Ethiopian software companies are appropriate cases for this study to identify DevOps implementation practices and challenges in the context of the Ethiopian software development industry.

1.2. Statement of the Problem

The global demand for IT users has increased, making it more difficult for IT companies to develop new technologies to meet customer demands. As a result, the IT industry is poised to change trends with rapidly changing consumer behavior in order to mitigate the significant challenges of customer demands in the business world. With new inventions, the software industry is attempting to improve their technological advancement process by utilizing various philosophies, methodologies, tools, processes, practices, and implementation cultures. This technological advancement aids in the production of high-quality goods and is viewed as a component of competition among them.

Recently, rapid changes in software development processes have occurred, and consumer satisfaction demands have become significant challenges. To improve the ever-changing nature of software, multiple Software Development Life Cycle (SDLC) models, development process operations, delivery, and maintenance of software, such as the Waterfall methodology, Iterative Model, Spiral Model, Agile, and derived forms, have evolved. Because of the sophisticated and complex interests of end users, software companies devote their time to developing new advanced technologies. The industry is currently shifting toward a new software development concept known as "DevOps" (Perera et al., 2016).

According to Yarlagadda (2021), DevOps is a new concept that brings together development and IT operations to improve overall organizational capacity and performances in order to meet the needs of customers and markets. Several Ethiopian software development companies are implementing the DevOps methodology to improve their software development and product delivery processes. However, the incorporation of DevOps into software development has not been without challenges. Each method of adaptation has its own set of constraints in terms of the working cultures of various environments and customization process. DevOps problems arise in part because it requires cross-functional competencies, hard and soft skills in IT team, as well as trends in the working culture of the given people.

Organizations frequently face challenges when integrating DevOps methodologies into their operations. According to Rakjumar et al. (2016), due to the complexity of prior deployment process and a lack of technical deployment ability among their software developers, organizations frequently face difficulties integrating DevOps methodology into their existing projects. In this case, DevOps implementation in practice has been linked to structural and task-related changes, increased responsibilities, full-fledged management, and team integration and collaboration. As a result, the misconception of DevOps implementation has been linked to job security threats and complex performance metrics, which can lead to resistance from IT development and operations professionals.

According to Violino (2016), efforts to implement DevOps in some environments around the world have failed, or have been difficult to sustain and integrate into the software development culture. DevOps has also been implemented in various ways and to varying degrees across organizations (Wiedemann, 2017). However, there is a lack of consensus on the appropriateness of DevOps practices, as well as evidences on the benefits and challenges of developing DevOps expertise and culture (Lwakatare, 2017). While there is a growing body of work attempting to define and clarify DevOps concepts, empirical research into DevOps experiences in industry is limited.

There is currently little research that deals with the practices and challenges of DevOps implementation in the Ethiopian software development context. So, based on these issues and the effects they may have on software delivery time, collaboration, and product quality, this research

attempts to fill gaps in the research by assessing the challenges and practices of DevOps implementation. The researcher is, thus, motivated to contribute to a better understanding of DevOps concepts in the Ethiopian software industries. Accordingly, the research questions are forwarded based on the problems stated above.

1. How the DevOps implementation process is practiced in Ethiopian software development companies?
2. What are the challenges of the DevOps implementation process in Ethiopian software development companies?
3. What guidelines are needed to improve DevOps implementation in the Ethiopian software companies?

1.3. The Study's Objective

1.3.1. General Objective

The main objective of this study is to assess the practices and challenges of the DevOps implementation process in the Ethiopian software development industry and propose a contextually appropriate guideline for successful implementation.

1.3.2. Specific objectives

- To identify the practices of DevOps implementation in Ethiopian software companies.
- To identify the challenges of DevOps implementation in Ethiopian software companies.
- To propose applicable guidelines for the successful implementation of DevOps in the Ethiopian software development industry.

1.4. Study Scope and Limitations

The scope of this study is limited to identifying the implementation of DevOps practices and challenges, mainly focusing on identifying current practices in DevOps implementation and the challenges in the Ethiopian software development context. This research focuses on Ethiopian

software companies, either private or public. The limitations of this study do not address all Ethiopian software companies, so this research only focuses on some selected software companies, such as those that have partially or fully implemented DevOps.

1.5. Significances of the Study

This study is expected to benefit Ethiopian software companies and the software industry by understanding current practices and challenges of DevOps implementation in Ethiopian software companies. In addition, software companies' development teams, quality assurance teams, and IT operation teams to get more knowledge and understanding of DevOps implementation in the Ethiopian software development context.

The research is expected to produce results that can help organizations to create stronger bonds between the development team, IT operations team, and other stakeholders in the company. This research is also endeavor to propose guidelines for the successful implementation of DevOps in the Ethiopian software industry; and thus software companies and business stakeholders are able to develop their own specific strategies.

1.6. Research Organization

The first chapter is an introductory section in which the researcher describes in detail the study's background, problem statement, and objectives, and then presents research questions, significance, scope, and limitations. Chapter two mainly discusses the concept of DevOps, implementation practices, and empirical and theoretical literature. The third chapter discussed the research methodology. It uses questionnaires and interviews to answer the research questions. Data presentation, analysis, discussions, and guidelines are presented in the fourth chapter. Finally, Chapter five summarizes the findings of the study with a conclusion and recommendation. It also includes limitations and recommendations of the study for future research.

CHAPTER TWO

2. REVIEW LITERATURE

2.1. History of DevOps

The emergence of DevOps starts with Agile. Agile methodology is a collaborative, self-driven, customer-oriented, iterative, and evolutionary approach. Agile is popular in the software industry because of its benefits like faster delivery and improved quality features. Agile is better than existing traditional approaches like a waterfall and incremental approach in terms of customer satisfaction (Batra, 2020). At the agile conference in Toronto, Debois (2008) helps to plant the seeds of the DevOps movement, resolving the conflict between the software developers and the operations teams when it comes to getting great work done quickly. Similarly, in 2009, Allspawin and Hammond discussed 10+ Deployments per Day: Dev and Ops Cooperation at Flickr at the O'Reilly Velocity conference. In 2010, the first US Developer Days were organized with the help of Willis. The events quickly established themselves as a regular global series of community-organized conferences, as well as a major driving force in the DevOps community. In 2011, the DevOps community started to build open-source tools like Vagrant (for creating and configuring virtual development environments) that work with existing management tools like Puppet and Chef.

According to Hamunen (2016), DevOps has evolved from a collection of ideas and buzz into a more coherent and standardized process. While the name became popular and got a lot of attention, the initial traction was limited to startups, more specifically, organizations that provide web applications. These apps were created by developers (the Devs), who typically delivered changes and updates to their web apps in a very short period of time. The main common barrier they encountered was that of operations (the Ops), which was slow in implementing those changes due to rigid and rigorous change management processes (Sharma, 2017).

According to Sharma (2017), the DevOps movement aims to address this impedance mismatch between the Dev and Ops teams, bridge the gap, and foster more communication, collaboration, and trust. At its core, it was a cultural movement aimed at bridging the gap between Dev and Ops

as well as using automation to make application delivery faster, more efficient, and eventually continuous. In 2010, Humble, then at Thought Works, took DevOps to practitioners throughout the industry with his book Continuous Delivery, codifying some of the practices that make up the core of DevOps and making DevOps adoption tangible and available to all (Sharma, 2017).

2.2. The Concept of DevOps

DevOps is an emerging concept, but the efforts to harmonize various aspects of the entire development-to-operations process mark the start of a new era. An IT service delivery approach rooted in agile philosophy with an emphasis on business outcomes, not business orthodoxy. In simple terms, DevOps refers to an umbrella concept that encompasses people, processes, technologies, culture, and artifacts required to connect development to execution. People are the critical element in the equation in both processes. To effectively use the tools across the process or lifecycle, people should be encouraged to abide by some guiding principles to drive a common mindset (Menzel, 2015; and Macarthy and Bass, 2020).

DevOps is a concept linked to organizational success when implementing information technology solutions. According to Hart and Burke (2020), DevOps is a framework for improving operational efficiency and insight into organizational needs by bridging the gap between development processes and organizational operations, promoting faster deployment and automated IT implementation. According to Menzel (2015), higher automation, improved visibility, and tighter control of the pre-production and production environments, as well as deployment or promotion of code across varied environments, are all part of DevOps' approach to addressing the aforementioned difficulties. It also aims to break down the "wall of confusion" that exists between development and operations workers by synchronizing development and operations tools (enabling input from operations to development) and realigning objectives and incentives (Menzel 2015).

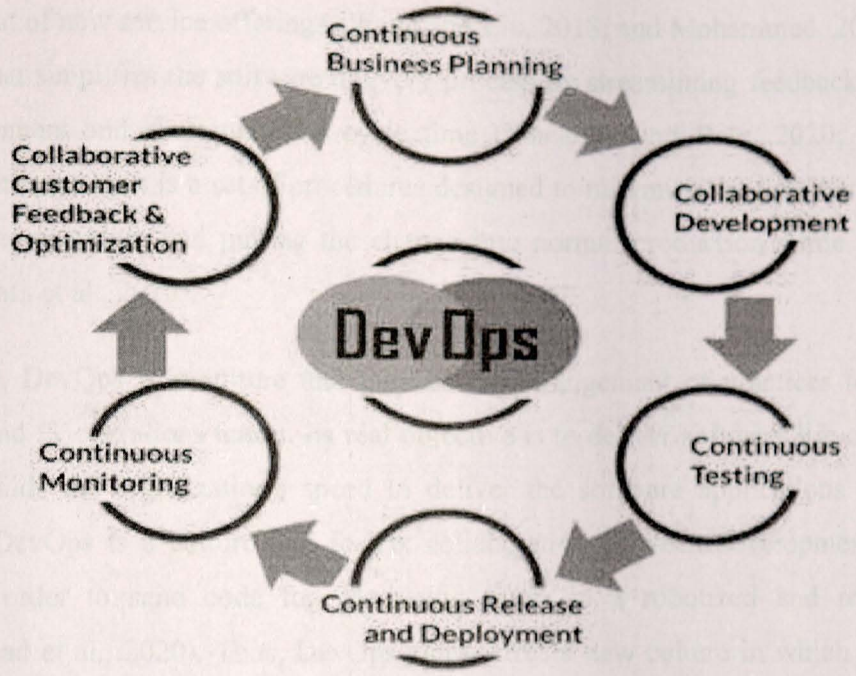

Figure 1: DevOps life cycle processes (Yarlagadda 2021).

2.2.1. Definitions of "DevOps"

Before looking deeper into the core capabilities and practices that must be adopted, as well as the various plays that must be executed, to implement DevOps in an organization, it is better to understand the definition of the term DevOps. Software companies have become increasingly passionate about and involved in the DevOps movement over the last several years. It is so exciting to see all the impact of DevOps that is coming from the innovation happening in the industry on individuals and organizations. However, DevOps is also such a broad, vague, and confusing term that everyone has their own take (Macarthy and Bass, 2020). DevOps is a combination of development and IT operations, and it is defined as a methodology, practice, tools, culture, and philosophy that employs cross-functional teams to develop, check, and discharge software faster and more accurately through automation.

DevOps is a development methodology that emphasizes cooperation and interaction, team collaboration, quality standards, and delivery with automated deployment, all while leveraging a

variety of development procedures and removing barriers to the instant and continuous deployment of new service offerings (Wang and Liu, 2018; and Mohammed, 2016). DevOps is a practice that simplifies the software delivery process by streamlining feedback from production to development and shortening the cycle time (Macarthy and Bass, 2020; and Hüttermann, 2012). Furthermore, it is a set of procedures designed to minimize the time between committing a change to a system and putting the change into normal production while maintaining high quality (Bass et al., 2015).

In general, DevOps is a culture that pursues the arrangement of practices to consolidate the DevOps and IT operations teams. Its real objective is to deliver software life cycles faster. This culture builds the organization's speed to deliver the software applications and their related services. DevOps is a culture that fosters collaboration between development and operations teams in order to send code for generation faster in a robotized and repeatable manner (Muhammad et al., 2020). Thus, DevOps emphasizes a new culture in which development and deployment tasks are shared between developer and operations teams and where all team members accept responsibility for the quality of the final deliverable (Rowse & Cohen 2021). According to Muhammad et al. (2020), DevOps is fundamentally a combination of two words: "Dev" and "Ops." Dev refers to development (representing software developers, including programmers, testers, and quality assurance personnel) and Ops refers to operations (representing the experts who put software into production and manage the production infrastructure, including system administrators, database administrators, and network technicians).

Software Development

The software development process models used in the IT industry adhere to a common framework that is used to efficiently and productively structure, fully intend, and regulate the process of developing an information system. There are several software development life cycle models defined and designed to be used during the software development process. Each process model follows a series of steps unique to its type, ensuring success in the process of software development. Traditional software development life cycle models are the Waterfall Model, Iterative Model, Spiral Model, V-Model, and Big-Bang Model. In 2001, Agile was introduced to

the industry, and companies used to practice agile concepts such as Scrum and Kanban. A few years ago, the industry introduced DevOps, which was an improved version of agile focusing on operational aspects (Perera et al. 2017). Further, Dev is a process for creating a software product consisting of inter-relented activities of software requirements analysis, architectural design, coding, testing, integration, and acceptance testing (Lwakatare, 2017).

IT Operations

The IT operation is responsible for the provisioning of hardware and software; personnel with specialized skills; specification and monitoring of SLAs; capacity planning; business continuity; and information security. Most of these responsibilities have aspects that are included both inside and outside of DevOps processes. Any discussion of which aspects of Ops are to be included in DevOps must take into consideration all of the activities that Ops currently performs and involves functional activities, personnel skills, and availability (Bass et al. 2015). In addition, Ops is the process of putting into use and supporting end-users in the use of a software product in an operational environment. Furthermore, its activities include installation, upgrade, migration, operational control, monitoring, configuration management, alerting, availability, and support (Lwakatare, 2017).

Development and Operations

The solution to this battle between Dev and Ops is what DevOps addresses: to achieve the balance between innovation and stability and between the speed of delivery and quality. To achieve this, both development and operations need to improve how they operate and align (Sharma 2017).

The Dev View: The Ops need to change more than Dev, but Dev also needs to make several changes:

- Dev needs to work with Ops to understand the nature of the production systems their applications will be running on. What are the standards for the production systems (environmental patterns) and how should their applications perform on them? Within

what constraints do the applications need to operate? Dev now needs to understand system and enterprise architectures.

- Development needs to get more involved in testing. This means not just making sure that their code is bug-free but also testing the application to see how it will perform in production. This requires Dev to work closely with Quality Assurance (QA) and to test their application in a production-like system. Dev also needs to learn how to monitor deployed applications and understand the metrics Ops cares about.

- They need to be able to decipher how processes interact and how one process can cause another one to slow down or even crash. They need to understand how changes to their code will impact the entire production system, not just their application.

- DEV needs to communicate and collaborate better with Ops.

The Ops View: Ops need to be able to provision new environments rapidly, and they need to design their systems to absorb rapid change.

- Ops need to know what code is coming and how it may impact their system. This necessitates their involvement in development; beginning with understanding the requirements and system specs of the applications being developed. This process is referred to in Lean and DevOps as "shift left." They need to make sure that their systems can accommodate these applications as they are enhanced.

- Ops need to automate how they manage their systems. Rapid change with stability cannot be achieved without automation. Automation will allow not only rapid change but also rapid rollbacks if something does break.

- Ideally, operations need to version their systems. This can only be done when the infrastructure and all changes to it are captured and managed as version-controlled code. Thus, they need to leverage infrastructure as code or, even better, software-defined environments.

- Ops need to monitor everything throughout the delivery pipeline, whichever environment the Ops teams manage. They need to be able to spot potential instability as soon as it happens.
- Ops need to communicate and collaborate better with Dev.

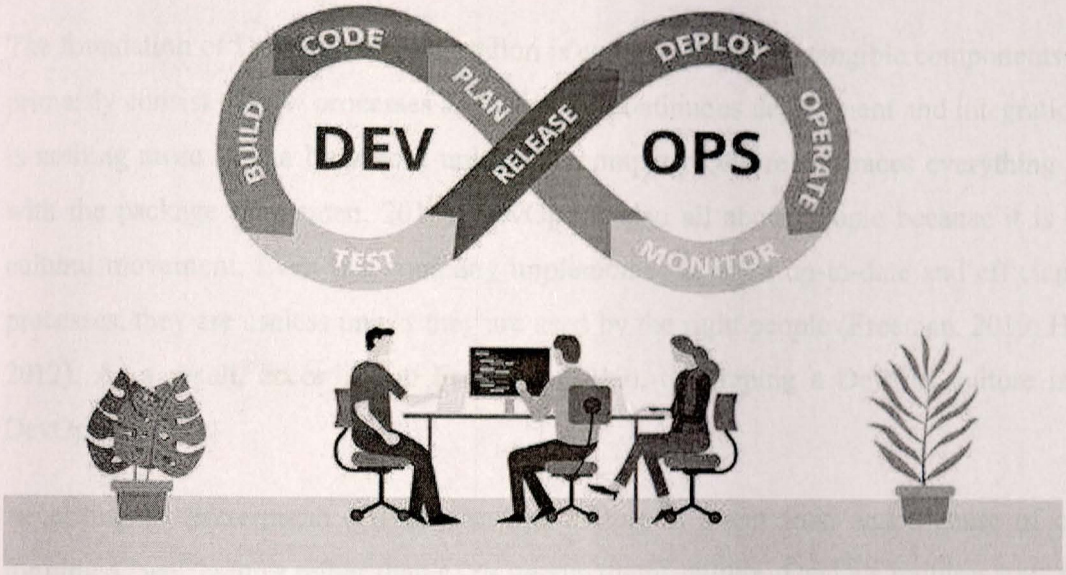

Figure 2: DevOps encourages collaboration between software development and operations (https://dhareshvadalia.medium.com/transitioning-from-traditional-sdlc-to-devops-model-92aaa23b0e97)

2.2.2. The key areas of DevOps

DevOps' key area includes fundamental characteristics that define the concept. The major features depict and characterize the various aspects of DevOps, as well as the steps taken to address problems caused by a lack of collaboration between software development and operations. Through the various elements, software practitioners can begin to debate the concept of DevOps, leading to a shared understanding of DevOps and identifying area where DevOps can be implemented (Lwakatare, 2017).

At the first DevOps Days conference in the United States in 2010, two DevOps pioneers, John Willis and Damon Edwards, coined the acronym "CAMS" to describe the DevOps Concept. "CAMS" is an acronym that stands for Culture, Automation, Measurement, and Sharing (Nybom

et al., 2016; Hamunen, 2016; and Macarthy and Bass, 2020). Humble and Farley (2010), the authors of the book on Continuous Delivery, later added the letter "L" for "Lean" to create "CALMS." Let's break down each of the "CALMS" elements so we can get a better sense of where an organization is on its DevOps journey.

Culture

The foundation of DevOps implementation is culture. While the tangible components of DevOps primarily consist of new processes and tools for continuous deployment and integration, DevOps is nothing more than a buzzword unless the company culture embraces everything that comes with the package (Hamunen, 2016). DevOps is also all about people because it is primarily a cultural movement. Even if a company implements the most up-to-date and efficient tools and processes, they are useless unless they are used by the right people (Freeman, 2019; Hüttermann, 2012). As a result, according to Freeman (2019), developing a DevOps culture is central to DevOps.

According to Hüttermann (2012), DevOps culture is about trust and a sense of community, forming a "we" culture rather than a "us versus them" culture. DevOps requires transforming the company culture from a collection of silos to an openly collaborative way of working. It necessitates involving operations personnel in the design and transition of the application. They should also attend necessary planning meetings, retrospectives, and project team displays in order to share their perspectives and knowledge early in the process (Hamunen, 2016). A DevOps culture is defined by Freeman (2019), Hüttermann (2012), and Kim et al. (2016) as a culture characterized by a high degree of collaboration spanning roles and teams, enabling continuous learning, high trust, accepting that failures occur, and making it acceptable to talk about problems to prevent and solve them.

Automation

DevOps relies heavily on automation. Automation is used to obtain immediate feedback. There are various approaches to optimizing delivery, but when aiming for a high degree of automation, pitfalls must also be considered (Hüttermann, 2012). Automation, according to Hüttermann (2012), is the use of solutions. Automation, according to Hüttermann (2012), is the use of

solutions to reduce the need for human labor. Automation can ensure that the software is built the same way every time, that the team sees every change made to the software, and that the software is tested and reviewed in the same way every day, ensuring that no defects slip through or are introduced due to human error.

Automated software releases can be deployed to any deployment environment, including testing, staging, and production, with the click of a button. Automation tools are required for the automated release and deployment of software. Tools for managing application life cycles, such as development, testing, release management, configuration management, monitoring, and control (Freeman, 2019). In addition to supporting self-service technologies and platforms, DevOps strives to establish autonomous teams that can apply their expertise in their daily work without relying on other teams. They also claim that automation enables businesses to build an effective software delivery pipeline in which small teams can rapidly develop, test, and deploy code and value to customers (Kim et al. 2016).

Lean

The goals of lean manufacturing are to ensure the rapid delivery of high-quality products while focusing on waste elimination and cost reduction. In several industries, lean manufacturing has resulted in significant cost and resource savings, much higher-quality products, and faster time to market. In the field of software development, this philosophy is becoming more mainstream. Lean is certainly not limited to small systems. It was developed and applied to massive organizations and even entire economies (Humble and Farley, 2010).

Lean and Agile practices, according to Freeman (2019), are at the heart of DevOps. Lean emphasizes incremental improvements and divides work into small batches that can then be delivered to end customers quickly and frequently on a daily basis. Frequent releases enable users to provide quick and useful feedback, such as feedback on the most recent software version, and this feedback enables learning, improvement, and adaptation in response to the feedback. To optimize software delivery processes, the Lean software development movement teaches the importance of taking a holistic approach that connects all parts of the delivery process and everyone involved (Humble & Farley, 2010; and Asfar et al, 2017).

Measurement

Measurements are used to monitor and track performance, and DevOps takes a unique approach to measurement. Application monitoring in production is another point of view on measurement. Measurement is emphasized in traditional projects as an important tool for tracking progress, identifying current status, and scheduling dates. Agile project settings try to find different approaches to creating measurements, but they frequently end up on dead ends when attempting to bridge operations to development. Measurement is important in both traditional and agile projects because you can only improve if you measure (Hüttermann, 2012).

Measurement in DevOps is also defined by Humble and Molesky (2011) as "monitoring high-level business metrics such as revenue or end-to-end transactions per unit time." At a more fundamental level, it necessitates the careful selection of key metrics in the delivery process, such as lead-time and the impact of releases on system stability. Measurement influences how people work, and visualization of these metrics is critical so that everyone can see how well the team is doing at any given time. Metrics also influence the discovery of process bottlenecks, which is an important step in keeping the process lean (Humble and Molesky, 2011).

Sharing

The final important aspect of the CALMS model is sharing. A transparent culture with effective channels for exchanging knowledge across teams is required to break down the barriers between Development and Operations. Sharing occurs on several levels. The development and operations teams can celebrate successful releases together, which is a simple but effective form of sharing. It also entails sharing knowledge, such as ensuring that the relevant operations team is aware of any new functionality that will be available to them as soon as possible, rather than on the day of the release.

DevOps also emphasizes the sharing of development tools and techniques for managing environments and infrastructure (Humble and Molesky, 2011). In addition, knowledge management is a process that promotes sharing. Three-tier knowledge management is a method of representing knowledge types related to exploration, evaluation, and execution. Exploration is concerned with observing technological trends and developing plans to innovate and improve.

The study of the feasibility and specification of software is referred to as evaluation. The construction, testing, and operation of a software product is referred to as execution (Floris et al, 2014).

2.3. The Implementation of DevOps

DevOps perpetration is still a challenge for numerous companies that struggle with further than just software development. DevOps' growing perpetration by utmost associations throughout the world demonstrates that it has the implicit to be a significant enabler for achieving product delivery scale. DevOps principles enable a business to give software that's briskly, better, advanced quality, and more dependable. A culture of cooperation and collaboration among all functions of the association is needed for successful DevOps perpetration. According to Lwakatare (2017), no generally accepted frame exists for DevOps perpetration or relinquishment. The perpetration of DevOps in software associations isn't homogeneous due to numerous differences, similar as the objects and the starting birth of the association, among others. DevOps perpetration entails a series of organizational and socio-specialized practices employed by software associations to grease collaboration between software development and operations (Lwakatare, 2017).

2.3.1. Implementation of DevOps: An Organizational Perspective

The majority of DevOps practices focus on organizational structures and concern how organizations deal with the connections, communication gaps, and liabilities between software development and operations, as well as the corresponding artistic changes (Nybom et al. 2016; Jones et al. 2016; and Chen, 2017). According to Cito et al. (2015), four practices on organizational structure can be observed from software interpreter responses in the software development process. These are: 1) high collaboration between software development and operations; 2) detachment of software development and operations with a mediator (an individual or platoon) between the two groups; 3) the software development platoon gaining additional responsibility for software operations; and 4) no noticeable software operations platoon.

The most current practice is stated to be first on the list in large associations, whereas the final two are more common in small and medium- sized associations (Cito et al. 2015). According to Lwakatare (2017), the configuration operation platoon at Facebook advocated the practice, which consisted of a small group of masterminds responsible for enforcing configuration tools as well as masterminds developing a Facebook web operation (Lwakatare, 2017). The configuration operation platoon masterminds are said to be in charge of the development and product deployment of new features, as well as covering and resolving product issues of configuration tools used by thousands of Facebook masterminds to manage configurations of hundreds of thousands of waiters and over one billion mobile devices (Lwakatare, 2017). Facebook neural pathways are said to be particularly interested in developing and testing legislation that must support functional use while also keeping the system running smoothly at all times. The formation of a separate platoon to act as a bridge between the two groups was an essential practice for software inventors assuming additional operational responsibility, particularly in large organizations (Lwakatare, 2017). According to Lwakatare (2017), large organizations form a dedicated platoon composed of a slight number of members with interdisciplinary contexts, such as software development and operations.

2.3.2. Implementation of DevOps: A socio-technical perspective

Practices concerning defined way, specialized structure, and mortal agents involved in software development and operations are central to the socio-technical aspects of DevOps perpetration (Cito et al. 2015). Cito et al. (2015) linked similar practices in a software interpreter check after pertaining areas of software development that are changing as a result of pall computing. The practices are classified as follows: 2) software deployment and provisioning; 3) software design and implementation; 4) software is monitoring and product data operation; and 5) tooling (Cito et al. 2015). According to Lwakatare (2017), the early and prominent DevOps adopters are software companies that do pall and web operations.

The specifics of the practices vary from one organization to the next. Nonetheless, a common practice observed is the use of a deployment channel (Lwakatare, 2017). The deployment channel describes the path that software changes take and the conditioning that is done on their way to product. The deployment channel includes many aspects of software development that

aren't always prioritized by the software development platoon. The deployment channel, similarly, is where cross-cutting enterprises and knowledge between software development and operations intersect (Bass et al. 2015). The deployment channel provides visibility into the software delivery process and allows for immediate feedback on the impact of software changes on product readiness.

Furthermore, the deployment channel provides control over the software delivery process, allowing the software development and operations teams to quickly release any chosen interpretation of software into the environments that the teams manage (Lwakatare, 2017). The high-level overview of workflow conditioning in the deployment channel begins with a law committed by an individual software inventor to an interpretation control system in a development terrain in which all platoon members participate. To complete the task of making a software change, point branches can be used. In the following stage, continuous integration monitors the law commits to validate law changes against automated tests, such as unit and integration tests, and generates a figure as an affair if all tests pass. Following that, the figure packages that passed the tests are stationed to the posterior surroundings and, eventually, to the product environment.

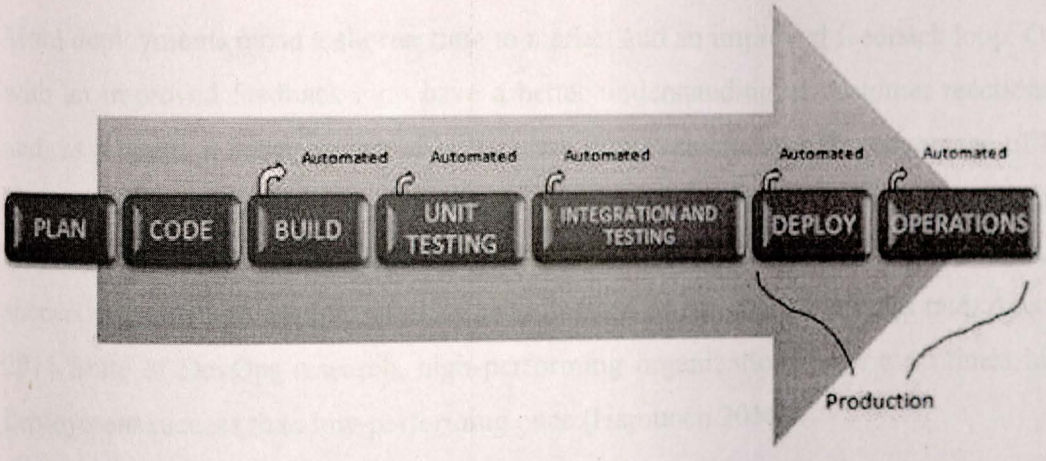

Figure 3: Deployment process (Batra and Jatain 2020).

2.4. Practices, Benefits, and Challenges of DevOps Implementation

DevOps arose as a result of issues with the traditional method of managing development and operations. To overcome this problem we implement DevOps in software organizations. During

the successful implementation of DevOps in software organizations there are numerous benefits and challenges. This section will give an overview of the main practices, benefits and challenges on the successful implementation of DevOps.

2.4.1. Practical Benefits of DevOps Implementation

Currently, empirical studies have found numerous advantages to implementing DevOps. Reduced lead-time, enhanced change success rate, and improved collaboration between developers and operations are just a few advantages. Reduced lead time and increased deployment frequency are the main advantages of DevOps. The time it takes for changes to go from code committed to code successfully running in production demonstrates how lead time is defined. Furthermore, deployment frequency refers to how frequently the company deploys code. Using Lean methodologies to reduce batch sizes and automating the delivery process are important contributors to these metrics. High-performing IT organizations outperform low-performing organizations by 200 times. Furthermore, by minimizing bottlenecks and easing deployment pains, high-performing organizations deploy 30 times more frequently than low-performing ones. High-performing organizations employ DevOps approaches in this context (Hamunen, 2016).

More deployments mean a shorter time to market and an improved feedback loop. Organizations with an improved feedback loop have a better understanding of customer reactions to features and, as a result, a better means of continuous improvement. Another advantage of DevOps is a higher rate of change success. Organizations have seen a reduction in deployment pains with a well-established continuous development pipeline in place. The percentage of changes that succeed when rolled out into production is defined as the change success rate. According to the 2015 State of DevOps research, high-performing organizations have a 60 times higher rate of deployment success than low-performing ones (Hamunen 2016).

The use of version control, infrastructure as code, automated testing, and common build mechanisms for different environments to prevent code errors and configuration differences is critical to realizing this benefit. Employee satisfaction and a lower level of burnout in staff are attributed to a high change success rate when unplanned, meaningless work and related stress decreases. DevOps also contributes to increased stability. In this context, stability refers to a

system's ability to process requests while remaining error-free. Enhanced stability is achieved by incorporating quality into the system through the use of automated tests, developing applications with testability and deployability in mind, and fostering a culture of continuous improvement (Hamunen, 2016).

It has also been demonstrated that implementing DevOps has cultural and organizational benefits. Developers and operations have been observed to work more closely together, and the necessary expertise has been embedded in the software development team, especially when dealing with operations-related tasks. The latter aided in improving team morale and easing communication between software development and operations, particularly because teams could release software more frequently and confidently. Despite their importance, improved cultural changes as a result of DevOps have been reported to be difficult to quantify (Lwakatare, 2017).

2.4.2. Implementation Challenges for DevOps

In the literature, major barriers to promoting DevOps include a lack of a clear definition, insufficient communication, ingrained company culture, association structure, and geographical dispersion. Some research looks at how DevOps is relinquished and perpetrated to describe the issues. The following issues have been identified based on previous publications and trials conducted over time.

i) DevOps is not Well-Understood

Despite setting a clear goal of understanding DevOps through cutting-edge research before embarking on a DevOps journey, one barrier to acceptance was that people understood DevOps in a variety of ways. As a result, some colorful stakeholders working in chaos had some unrealistic DevOps expectations. As a result, a diverse set of stakeholders had some unrealistic DevOps expectations, leaving them perplexed and confused (Elberzhager et al., 2017).

ii) Obstacles in the Workplace and in the Community

Organizational and cultural barriers are two of the most visible issues that DevOps seeks to address. The division of development and operations into different brigades or "basically approximately connected working groups" is a major impediment to achieving a common goal of

producing value-added software quickly. In addition to being in separate brigades, development and operations may not have a consistent language for expressing their ideas and issues. This causes misunderstandings, confusion, and unnecessary work on both sides, as well as on the business side. Fear is the final impediment. Fear is a negative attitude toward learning that is common and is associated with old habits and the fear of losing power, character, or influence (Hütterman, 2012).

iii) Liabilities between software development and operations are muddled

Some argue that close collaboration between software development and operations blurs the line between the two (Nybom et al. 2016). Because there is no participated law of conduct that includes a clear assignment of liabilities, similar to product and design directors, the new culture is poorly understood or appreciated by those affected, including stakeholders in software development with the operation portion (Elberzhager et al. 2017, Jones et al. 2016). Furthermore, the DevOps practice of combining development and operational liabilities introduces a new point of contention, especially in the absence of confidence (Nybom et al. 2016). The operations labor force is unable to completely delegate their responsibilities to software masterminds due to a lack of expertise in operations jobs among software innovators.

iv) The Operations Team's Defiance

The resistance of the operations team has also been identified as a barrier to DevOps abandonment. As forms of resistance, exposures generated by operations in using specific technologies presented by software development, as well as the notion that software inventors would have to accomplish everything on their own, were observed (Jones et al. 2016).

2.5. Related Works

DevOps is defined in several ways, with an emphasis on the culture of DevOps and its reliance on a set of software engineering processes. For example, Leite et al. (2019) led a literature assessment of DevOps renunciation provocations and found that while specific concerns like DevOps automation are more easily established in literature, there isn't yet agreement on how to

deploy DevOps culture effectively. This is due to the fact that studies aren't focused on specific examples of DevOps practice and culture.

Despite this, Gupta et al. (2017) conducted numerous large-scale researches in order to discover the attributes relevant for determining the maturity of a DevOps program. They identified source control and automation technologies as critical to achieving nonstop delivery principles in their research. They also identified "cohesive brigades" as a key set of DevOps characteristics. Luz et al. (2018) surveyed DevOps interpreters from 15 different organizations in Brazil, Ireland, Portugal, Spain, and the United States. Their research aided our understanding of successful DevOps decommissioning strategies. They listed automation, transparency and sharing, dexterity, adaptability, nonstop dimension, and quality assurance as DevOps enablers and issues.

Some research has also focused on the problems of DevOps and how to investigate them qualitatively. DevOps issues are identified by Lwakatare et al. (2016). Client dependency, software incompatibility, restricted sight of client surrounds, and a lack of technology to automatically embed new capabilities into client environments were identified as the top obstacles in their study. In three case studies in Finland, Riungu-Kalliosaari et al. (2016) studied the benefits and limitations of DevOps. They set up challenges largely girding the DevOps culture and implementation across various environments, with benefits including more enforced features and frequent releases, improved release quality from automation, improved collaboration and communication between functions, and the well-being of the DevOps implementation team.

Literature focuses on social and creative miracles in addition to the particular miracle of DevOps. Jones et al. (2016), for example, highlighted DevOps' operational structure issues, as well as the unclear obligations between development and operations units. They also found that innovators are usually tasked with maintaining outdated systems, which hinders their understanding of new DevOps technology. The scarcity of specialized skills, the complexity of setting up tools for the deployment channel, as well as reluctance to change and a lack of clear liabilities, were among the DevOps hurdles that needed to be solved. Erich et al. (2017) describe issues that arise when promoting DevOps. They created a demand for a diverse range of specialized and interpersonal skills, as well as collaboration. This is necessary since the

development and operational channels contain personnel from several disciplines who contribute to the pipeline's specific remnants and components.

Major Gaps of Related Works

All of the previous research has revealed the implicit benefits, as well as the problems and practices related with DevOps. They also aid in highlighting the relative positions of culture and practical implementation in various software diligence. Nonetheless, that literature examines unique issues and challenges in several advanced countries. When it comes to developing countries, the circumstance that makes borrowing and applying new technology so difficult encounters even greater hurdles, and the country's major problems become even more convoluted. As a result, DevOps societies, advantages and/or practices, and difficulties should be considered in the context of a certain country's working environment and people's mindfulness capacity. Accordingly, this study concentrated on Ethiopian software development firms in order to fill some research gaps on developing countries' approaches to adopting and customizing DevOps implementation processes.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Overview

This chapter describes and explains the research methodology that was used throughout the research process, as well as why it was important. First, the study's overall research approaches are explained. Second, the research design selection is presented. Third, the researcher discussed the study's population and sampling; data collection methods and procedures; data analysis approaches and techniques; and the study's validity and reliability.

3.2. Research Method Approach

Procedures for collecting, analyzing, interpreting, and reporting data in research studies are known as research designs. There are three types of research methods approaches to conducting research: qualitative, quantitative, and mixed methods (Creswell, 2014). In order to answer the designed research questions, the researcher used a mixed research method approach, which is a combination of important elements of quantitative research and qualitative research. Mixed methods research can provide a more complete picture than a standalone quantitative or qualitative study because it combines the benefits of both methods.

3.2.1. Research Design

Following the decision to use a mixed methods approach for a study, the next step is to select the specific design that best addresses the research problem. As a result, the researcher chose an explanatory research design. The Explanatory Research Design (also known as the Explanatory Sequential Design) is a two-phased mixed methods design.

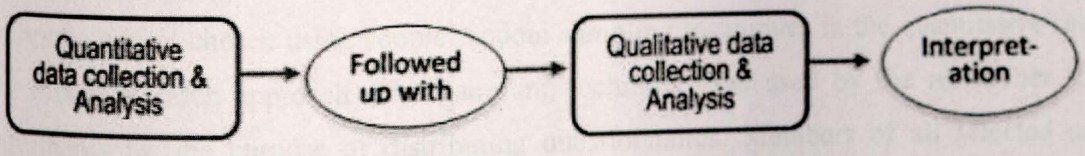

Figure 4: Explanatory Sequential Mixed Methods Design (Creswell, 2012)

The overall goal of this design is for qualitative data to help explain or build on preliminary quantitative results (Creswell, et al., 2003), and it also aims to provide relevant information required to understand the research problem more efficiently. It begins with quantitative data collection and then moves on to qualitative data collection in order to help explain or elaborate on the previously obtained quantitative results.

3.2.2. Source of Data

The data sources were divided into two categories: primary and secondary. Primary data is information that can be gathered directly from informants as a first-hand source. Secondary data, on the other hand, is information gathered from documents and third parties. Secondary sources of data were only used for this study to obtain information about the number of registered ICT companies in Ethiopia from the Ministry of Information Technology. Because the information was extracted from the document, it is classified as secondary data. Except for this information, almost all of the information in this study came from primary sources. All information was gathered from key informants through the distribution of structured questions and face-to-face interviews with primary target informants about the practices and challenges of the DevOps implementation process. In this regard, for both quantitative and qualitative data collection, the researcher relied solely on first-hand informants.

3.2.3. Sampling Techniques

In this explanatory sequential mixed method research design, quantitative research was carried out first, followed by a qualitative phase. As a result, the sample method used to select informants for a quantitative and qualitative approach differs. Because the data was collected separately, informants were selected using various techniques, as briefly explained in the following section.

3.2.3.1. Simple Random Sampling Techniques

Informants were chosen using simple random sampling techniques in the quantitative section of the mixed research approach. This sampling technique was used by the researcher to select informants for the purpose of distributing questionnaires. Members of all selected software companies' DevOps implementation teams had a chance of being chosen as first-hand

informants. As a result, the researcher determined that using a simple random sampling method, all members of the DevOps implementation team were taken as representative for the study. Because the DevOps team's staff was small, no other systematic selection methods were required; rather, all of them had a chance of being included in the research and could be easily managed by the researcher.

3.2.3.2. Purposive Sampling Techniques

To cover the qualitative section of the explanatory sequential research design, the researcher used the purposive sampling technique to select informants. The researcher carefully selected experienced and senior staff from the selected software companies to conduct interviews in order to collect data. This is because senior staff, managers, directors, and other team members were purposefully chosen to obtain comprehensive information about the practice and challenges of DevOps implementation. As a result, all informants were purposefully chosen to obtain qualitative information based on open-ended interview questions designed to achieve the study's objectives.

3.2.4. Target Population

The Ethiopian software development companies that have fully or partially implemented the DevOps process in their organizations are the study's target population. Companies that have not yet implemented DevOps in their organizations are excluded from this study because the objective of the study is to identify DevOps implementation practices and challenges. According to the Ministry of Information Technology, there are currently 435 registered ICT companies in the information and communication technology sector. Only 143 of the 435 registered ICT companies are software development companies. However, not all of the 143 software development companies used DevOps in their operations. DevOps is used entirely or partially by some businesses.

As a result, the researcher chose 13 of the 143 companies based on their current stage of DevOps implementation. After gathering information via cell phones and making physical contact, the researcher chose these 13 companies using a purposive sampling technique. After a physical inspection, the researcher chose ten of the thirteen companies. Due to cooperation constraints,

three companies were excluded from the survey for various reasons. The remaining ten are eager to contribute data and take part in the survey for this study. As a result, all employees of software development firms involved in the DevOps implementation process to improve development, quality assurance, and IT operations contributed to the data collection for this study.

Table 1: The number of respondents and participants

Software Companies	Number of Respondents/Participants
Company A	12/15
Company B	7/8
Company C	14/20
Company D	36/50
Company E	4/4
Company F	6/7
Company G	6/8
Company H	6/6
Company I	9/16
Company J	7/11
Total	107/145

NB. The name of the selected companies is anonymous due to security purposes. They did not give consent to mention their name in this research.

3.3. Data Collection Instruments

Data collection is the process of gathering and measuring empirical information on variables of interest using established scientific tools to answer the stated research questions. As a result, closed-ended structured questions were developed to collect quantitative data, while open-ended semi-structured questions were produced to collect adequate qualitative data. As a result, both the questionnaire and the interview were used as data collection tools in the designed explanatory sequential mixed research method. The section that follows briefly describes the data collection procedures and tools for integrating the research questions with empirical data.

3.3.1. Questionnaire

A questionnaire is one of the basic research instruments or tools used to collect data in a quantitative research approach. As a result, questionnaires are typically used to collect straightforward information that is brief and unambiguous (Johannesson and Perjons, 2014). The data collection instrument was primarily developed through a combination of reviews of literature relevant to meeting the study's objectives. As a result, a type of psychometric response scale was used in which respondents specified their level of agreement to a statement, typically on a 5-point Likert scale such as 1-strongly disagree, 2-disagree, 3-neutral, 4-agree, and 5-strongly agree. As a result, the questions for the DevOps team were adapted using Likert scale questioners.

The following two-step techniques were used to develop the main questionnaire for quantitative phase of study.

First, the questionnaire items were chosen from the literature. Minor adjustments were made to some of the questions to better reflect the study's objective. However, most questionnaires newly developed by the researcher to conduct data based on the statement of the problems and proposed research questions. Furthermore, based on the studied material, some additional questions were suggested. Second, the questionnaire was tested in a pilot research with 12 volunteer respondents who worked in the projected Ethiopian Software Companies' DevOps implementation team. The goal of the pilot study was to see if the questionnaires gave reliable and consistent results. As a result, based on the comments of the respondents, changes were made to the questionnaire items in order to eliminate bias and improve accuracy.

Then, after the feedback of the pilot study, all questionnaires were thoroughly revised and distributed to the selected informants to accommodate the respondent's experience and obtain important information. Informants were in various team structures in the selected software development companies such as development team, quality assurance team, and DevOps team. As a result, the questionnaire was distributed to all team members in software development companies, with the goal of identifying DevOps implementation practices and challenges in the Ethiopian software development industry.

3.3.2. Interview

The other main tool, which is basically used to collect data from informants in qualitative phase explanatory sequential research design of mixed research methodology, is an interview. This research tool was applied to cover the qualitative section of the study. The study consists of open-ended semi-structured guiding questions were used to collect empirical data from informants.

Using the guiding semi-structure questions, the interview was made by selected the informants using purposive sampling techniques. Accordingly, the interviewees were DevOps experts who work in the DevOps team of their software development companies or software development team leaders who have experience in DevOps implementation. After the researcher selected main informants purposively, the next step was contacted the selected respondent's of the DevOps implementation senior staff expertise to debrief the objective of the study. One of the major contacting methods to have permission was internet methods like telegram and google mail.

After getting appointment, the interview has been conducted face-to-face and taken place after a brief introduction regarding the objective of the study and why the researcher was motivated to study DevOps implementation practices and challenges, as well as the expected benefits of the proposed study. The researcher was taking some notes while interviewing and used record with smart phone recorder based on the consent of the interviewee.

3.4. Data Collection Procedures

3.4.1. Getting Research Permission

Before contacting the informants, getting research permission from the appropriate bodies was required. Addis Ababa University's School of Information Science and Department of Information Systems granted permission, and they provided supportive letters to specific authorized offices and organizations that requested cooperation during data collection. As a result, letters were sent to the Ministry of Information Technology as well as the ten Ethiopian software development firms that agreed to take part in this study. Following receipt of the one of the letters granting permission to conduct research, the researcher contacted all selected Ethiopian software companies, followed by informants.

3.4.2. Pilot Study

After granted research permission, a pilot study was conducted with a sample of 12 purposively selected respondents in order to test the validity and reliability of the proposed questionnaires. In addition, it helps to ensure that the instruments are free of ambiguity and irrelevant items. Pilot study also helps for controlling bias in data interpretation prior to disseminating the survey to the actual full-scale group. The pilot survey participants were selected from four software development companies. Participants were works on software developments and IT operations in their companies. One to one discussion was held with the pilot study respondents prior to distributing the questionnaires. All of the participants filled out the questionnaire, which indicated a 100% response rate of the pilot study. Once the questionnaire was filled out, the feedbacks were gathered from the participants. In accordance with the pilot test feedbacks, the questionnaire was amended to improve the clarity of the questions, minimize data interpretation bias and increase the likelihood of success. In addition, to measure the internal consistency of the questionnaire, a reliability test was conducted. Accordingly, Cronbach's coefficient alpha was used as a reliability criterion. Moreover, Cronbach's alpha is the most common measures of the reliability of the internal consistency. As Norman (2003:219), this coefficient ranges between 0 and 1, with a high value indicating a high level of consistency among the items. In view of that, the result (0.813) obtained is greater than 0.70 which is the minimal alpha value to prove the internal consistency and reliability. Hence, the Cronbach's alpha test is an indication that the survey questionnaire's reliability and internal consistency to use for the study as can be seen in the table below.

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.813	.822	31

Figure 5: Reliability Statistics (Source: Own survey, 2022)

3.5. Data Analysis

The analysis in the explanatory sequential design began with a quantitative survey study, which identified statistically significant differences and anomalous results. They then conducted an in-depth qualitative study to explain why these results occurred. Following this first phase, qualitative data were collected and analyzed. The second, qualitative phase of the study is planned to build on (or connect to) the findings of the first quantitative phase.

As a result, descriptive statistics methods such as frequency distribution, mean calculation, percentages, tabular and graphical representations are used to summarize the collected quantitative data. Similarly, qualitative data was analyzed using thematic analysis, which involved organizing and preparing the data into themes from the beginning of data collection. The researcher rehearsed and edited this method of analysis to avoid unnecessary information redundancy. Furthermore, an OpenCode software tool with a narrative style was used to conduct thematic analysis, demonstrating both the practices and challenges of DevOps implementation.

Based on the results of the analysis and existing literature, the discovered practices and challenges during DevOps implementation were investigated. Furthermore, using the quantitative and qualitative analysis results, a successful DevOps implementation guideline was proposed that can address the identified DevOps implementation practices and challenges in order to improve DevOps implementation practices in the Ethiopian software industry. As a result, the use of a well-executed approach to evaluating the quality and acceptability of a research result was regarded as a critical component of the information system research process.

3.6. Validity and Reliability

Data validation, according to Nahid (2003), is "a process that validates the correspondence of the final data with a variety of quality attributes." Its goal is to ensure a particular degree of quality by comparing, cohering, and completing the following dimensions: relevance, correctness, timeliness, and punctuality, accessibility and clarity, comparability, and completion. Furthermore, in qualitative ways to analyze or control bias and establish valid proposals, the triangulation method is an important methodological subject. Because the quantitative data

questionnaire items were adapted from the literature, they were statistically validated to confirm the data's validity.

In this study, prior to releasing the questionnaire to the actual participants, a pilot study was undertaken to evaluate the reliability and validity of the questionnaire items. This helped to guarantee that the instruments were devoid of ambiguity and irrelevant items. Prior to spreading the survey to the actual full-scale group, the pilot research is also useful for controlling bias in data interpretation, and the technique of triangulation would be used. This entailed looking at material from a variety of sources and putting it together to form a logical explanation of concepts or themes. The member checking technique, on the other hand, was used to return the analytic results for a case study to the participants to see if they thought they were accurate.

3.7. Conceptual Framework

Because DevOps is a new technology that is an extension of agile, the key hurdles to practice comprehend the concept and the implementation process. Each organization has a unique environment and set of procedures, and each will confront unique problems while implementing DevOps. Understanding the issues that companies confront and how they attempt to address them is critical. One key question that may emerge in this regard is, "What is the best practice that can be implemented in DevOps?" As a result, enterprises will be able to overcome these obstacles and successfully deploy DevOps within their organizations. The researcher proposes a conceptual framework for DevOps implementation to identify practices and obstacles. This thought arose from our ideal concept and other literature. DevOps is a new technology in software development businesses that aims to improve delivery by expanding agility practices within a collaborative culture.

In the Ethiopian software development industry, four dimensions are required to identify the problems and practices of DevOps implementations. They include technology and tools, as well as procedures, people, and culture. Culture and people are more related to the management perspective notion, whereas technology, tools, and procedure are more related to the engineering perspective concept. As a result, the conceptual map is shown in figure 6 below.

- **Technology and Tools**: This component examines the technologies and tools that assist the DevOps process in continuous delivery and collaboration between the development and operations environments.

- **Process**: Continuous delivery, continuous testing, continuous integration, and continuous monitoring are all part of the DevOps process. In software development, DevOps processes use agile practices.

- **People:** DevOps team members should be skilled individuals who have a strong ability to improve their abilities through self-learning and team-learning. Team members should collaborate closely and help one another.

- **Culture**: DevOps is a culture of collaboration among all team members. DevOps requires a new culture that promotes transparency and collaboration between development and operations teams.

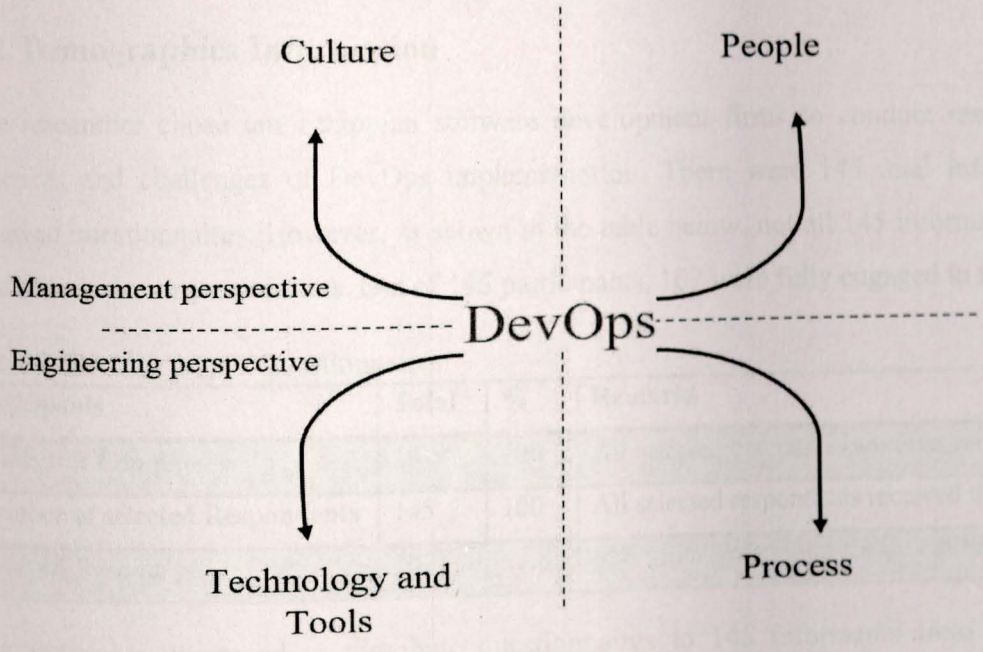

Figure 6: DevOps Implementation Conceptual Maps (Leite et al. 2019)

CHAPTER FOUR

4. DATA PRESENTATION, ANALYSIS, AND DISCUSSION

4.1. Overview

Based on the objectives, this chapter presents, analyzes, and discusses data from various sources. First, respondents' demographic information is presented, such as their educational level, work experience in software development and/or IT operations, whether they have implemented DevOps, and how they have implemented DevOps in their companies. Second, the descriptive analysis of quantitative data was presented first, followed by the presentation and analysis of qualitative data using the thematic analysis method, in accordance with the explanatory sequential research design used in the mixed research approach. Following the completion of the quantitative and qualitative data presentation and analysis phases, a combination of data was discussed in order to strengthen and validate the study findings. Finally, based on the findings, a guideline was presented as an expert suggestion to improve the challenges of DevOps implementation.

4.2. Demographics Information

The researcher chose ten Ethiopian software development firms to conduct research on the practices and challenges of DevOps implementation. There were 145 total informants who received questionnaires. However, as shown in the table below, not all 145 informants took part in the study for various reasons. Out of 145 participants, 107 were fully engaged in this study.

Table 2: Distribution of Questionnaires

Participants	Total	%	Remarks
Number of Companies	10	100	All selected companies are involved
Number of selected Respondents	145	100	All selected respondents received the questionnaire
Received Responses	107	73.8	Majorities questionnaires are collected

The researcher attempted to distribute questionnaires to 145 informants from ten software development companies. However, due to a variety of factors, it was only able to reach 115 respondents. Eight of the 115 respondents provided incomplete responses, while the remaining

107 provided valid responses. As a result, eight of them were rejected from the study, while the remaining 107 were represented as a study sample and included in the study. During this study, all respondents were worked in the software development, IT operations, quality control, or testing for their respective software companies. The data collection period lasted one month and twenty days

4.2.1. Demographic Findings

The demographic information gathered by the survey questionnaires was used by the researcher to draw conclusions about DevOps implementation practices and challenges among developers and IT operations. The information was gathered using three key parameters: educational level, work experience, and DevOps implementation. Educational level and work experience are important in understanding the demographic nature of the study population, whereas DevOps implementation is important in understanding how and why they use DevOps in their services. The researcher briefly discussed those parameters in the following sections to explain why they are important in this study.

Table 3: Demographics of participants

Selected Companies	Total IT Staff	Distributed Questionnaires	Collected Questionnaires						
			Education Level			Experience			
			MA/Msc	BA/Bsc	Diploma	<2yrs	2-5yrs	6-10yrs	>10yrs
Company A	15	15	4	8	0	4	2	6	0
Company B	8	8	1	6	0	3	1	2	1
Company C	20	20	4	10	0	1	4	6	3
Company D	50	50	7	29	0	5	17	13	1
Company E	4	4	0.	4	0	1	2	1	0
Company F	7	7	1	4	1	4	1	0	1
Company G	8	8	1	5	0	1	3	1	1
Company H	6	6	0	6	0	1	5	0	0
Company I	16	16	1	8	0	0	5	4	0
Company J	11	11	0	7	0	2	3	1	1
Sub Total	**145**	**145**	**19**	**87**	**1**	**22**	**43**	**34**	**8**
Grand Total	**145**	**145**	**107**			**107**			

NB. The name of the selected companies is anonymous due to security purposes. They did not give consent to mention their name in this research.

4.2.1.1. Respondents'' Educational Level

Knowing the respondents' level of education allows the researcher to assess their understanding of DevOps concepts and their ability to transition from the traditional way of implementing to the new culture of working with new technology. Employees resist the implementation of new technology due to a lack of education or a lack of proper management. Sometimes there is a knowledge gap of new technology due to low education level and difficulty accepting new ways of working. If the company has a high-level of education staff, DevOps implementation is either better or has some challenges due to training and workshop gaps rather than a knowledge gap. As a result, it assists the researcher in determining how education level affects DevOps implementation.

As a result, as shown in Figure 7, the majority of respondents in this study have a first degree, which represents (N = 87), or 81.3 percent of all respondents. Master's degree holders are the next largest category of respondents (N = 19), accounting for 17.8 percent of all respondents. Only one respondent (N = 1) has a diploma, accounting for 0.9 percent of all respondents. This demonstrates that the survey includes personnel of various educational levels to ensure that the sample is well represented.

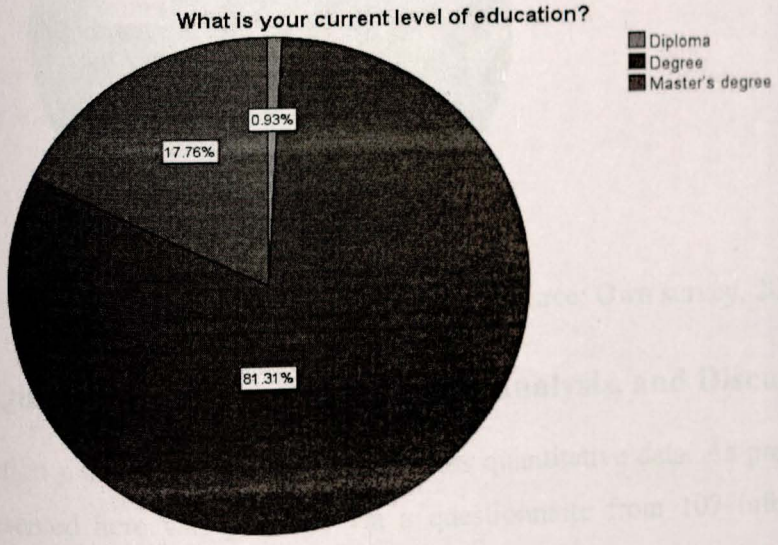

Figure 7: Educational Level of the Respondents' (Source: Own survey, 2022)

4.2.1.2. Respondents' experience or Service Year

This demographic information assists the researcher in determining whether the experience of software company employees influences DevOps implementation practices. Figure 8 depicts respondents' work experience in software companies. According to the data, nearly 40.19 percent of respondents have two to five years of experience, accounting for 43 of the 107 respondents. Around 31.78 percent of respondents have six to ten years of experience, accounting for 34 of the total number of respondents. Approximately 20.56 percent of all respondents have served for less than two years. Furthermore, 7.48 percent of respondents have more than ten years of work experience. This implies that respondents in the survey had sufficient experience in software companies to understand DevOps implementation practices and difficult processes in their software companies. This demonstrates the sample's representativeness in this study. Furthermore, it helps in determining the impact of employees' service years on DevOps implementation success.

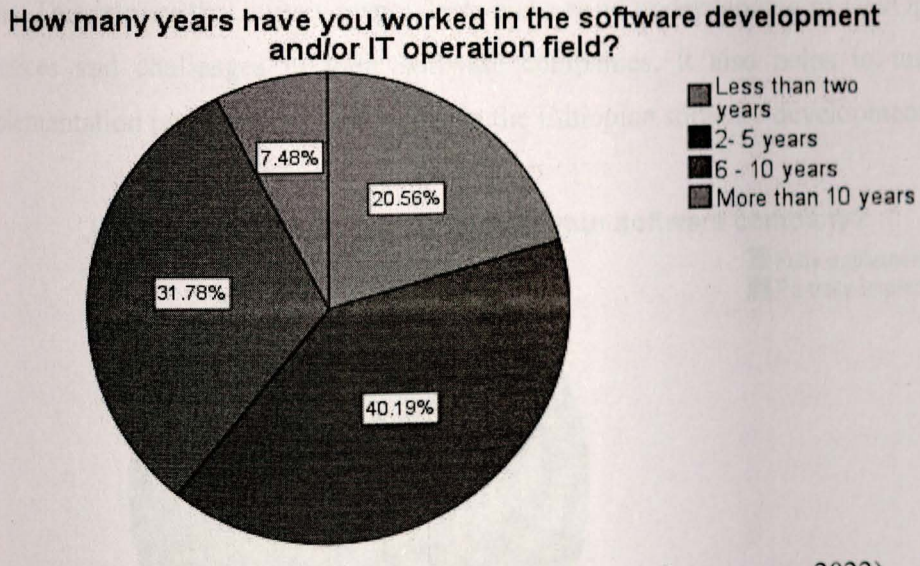

Figure 8: Educational labels of the respondents (Source: Own survey, 2022)

4.2. Quantitative Data Presentation, Analysis, and Discussion

This section presents, analyzes, and discusses quantitative data. As previously stated, all of the data presented here was gathered via a questionnaire from 107 informants. All of the data gathered was valued in order to demonstrate the practice and challenges of DevOps

implementation in the selected Ethiopian software companies. Following the question forwarded to the respondents, the major points of the findings are discussed briefly in the following sections. The first two sections focused on the respondents to assess their familiarity with DevOps practices and challenges, as well as to ensure that the selected software companies had practically adopted DevOps and how they implemented their services in accordance with the needs of their customers and markets.

4.3.1. Implementation of DevOps in the Software Companies

Discussing DevOps implementation in software companies is critical to determining employee exposure to the DevOps concept. Figure 9 depicts respondents' DevOps implementation in software companies. According to the graph, nearly half of the respondents (65.42%), or approximately 70 responses out of the total number of respondents, have DevOps partially implemented in their software companies. The remaining survey respondents represent approximately 34.58 percent of those who have fully implemented DevOps at their software firms. This implies that survey respondents had a solid understanding of DevOps implementation practices and challenges in their software companies. It also helps to understand DevOps implementation practices and challenges in the Ethiopian software development industry.

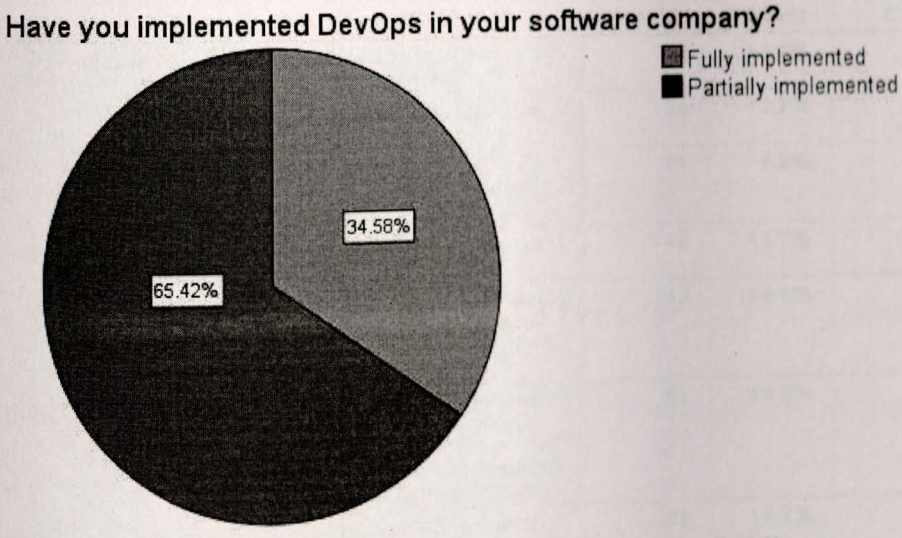

Figure 9: Educational label of the respondents (Source: Own survey, 2022)

4.3.2. Methods of DevOps Implementation in the Selected Software Companies

How does your company implement DevOps?

This question was presented to respondents in order to determine how Ethiopian software companies have implemented DevOps. The researcher attests to DevOps implementation in the selected software companies and is interested in expanding its use. According to Table 4, 19% of respondents who have implemented DevOps in their organizations need to increase the level of automation in testing by documenting the procedure for configuring and monitoring services. To implement DevOps in their company, 15.7 percent of respondents' software companies use Dev and Ops pairing to form one team and early incorporation of Ops into Dev projects. 14.9 percent of respondents said their company implements DevOps by increasing Ops competence in Dev teams, particularly in outsourced operations. In addition, nearly 14.2 percent of respondents said their organizations are implementing DevOps through organizational structural changes, such as the incorporation of feature teams. The remaining respondents describe how their company implements DevOps by forming a DevOps team and implementing infrastructure-as-code.

Table 4: Educational label of the respondents

How does your company implement DevOps?

		Responses		Percent of Cases
		N	Percent	
How does your company implement DevOps?[a]	1. Dev and Ops paring to form one team	42	15.7%	39.3%
	2. Formulating DevOps team	34	12.7%	31.8%
	3. Employing infrastructure-as-code practice	21	7.8%	19.6%
	4. Early incorporation of Ops in Dev projects	42	15.7%	39.3%
	5. Increase competence of Ops in Dev teams especially in outsourced Operations	40	14.9%	37.4%
	6. Need to increase the level of automation in testing, documenting the procedure for configuring and monitoring services	51	19.0%	47.7%
	7. Through organization structural changes e.g. incorporation of feature teams	38	14.2%	35.5%
Total		268	100.0%	250.5%

a. Dichotomy group tabulated at value 1.

4.3.3. DevOps Implementation Practices

Organizations are implementing a variety of DevOps practices in order to simplify their software development processes. According to several researchers, DevOps is a culture, people, process, technology, and tool combination that enables cross-functional collaboration between software development and IT operations teams to accelerate the delivery of changes made to resilient systems and to operate them (Nagarajan and Overbeek, 2018). DevOps is associated with both technical and non-technical practices, which are common collaborative practices involving people interactions and processes that are mostly automated through the use of technology and tools supported by the DevOps culture (Lwakatare et al. 2019). There are four dimensions in this analysis that investigate DevOps implementation practices. These are the practices of technology and tools, process, people, and culture. All of these dimensions will be discussed and analyzed briefly in the following sections.

4.3.3.1. Technology and Tools Dimension

In terms of technology and tools, DevOps implementation practices include deployment automation, software configuration management tools, DevOps environments, issue tracking tools, and DevOps technology capability. Table 5 summarizes the practices of DevOps implementation in terms of technology and tool dimensions, which are further discussed in the following section.

Table 5: Technology and Tools Dimension Practices (Percentage distribution and Mean)

DevOps Implementation Practices in terms of Technology and tools	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Deployment Automation	35.5	40.2	15.0	8.4	0.9	1.99
DevOps Environments	32.7	35.5	24.3	24.3	0.0	2.07
Issue Tracking Tool	52.3	30.8	10.3	4.7	1.9	1.73
Software Configuration Management Tools	33.6	44.9	16.8	4.7	0.0	1.93
DevOps Technology Capability	25.2	39.3	21.5	14.0	0.0	2.24

Source: Own survey, 2022

The technology and tools dimension was created to learn how technology and tools can intervene and simplify the processes of successful DevOps implementation in software development firms (Nagarajan and Overbeek, 2018). Practices in technology and tool dimension have a significant impact on successful DevOps implementation. The majority of respondents agreed that technology and tools are important, particularly in their DevOps workflow. Companies that use the right technology practices have a better chance of growing their daily operations while remaining competitive in their industries and markets. One of the most common inventions that have provided a means of improving day-to-day corporate operations is technology. DevOps tools and practices provide appropriate technology support for successful DevOps implementation. The aggregate mean (mean of the mean) result for the technology and tools dimension practice is 1.992, putting it in the agreement category.

The majority of respondents (52.3 percent) strongly agreed that their software development companies used issue tracking tools based on the technology and tools dimension practices item questions. This suggests that Ethiopian software development companies use issue-tracking tools during the software development process. Tracking issues is a collaborative process. For any process to be successful, each team member must be aware of and understand their role and responsibilities within the process. Once issues have been identified, the head of product development must assign them to team members and drive solutions. This is the benefit of issue tracking tools.

Furthermore, deployment automation, including DevOps environments, is a DevOps approach in the deployment pipeline that aims to eliminate or reduce manual system handovers from the development team to the operations team. The majority of respondents agreed that deployment automation, DevOps environment management, and software configuration management tools are used in the software development process by their software development companies. This agreement certifies that all employees in their software development company understand DevOps tools and have a solid understanding of DevOps technology.

As communication and collaboration, continuous deployment, continuous delivery, continuous planning, automated pipeline, and quality assurance are all conceptual elements that describe the

complex phenomenon of DevOps, the process dimension that will be analyzed below has been possible to achieve.

4.3.3.2. Process Dimension

DevOps implementation practices in terms of process dimension include continuous development, continuous testing, continuous deployment and delivery, and continuous integration, monitoring, and feedback. Table 6 summarizes the above practices, which are also discussed further below.

Table 6: Process Dimension Practices (Percentage distribution and Mean)

DevOps Implementation Practices in terms of Process	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Continuous Development Process	55.1	33.6	4.7	5.6	0.9	1.64
Continuous Testing Process	49.5	29.0	13.1	8.4	0.0	1.80
Continuous Deployment and Delivery Processes	50.5	33.6	10.3	5.6	0.0	1.71
Continuous Integration, Monitoring, and Feedback Processes	38.3	38.3	15.9	7.5	0.0	1.93

Source: Own survey, 2022

DevOps is a set of practices that aim to produce error-free software while also shortening the time between committing a change to a system and its implementation. The DevOps method consists of two practices: (1) collaboration between development and operations; and (2) management of the deployment environment and automation in agile environments. Process dimension practices have the greatest impact in a successful DevOps implementation. According to survey results, the aggregate mean (mean of the mean) for the process practice is 1.77, placing it in the strongly agreed category. This implies that the majority of Ethiopian software development companies have well-defined practices in the continuous development, testing, deployment, and delivery processes, as well as continuous integration, monitoring, and feedback.

On the process dimension practices, the main components of DevOps implementation processes are continuous development, continuous testing, continuous deployment and delivery,

continuous integration, monitoring, and feedback. According to the respondents, the majority of software companies (55.1 percent) use continuous development; almost all software companies use continuous development in their software development process. Continuous development is a software development process that includes several DevOps processes such as continuous integration, continuous testing, continuous delivery, and continuous deployment.

As shown in Table 6, the continuous testing process accounts for 49.5 percent of the software development process, which is one of the most important aspects of ensuring product quality before it is deployed to the end user. The goal of continuous testing is to provide quick and continuous feedback on the amount of business risk in the most recent build, which is used to determine whether the software is ready to proceed through the delivery pipeline at any given time. According to the findings in Table 6, 50.5 percent of respondents strongly agree that most software companies use continuous deployment and delivery processes. Continuous deployment and delivery is a software engineering approach in which teams produce valuable software in short cycles and ensure that it can be released reliably at any time. Continuous deployment is a DevOps practice that complements continuous delivery by automatically deploying new changes to production. Furthermore, continuous delivery is a DevOps practice that focuses on maintaining a software solution in a releasable state at all times.

The components of continuous integration, monitoring, and feedback processes received equal responses in the Likert scale of agree and strongly agree, accounting for 38.3 percent of all respondents. Continuous integration is a DevOps technique in which developers merges code into a versioned source control repository on a daily or more frequent basis. Continuous monitoring is a stage in which the operational variables of the entire DevOps process are included. This is the location where a large amount of critical data about the application's use is saved and meticulously prepared with the goal of discovering areas of error and observing the current patterns that include the application. Customer service receives immediate feedback from customers about a product's features and causes them to change accordingly. This can provide a reasonable idea of the software applications that should be used at various stages to gradually improve this product for a product-controlled organization.

4.3.3.3. People Dimension

In terms of the people dimension, DevOps implementation practices include team organization, continuous learning practices, competent and capable employees, and qualified DevOps employees. These practices are summarized in table 7 and discussed further below.

Table 7: People Dimension Practices (Percentage distribution and Mean)

DevOps Implementation Practices in terms of People	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Team Organization	54.2	37.4	7.5	0.9	0.0	1.55
Continuous Learning Practice	50.5	36.4	11.2	1.9	0.0	1.64
Competent and Capable Employees	45.8	40.2	12.1	1.9	0.0	1.70
Qualified Employees with DevOps	19.6	41.1	26.2	12.1	0.9	2.34

Source: Own survey, 2022

DevOps is based on the concept of bringing together people from development and operations through a collaborative culture. DevOps allows team members to have a cross-functional skill set because DevOps implementation teams are made up of many trained experts, such as developers, quality assurance, and IT operations. These cross-functional skills include all of the various skill sets required to complete an end-to-end task for a software product or service. As a result, people are critical to the successful implementation of DevOps. Continuous learning is viewed as a process rather than an event that occurs only once in a lifetime. Organizations may need to develop a well-organized and structured learning strategy for their employees and engineers in order to improve their knowledge of the DevOps implementation process. The aggregate mean (mean of the mean) result for the people dimension practice is 1.8075, which is classified as strongly agreed.

Most respondents strongly agreed that their software development companies used team organization, continuous learning practices, and competent and capable employees across each team in the DevOps implementation practices in people dimension item questions. Furthermore, the majority of respondents (41.1 percent) agreed that their software development companies

have qualified DevOps employees. According to empirical data, more than half of respondents (approximately 54.2 percent) strongly agreed with the practice of organizing teams in software development firms. Because there is good collaboration and communication across the teams, team organization aids the development and IT operations processes in software development.

4.3.3.4. Culture Dimension

In terms of cultural dimension practice, DevOps implementation practices include communications and collaborations, DevOps cultural understanding, innovation drivers, and cultural capability practice. These cultural DevOps practices are summarized in table 8 and discussed further below.

Table 8: Culture Dimension Practices (Percentage distribution and Mean)

DevOps Implementation Practices in terms of Culture	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Communications and Collaborations	46.7	45.8	6.5	0.9	0.0	1.62
Cultural Understanding	33.6	46.7	15.9	3.7	0.0	1.90
Innovation Drivers	29.9	37.4	27.1	4.7	0.9	2.09
Cultural Capability Practice	15.0	48.6	33.6	1.9	0.9	2.25

Source: Own survey, 2022

DevOps implementation begins with culture. A culture of information sharing characterizes DevOps, and information sharing has enormous potential for improving software quality. Culture is important in every organization because it affects how employees work and who is responsible for ensuring the quality of the end product (Alok & Ziadoon, 2020). DevOps is first and foremost a practice and culture that fosters strong collaborative bonds between software development and IT operations teams across the organization. The aggregate mean (mean of the mean) result for the Culture dimension practice is 1.965, placing it in the agreement category. This implies that Ethiopian software development firms are well-versed in DevOps cultural practices.

As a result, 46.7 percent of respondents strongly agreed that communication and collaboration are commonly used across teams in their software companies. Communication and collaboration

are practices established in DevOps implementation to bring teams closer together by building workflows and distributing responsibilities. The majority of respondents (46.7 percent) agreed that there is enough cultural understanding and cultural capability practices (48.6 percent) in software companies. There are also drivers of innovation (37.4 percent) in software companies.

4.3.3. DevOps Implementation Challenges

Due to various challenges, DevOps may not always be successfully implemented in software development companies. Respondents identified critical DevOps implementation challenges related to technology and tools, the DevOps implementation process, employees' technical capability and understanding, and organizational cultures.

4.3.3.1. Technology and Tools Challenge

The DevOps development pipeline is founded on a technology stack that enables automation, efficiency, communication, and collaboration. DevOps tools can also automate planning, development, testing, deployment, operations, and monitoring. Furthermore, some tools can orchestrate entire pipeline processes and have a view of the entire DevOps pipeline. The most difficult task for developers in DevOps implementation was managing new types of tools and circumstances.

Table 9: Technology and Tools Challenge (Percentage distribution and Mean)

Technology and Tools Challenge	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Using DevOps technology and tools	8.4	26.2	29.0	30.8	5.6	2.99
Access to production systems by DevOps	12.1	29.9	25.2	28.0	4.7	2.83

Source: Own survey, 2022

According to the findings in Table 9, using DevOps technology and tools is not a major challenge for DevOps implementation. In response to the question about technology and tools as a challenge, 30.8 percent of respondents said it is not a challenge, while 26.2 percent agreed it is. Respondents to this item debated whether or not to consider it a challenge. Thus, respondents'

argumentative responses imply that using technology and tools for DevOps implementation is a challenge in Ethiopian software development companies, but that this challenge has plenty of opportunities to be solved through simple technical training. As shown in Table 9, the mean for using DevOps technology and tools is 2.99, which is in the same range as the aggregate mean of 2.91, which is classified as neutral. According to the respondents, access to production systems by DevOps (29.9 percent) is a challenge for DevOps implementation in Ethiopian software development companies.

4.3.3.2. Process Challenge

DevOps has a process for implementing any delivery, from the beginning to the end. There are numerous challenges that affect the success of DevOps implementation during this process. Thus, implementing DevOps practices enables an organization to deliver software that is faster, better, higher-quality, and more reliable. A culture of cooperation and collaboration among all functions of the organization is required for successful DevOps. This temporarily increased the difficulties of the DevOps implementation process life cycle. This is due to the fact that the DevOps implementation process includes integrating teams and breaking down silos within IT departments. The journey begins with establishing a vision of how this will work for the specific business company. Understanding the roles and responsibilities of where development stops and operations begins, and how these can best be integrated together, is the major challenge in the DevOps implementation process.

Table 10: DevOps Implementation Process Challenge (Percentage distribution and Mean)

DevOps Implementation process challenge	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Increasing the complexity of the build process in DevOps	13.1	26.2	32.7	24.3	3.7	2.97
Adapting to organizational processes to DevOps	11.2	29.9	23.4	27.1	8.4	2.92

Source: Own survey, 2022

The agility and automation are central to the DevOps process flow. Each stage of the DevOps lifecycle focuses on closing the development-operations gap and driving production through

continuous development, integration, testing, monitoring, feedback, delivery, and deployment. This DevOps implementation lifecycle process increases the complexity of the DevOps implementation. According to this understanding, increasing the complexity of the build process in DevOps implementation is a challenge for the selected Ethiopian software companies, with 13.1 percent strongly agreeing and 26.2 percent agreeing. However, approximately 32.7 percent of respondents are neutral, meaning they neither agree nor disagree. Respondents' neutrality indicates that they either do not understand the processing lifecycle of DevOps implementation or are under managerial bureaucratic pressure and are afraid of the consequences. The other issue that respondents strongly agreed on (11.2 percent) and agreed on (29.9) is adaptation, while the aggregate mean (mean of mean) is 2.945, which is rated in the neutral category and indicates a high level of consistency.

4.3.3.3. People Challenge

One of the most difficult aspects of the DevOps implementation process is the human aspects—the people, which include understanding the DevOps concept, implementing with capable members of the DevOps team, obtaining management support, and gaining professional experience. The most critical challenge for DevOps implementation has been identified in various literatures as a lack of collaboration and communication. Developers and IT operations teams do not share common goals and plans, making appropriate communication difficult and resulting in software delays (Katal and Dahiya, 2019). When people come from different professional and personal backgrounds, it can be difficult to communicate effectively. According to Table 11, 37.4 percent of respondents agreed that one of the major challenges for DevOps implementation is a lack of professional experience.

16.8 percent of respondents strongly agreed that poor management has the potential to impede proper DevOps implementation, while the majority of respondents (30.8 percent) were neutral in offering any suggestions for addressing this challenge. This alone indicates that the DevOps organization team in Ethiopian software development companies faces some challenges in defending their leaders as bureaucracy obstructs their future benefits. They may become disoriented and lose sight of their goals if they are not properly managed. As a result, poor management will be unable to create a culture of mutual cooperation, which is a significant step

because it requires people to evaluate and rearrange their perceptions of their teams, organizations, and clients.

Table 11: DevOps Implementation People Challenge (Percentage distribution and Mean)

DevOps Implementation People Challenge	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Understanding of the DevOps Concept	11.2	24.3	22.4	26.2	15.9	3.11
Getting DevOps capable members	12.1	33.6	23.4	24.3	6.5	2.79
Obtaining management support for DevOps Practices	16.8	24.3	30.8	22.4	5.6	2.76
Finding experienced professionals to support DevOps Practices	20.6	37.4	24.3	15.0	2.8	2.42
Identifying The responsibility of DevOps between the developers and IT operations	9.3	31.8	28.0	23.4	7.5	2.88
Adopting Employees' mindsets to achieve successful DevOps implementation	15.9	35.5	27.1	16.8	4.7	2.59

Source: Own survey, 2022

People face challenges such as psychological safety and fostering a culture of trust. In this case, leadership behavior drives the transformation and serves as a good model for the DevOps implementation process. The major challenges, according to this understanding, are ways to investigate how the team perceives the organization's performance in terms of alignment, accountability, autonomy, empowerment, and feeling supported. Some of the companies chosen for this research look at dynamic learning and constructive collaboration, but basic challenges emerge from the flow through the lens of customer focus and value stream thinking about clients' lack of capable members (33.6 percent agree). These difficulties include the generation, receipt, and review of feedback in retrospectives.

This is due to a lack of clearly defined responsibilities (31.8%) during the DevOps implementation process. The adoption of employees' mindsets to achieve successful DevOps

implementation is the other major challenge, with 35.5 percent agreeing and 15.9 percent strongly agreeing. As a result, as 56 percent of the findings in the published article indicate, a lack of skill and knowledge is a significant challenge in the process of adopting employees' mindsets in various software companies (Yiran and Yilei, 2017). Some organizations lack skilled employees because implementing DevOps practices necessitates both development and operation skills and knowledge, as well as a mindset shift.

4.3.3.4. Culture Challenge

The most difficult challenge for most software companies on the path to DevOps implementation and practices is culture. In the various literature we have seen, there is a lack of technical knowledge and understanding of the key concepts, methods, tooling, and key benefits and challenges of implementing DevOps. Many organizations lack the necessary training and motivation to learn DevOps, and people are only interested in their area of expertise, creating a slew of issues. Criticism practices have highlighted 50% frequency as a critical challenge that harms the DevOps culture in the published literature (Muhammad et al. 2022). A blaming culture frequently focuses on punishment, bad fights, finger pointing, and creating negative perceptions that lead to destructive behavior, causing friction in the workplace. The most critical issue in implementing DevOps in an organization is that people oppose attitudes that are typically an impediment to organizational change success.

During the implementation phase of DevOps, the working culture changes. Because DevOps is a culture in and of itself, it collides with the existing work culture during implementation. Change, as previously stated, is difficult. Employees must devote a significant amount of time to assimilating into a new culture. Working culture transformation is not a quick process. As a result, as a business owner, you must address this issue before fully implementing DevOps in your software development process. You can begin with small DevOps training programs to get your team on board with the concept. Building a team of pro-DevOps influencers within the organization is always a good idea.

Table 12: DevOps Implementation Cultural Challenges (Percentage distribution and Mean)

DevOps Implementation Cultural Challenges	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Mean
Achieving Compatibility between DevOps and legacy system in the organization	14.0	29.9	27.1	27.1	1.9	2.73
Changing company culture to support DevOps implementation	19.6	37.4	23.4	15.9	3.7	2.47
Communication and collaboration between the development and IT operation team	6.5	39.3	21.5	23.4	9.3	2.90

Source: Own survey, 2022

According to the responses, the majority of respondents (29.9 percent) agreed that achieving compatibility between DevOps and legacy systems in the organization is a major challenge for DevOps implementation in Ethiopian software companies. Although 27.1 percent of respondents were neutral or disagreed with the challenge, the aggregate mean (mean of the mean) result is 2.7, indicating which is the high value of consistency. On these challenges in Ethiopian software companies, 1.9 percent and 14.0 percent strongly disagreed and strongly agreed, respectively. One of the most difficult challenges (37.4 percent) is changing company culture to support DevOps implementation. The majority of respondents agreed that this is the most difficult issue in implementing DevOps in Ethiopian software companies.

19.6 percent of respondents strongly agreed that changing company culture to support DevOps implementation was a major challenge. This challenge's total mean is 2.47, which is the agreed-upon rating category of high reliability. Furthermore, communication and collaboration are major issues in DevOps implementation because, as the researcher previously discussed, DevOps is a combination of two departments, but successful product delivery is impossible unless they communicate with each other. Communication and collaboration between development and IT operations teams are viewed as a challenge by 39.3 percent of respondents. The majority of respondents agreed that this challenge was within Ethiopian software companies; however, the mean 2.9 of this challenge still indicates high consistency of neutrality based on the results of

neutral (21.5 percent) and disagreed (23.4 percent) respondents. This indicates that the DevOps implementation management system has irresolvable bureaucratic issues. The irresolvable bureaucratic issue indicates that the management system either lacks confidence in the changes brought about by DevOps implementation, as evidenced by the qualitative analysis below, or that traditional work culture has the potential to influence the managerial side.

4.4. Qualitative Data Presentation, Analysis, and Discussion

A qualitative data analysis is carried out in addition to the quantitative study to supplement and complete the survey findings. As a result, interviews were conducted with three DevOps team leaders and one developer team leader at their respective software companies to acquire information about DevOps' effective implementation practices and problems. Furthermore, it helped in the discussion of DevOps implementation practices and challenges with the dimensions of technology and tool, process, people, and DevOps implementation culture in the Ethiopian software development context. As a result, the interview output has been examined using open coding, as shown below. Patterns, according to Braun and Clarke (2006), are discovered through a rigorous process of data familiarization, data coding, and theme generation and refinement. Because there was a vast amount of interview material to manage, the researcher decided to use open source software to analyze the interviews thematically.

As a result, the researcher gathered actual data from Ethiopian software companies that adopt DevOps. Interviewees could wander from the questions, and the interviewer could widen or deepen the topic by asking follow-up questions. The researcher discovered many themes after reviewing the interview transcripts. The practice and challenges of DevOps deployment in the selected Ethiopian software companies were the higher-level subjects considered. As a result, Figures 10 and 11 depict the thematic map and the subjects addressed by the qualitative technique.

4.4.1. DevOps Implementation Practices

DevOps implementation strategies are widely used in a number of Ethiopian software companies. The researcher gathered information through interviews with company executives

about the procedures and benefits of DevOps implementation. The outcome of the interview is shown in Figure 10, and then the analysis proceeds.

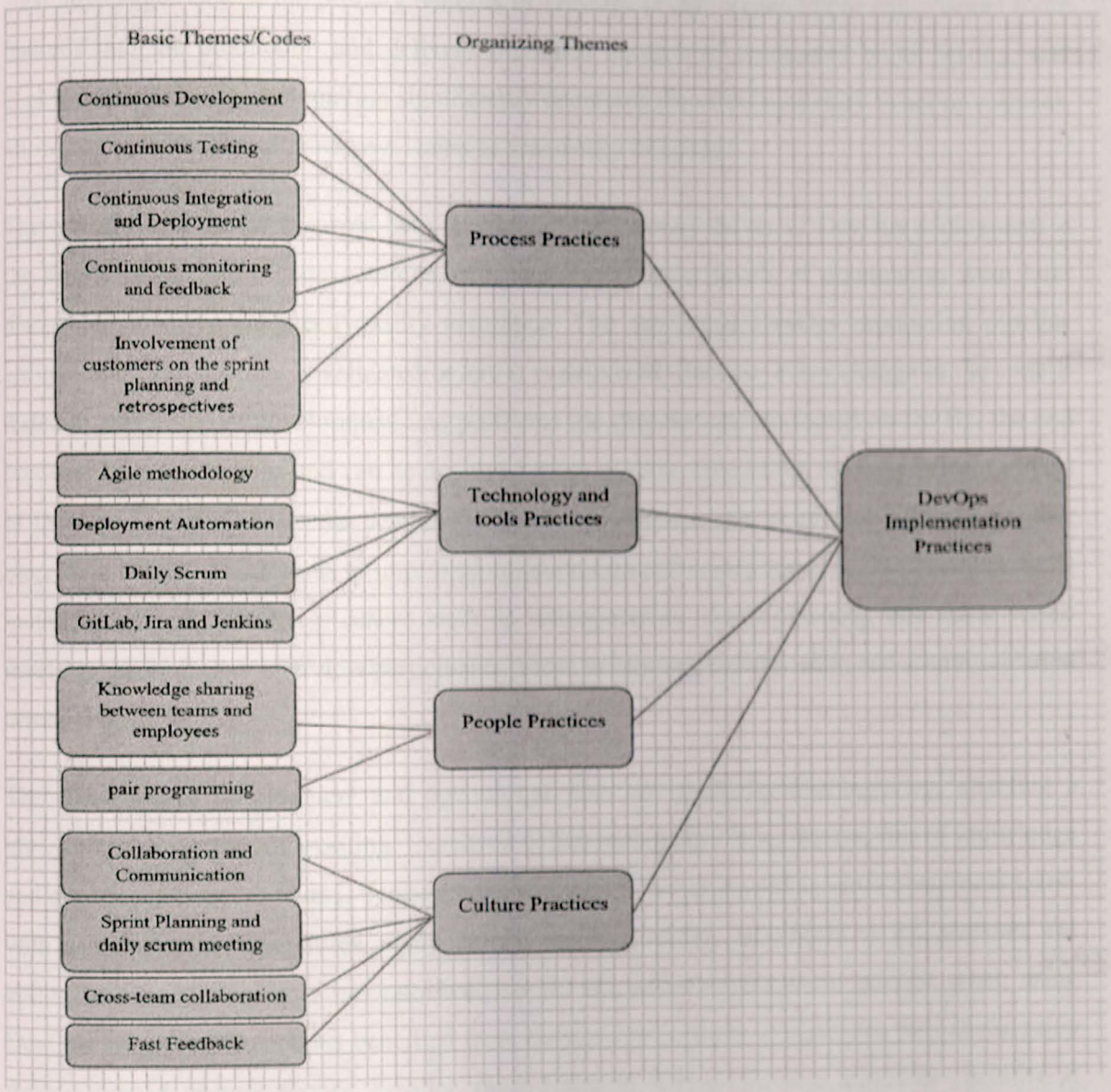

Figure 10: DevOps implementation practices related to issues of representation

Respondents believed that DevOps methods lead to a variety of other activities. DevOps was seen to positively enhance the rate at which software products could be provided and the quality of goods in general, for example, by strengthening feedback cycles. Internal engineering product

mechanics in the organization, including characteristics such as communication, were thought to be impacted as well. Respondents saw more implemented features and frequent releases as a clear advantage. One of the primary benefits of DevOps is that organizations can channel more features into the production and delivery pipelines because of automated build, testing, and deployment processes. Automation significantly decreases the effort necessary to set up releases, allowing organizations to produce releases as frequently as needed.

There are several things involved here, but one of the primary drives of DevOps deployment is the ability to get more code commits into each day. Higher levels of automation were later discovered to drive enhanced quality assurance. The automated DevOps production pipeline ensures that every change is validated before it is delivered. Because every change in the code is tested at every step of development and problems are detected and corrected on the fly, the end products have fewer bugs and can be deployed more quickly. One of the most significant effects of DevOps is that it forces the development and operations teams to contact with one another more than before, which leads to improved collaboration and communication. Furthermore, agile methodology and DevOps practices such as retrospectives, sprint planning, and daily scrum meetings, according to interview participants, serve to improve collaboration and communication across teams.

The conventional development and operational silos are gradually disintegrating, enabling a more cohesive manner of working. Increased collaboration speeds up the transfer of knowledge and experience between teams. Because a broader set of talents is used in multi-functional teams with a diversity of capabilities, competencies are maximized. According to the answers, the ability to release frequently has ramifications for the entire development process when using DevOps principles. Customers benefited from shorter development cycles since they could enjoy freshly created features sooner. Smaller, more frequent releases increase customer visibility of deployed features. Furthermore, because of the frequent releases, the development and operations teams are able to obtain early input from end-users and conduct testing with real consumers, which aids in the improvement of end products. Working with real customers provides organizations with a greater understanding of client preferences, allowing them to customize their products to market demands.

According to the respondents' perspectives, DevOps implementation strategies can help businesses swiftly test alternative concepts and make judgments. Continuous experimentation refers to the continuous testing of hypotheses in order to assess the value increases for both the consumer and the organization (Lindgren and Münch, 2015). Continuous experimentation, which necessitates well-established experimental infrastructure and environments, is supported and enabled by an efficient DevOps process. DevOps can also have a good impact on occupational welfare. A senior developer observed that frequent releases help to reduce stress levels because the anxiety associated with dealing with large releases is reduced. As a result, DevOps procedures not only benefit the organization but also improve the style of working, therefore positively contributing to the improved well-being of the DevOps team.

4.4.2. DevOps Implementation Challenges

DevOps implementation presents unique obstacles in the selected Ethiopian software companies. Figure 11 depicts the results of the interviews and discussions with chosen senior executives from the companies, as well as an analysis of the difficulties.

DevOps may not always be successful due to a variety of reasons. Respondents raised challenges related to communication and collaboration; organizational cultures; professional experiences; knowledge and skill gaps; resistance to new culture and technology; the adaptation of new tools and technology; and moving from traditional development to DevOps, which are not malleable; different constraints arising from the domain and environments; and the obscurity of DevOps' meaning. According to the majority of interviewees, one of the problems is the transfer from traditional development to the DevOps process, which necessitates the use of new tools and technology setups. As a result, with a complicated DevOps infrastructure and tools process, it is difficult to set up tools and technologies for the DevOps process. Furthermore, insufficient communication and teamwork is a major barrier to successful DevOps adoption. For example, it was claimed that operations teams do not always monitor or pass all of the performance and other metrics that developers might use, which can cause issues. In suboptimal circumstances, operations engineers and developers are concerned with separate, and potentially contradictory, metrics: operations people are concerned with server uptime, whilst developers are concerned with release frequency.

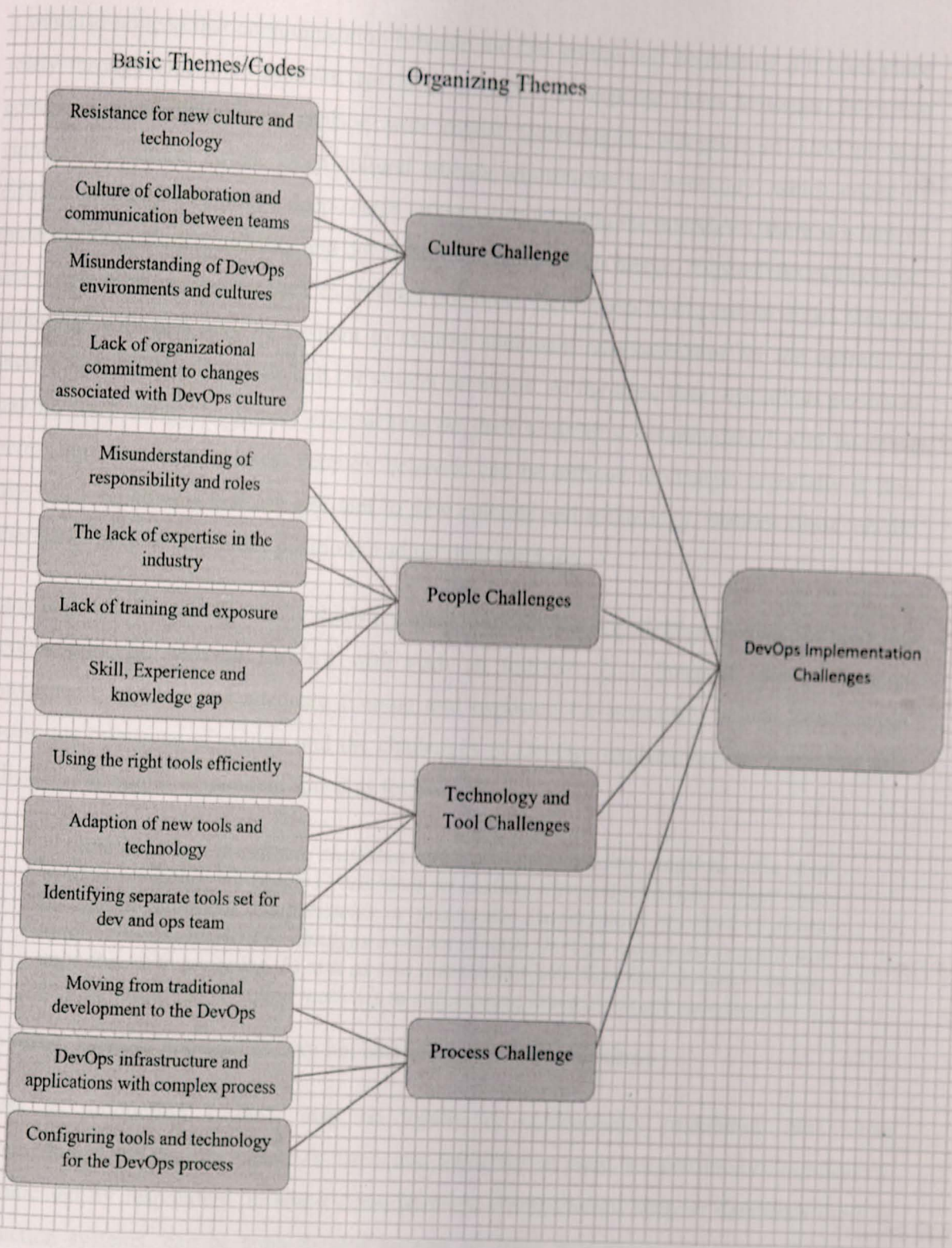

Figure 11: DevOps implementation challenges related to issues representation

It was also noticed that communication between the two groups may be missing if it takes place solely through electronic systems, resulting in delays in response times to issues. In-person conversation is difficult to replace with digital technologies. Adoption of DevOps also emphasizes cultural issues. Deep cultural attitude adjustments are necessary, which might be difficult given the company's deep-seated culture. According to the interviews, there is a lack of understanding of responsibility and duties between teams, roles merge, responsibilities fluctuate, and people must reconsider their established positions. Developers must perform duties they are unfamiliar with and may be hesitant to assume additional responsibilities for the operational environment, such as being on call for system problems. At the same time, operations personnel may be wary about developers encroaching on their territory or overburdened with handling increasingly frequent releases. People's behavior can be difficult to change, especially if they have had long careers and are resistant to new cultures and technology. Smaller firms, on the other hand, may be in a better position to adjust their methods.

Many interviewees said that individuals are open to modifying organizational culture to meet the DevOps concept. Moving forward with such an initiative is a fantastic idea if there is management backing and adequate training and exposure for personnel. Senior interviewees discuss the relevance of culture and the challenges of DevOps implementation. Regarding this difficulty, one senior respondent stated as follow.

> *"I think it's a major cultural shift that we're also experiencing, tough to solve, and what is probably the biggest roadblock in moving forward with this, bigger than the actual technical competencies or processes. Furthermore, the lack of firms' long-term commitment to changes associated with DevOps culture is a significant impediment to progress. I see it as two established jobs that must suddenly merge, and we simply need to find methods to collaborate toward one common goal."*

As a result, DevOps approaches may not be appropriate in all situations.

Access to production systems may be legally or contractually prohibited, thus when implementing DevOps, the industry's limits and feasibility in many sectors must be considered. As indicated by responders, in some circumstances, environments, such as databases used in production systems, can be sophisticated enough to make reproducing the environments for verification and testing problematic. As a result, automated testing becomes less trustworthy,

making diverse settings difficult for successful DevOps adoption. While there is agreement on some aspects of DevOps, its fundamental core remains somewhat ambiguous. Because there is no standard set of fixed DevOps processes, practitioners struggle to determine which techniques to employ for DevOps deployment. Access to production systems can be legally or contractually restricted. It has been observed that the meaning of DevOps has altered in recent years, and new DevOps technologies continue to emerge, therefore one of the issues is that DevOps is vague yet evolving.

4.5. Discussions and Analysis

The data collected through a questionnaire and an interview revealed that implementing DevOps had several advantages. Interviewees also emphasized the advantages of implementing DevOps. The majority of respondents saw DevOps as a way to increase the number of features implemented and deliver more releases. Rapid delivery is consistent with the DevOps concept of decreasing the time it takes for a software release to reach the production environment (Bass et al., 2015). DevOps promotes automation, which is thought to improve release quality. DevOps aids in the communication of developers and operations engineers. This encourages collaboration in order to improve the development process and ultimate result. Furthermore, the existing diverse abilities can be simply applied, boosting the team's reactivity to changing circumstances.

Interviewees confirmed the strong agreement of questionnaire replies that DevOps approaches allow real-time monitoring, which helps to foster fast feedback loops and an experimental culture that encourages more contact with end users. According to certain literature, such as Balalaie et al. (2016) and Cukier (2013), developers have previously highlighted real-time monitoring as a factor that helps to construct fault-aware systems, and DevOps has been observed to support an experimental culture (Neely and Stolt, 2013). Some of the issues we noticed included communication gaps and the current business culture.

A lack of knowledge and information exchange can cause critical truths to be obscured. Although guidelines for information exchange may be beneficial, changing a company's culture can be difficult. It is possible that the size of the organization or company-wide support for the change will be essential. Smaller businesses are better positioned to react quickly to changes.

Environmental constraints can be challenging at times. Working habits that prioritize security, for example, can make it difficult for some businesses to implement DevOps methods. The difficulties are exacerbated by difficult-to-recreate technical environments.

As DevOps evolves, its definition, procedures, and tools are expected to alter, posing a long-term problem. As organizations continue to adapt to change, Bass et al. (2015) advise that DevOps should not be attached to any specific must-have tools or communication techniques, but should instead be aligned with the higher-level goals that an organization desires to achieve. As the survey results reveal, there are no significant claims for causation and no challenges to internal validity. The replies of the interviewees are their personal views on DevOps. As a result, the described advantages and disadvantages are not universal. Respondents may have misunderstood DevOps due to a lack of a clear definition. Factors influencing the benefits and problems may have been overlooked if a responder had a limited understanding of DevOps, which may have influenced their view of its consequences.

4.6. Proposed Guideline for Successful DevOps Implementation

This section proposed a guideline to address DevOps implementation challenges in the software development industry as a result of this research. In response to the third research question, a guideline is proposed based on the study's empirical data findings and relevant literature. The quantitative and qualitative research findings suggest that the DevOps implementation challenges be reconsidered and included in the guidelines for future improvement. As a result, the proposed guideline attempted to categorize the aforementioned findings into three main activities: current DevOps implementation challenges, detailed solutions to current challenges, and future perspectives on ways to improve.

Through a mindset and culture change, development and IT operations are combined for quality assurance in a company. Thus, DevOps is first and foremost a culture and mindset that fosters strong collaborative bonds between software development and infrastructure operations teams. The DevOps culture is based on constant collaboration and communication, gradual change, shared end-to-end responsibility, and early problem-solving. However, software companies face numerous challenges as a result of DevOps implementations. Companies, however, have

different approaches to implementing a DevOps roadmap to address challenges in their practices due to different models in different countries. Some are attempting to tailor the DevOps implementation practice to their specific environment and working culture.

These differences emerge as a result of environmental and socio-cultural approaches to new technologies, as well as client companies' integration methods. Some businesses take a client-oriented approach to delivery, while others take a benefit-oriented approach. This implies that the benefits of DevOps for different companies and stakeholders are dependent on their specific DevOps implementation strategies, but there are still limitations on recommended common best practices. According to Lwakatare, there is still a lack of evidence of recommended common best practices for successful DevOps implementation (Lwakatare, 2017).

In general, DevOps requires a delivery cycle that comprises planning, development, testing, deployment, release, and monitoring with active cooperation between different members of a team. The following are the adapting roadmap of DevOps implementation proposed by the researcher and companies can develop their specific strategy based on the challenges they faced for successful DevOps implementation. This study proposed a guideline from two perspectives: 1) from the standpoint of each team's responsibilities and roles; and 2) from the standpoint of all organizational tasks. The diagram of the guideline is shown below.

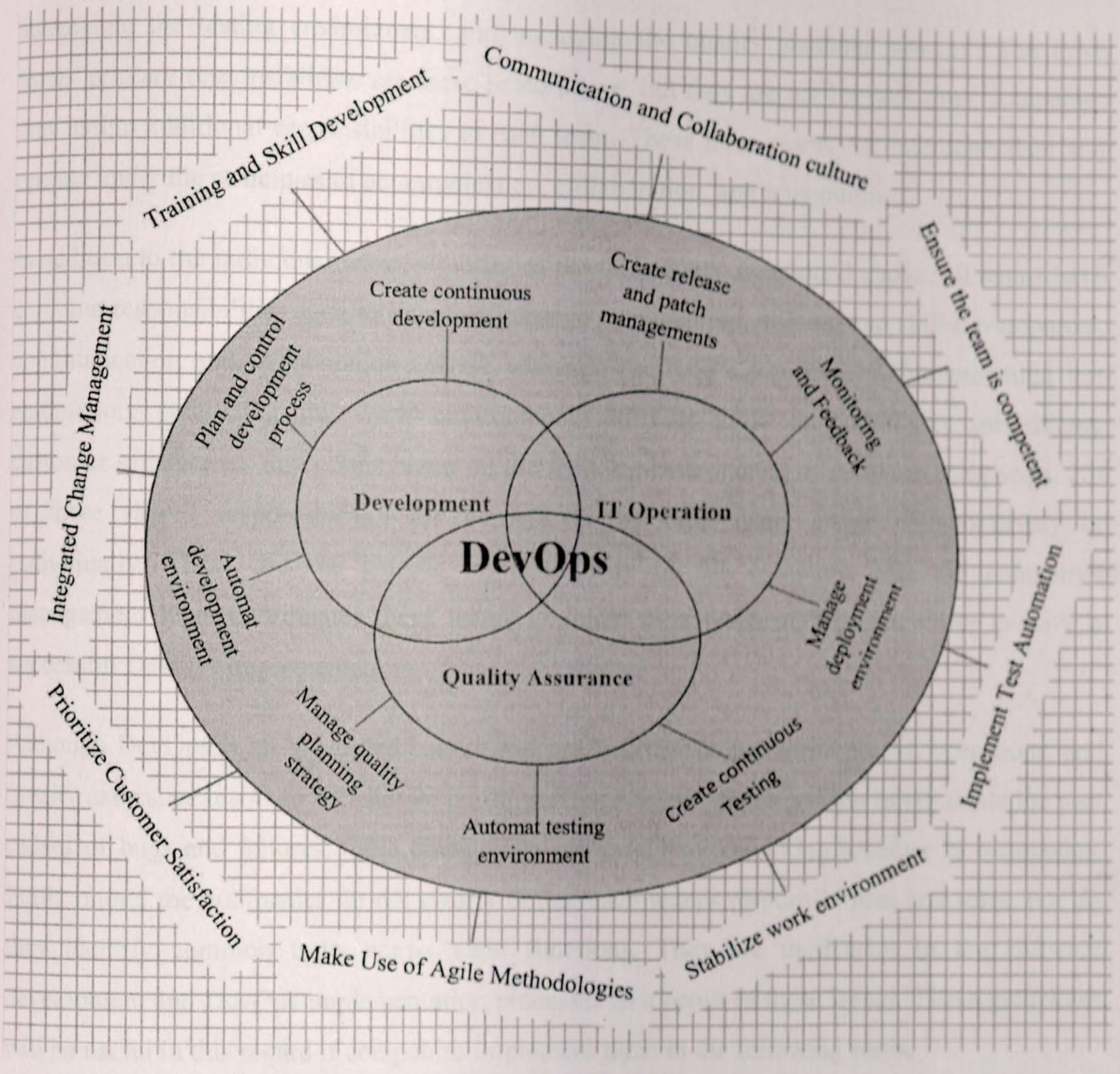

Figure 12: Proposed Guideline for Successful DevOps Implementation represented as a conceptual model (source: Own survey, 2022)

DevOps can be successfully accomplished with the collaboration of three key teams. They are as follows: Development, IT Operations, and Quality Assurance. These three teams have their own major responsibilities in order to achieve their common goals. As illustrated in Figure 12, the Development Team's responsibilities are to automate the development environment, plan and control the development process, and create continuous development. The IT Operations Team is responsible for creating release and patch management, monitoring and feedback, and managing the deployment environment. Quality assurance is responsible for creating continuous testing,

automating the testing environment, and managing the quality planning strategy. The three teams' primary responsibilities are those listed above, but they are not limited; thus, companies may assign additional responsibilities to each team. These three teams are interdependent and operate under the principles of communication, collaboration, and integration.

In addition to the individual responsibilities of the three teams mentioned above, all teams have common responsibilities such as integrated change management, training and skill development, communication and collaboration culture, ensuring the team's competency, implementing test automation, stabilizing the work environment, utilizing agile methodologies, prioritizing customer satisfaction, and others based on the working environment of software companies. All of these shared responsibilities are carried out by each team, either collaboratively or individually. These are the general responsibilities of the company, and the company's management team coordinates these teams to fulfill their responsibilities in order to have a successful DevOps implementation.

Although DevOps is an integrated culture and practice that unites development, operations, and quality assurance teams to collaborate on the software development process to eliminate the root causes of bugs and failures, these three teams are rarely seen working together in companies. Furthermore, the companies do not clearly define each team's responsibilities and roles, nor do they identify common tasks across team functions. They are ineffective in the software development and DevOps implementation processes as a result of these issues. These guidelines will be useful in this regard if companies implement them in the following ways.

- Determine each team's responsibilities and roles. As shown in Figure 12, the inner three circles represent each team, and the arrows around them represent each team's responsibilities and roles.
    - Developers have responsible for automating development environments, planning and controlling development processes, and establishing a continuous development process.
    - IT Operations is responsible for managing deployment environments, creating release and patch management, monitoring and feedback, as well as maintaining

and deploying the cutting-edge software and infrastructure that powers their products' technology.

- Quality assurance is responsible for taking a proactive role in the process of developing an automated testing environment, managing quality planning strategy, and developing continuous testing.

Technical innovation is not required in the DevOps culture. It is more about instilling appropriate values and principles in the organization. The DevOps methodology is founded on a set of fundamental business principles that promote transparency and a comprehensive development environment. As previously stated, out-of-circle activities represent organizational values and principles for the successful implementation of DevOps, and these are the primary activities performed by all teams as an organization.

CHAPTER FIVE

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Overview

This chapter presents the study's conclusions, as well as recommendations emphasizing on how DevOps implementation issues could be addressed based on the empirical data provided throughout the study, as well as suggestions for future research.

5.2. Conclusions

The purpose of this research is to identify DevOps implementation practices and challenges in the Ethiopian software development industry. The researcher believes that two of the research questions, "how is DevOps implementation being practiced in Ethiopian software development companies?" and "what are the challenges to implementing DevOps in Ethiopian software development companies?" were answered through descriptive and thematic analysis of the study.

DevOps is emerging as an appropriate term for a practical solution to addressing the critical integration challenges between development and operations capabilities in order to supplement today's agile approaches. However, some Ethiopian software firms have adopted DevOps. According to survey data analysis results, more than half of responding companies have partially implemented DevOps, and the majority of software companies have limited experience with DevOps implementation. However, this study focused on four dimensions of DevOps implementation in the context of the Ethiopian software development industry during DevOps implementation. It also identified the difficulties they encountered during the DevOps implementation.

According to the findings, while Ethiopian software companies are implementing DevOps, they are more focused on the practice of process dimension and people dimension than technology and tools. Respondents strongly agreed that continuous development, continuous testing, continuous deployment, continuous delivery, and feedback practices are more widely practiced on the process dimension. Ethiopian software firms encountered a number of obstacles that slowed the DevOps implementation process. This study identified challenges related to

transitioning from traditional development to a DevOps approach, building DevOps infrastructure and applications with complex processes, and configuring tools and technology for the DevOps process.

In terms of people dimension; they practiced team organization and continuous learning in their companies, and there is a limited practice on competent and capable employees and qualified employees with DevOps. However, a number of issues were identified in this study that slowed the DevOps implementation process in Ethiopian software companies from the people perspective. Misconceptions about responsibility and roles; a lack of professional expertise among industry employees; a lack of appropriate technical skills and training; and skill, experience, and knowledge gaps However, Ethiopian software companies have a limited capacity.

Although Ethiopian software development firms lack practical experience, cultural dimension practice is the primary driver of successful DevOps implementations. Communications and collaborations, cultural understanding practice, innovation drivers practice, cultural capability practice, and cross-team collaboration are all practices that fall under the cultural dimension. All companies are not focused on innovation drivers, cultural capability practice, and cross-team collaboration as a result of these practices. However, the main points for cultural practices in successful DevOps implementation are innovation drivers and cultural capability practice.

Furthermore, the researcher discovered that the study companies encountered various challenges during DevOps implementation in these culture-related practices. Employee resistance to new culture and technology, a culture of team collaboration and communication, a misunderstanding of DevOps environments and culture, and a lack of organizational commitment to changes associated with DevOps culture are the challenges. Furthermore, DevOps culture practices pose a greater challenge than technology and tools. Technology and tool challenges can be overcome, but cultural issues are difficult to identify and then resolve.

In terms of technology and tool dimensions, Ethiopian software companies are implementing DevOps. The majority of respondents agreed that they used these practices in their software companies. This dimension's practices include deployment automation, DevOps environment

configuration, issue tracking and version control tools, and software configuration management tools. However, challenges in technology and tool dimension practices have been identified in this study.

The challenges in technology and tool dimension are using DevOps technology and tools, efficiently using the right tools, adapting to new tools and technology, and identifying separate tool sets for the dev and ops teams. However, selecting the appropriate tools is a significant challenge. In the DevOps process, there are far too many tools, each of which serves a different purpose in different situations. It is extremely difficult to find the right tool for the right situation. Furthermore, implementing DevOps with legacy software is extremely difficult. Because DevOps is a new technology, Legacy Software may lack functionality that can be applied to the DevOps process.

Finally, this research contributed to the practice and challenges of implementing DevOps in the Ethiopian software development industry. Benefits of DevOps implementation are perceived by software practitioners in terms of improved software delivery speed, system quality, collaboration, communication, and productivity among stakeholders in software development and operations teams. However, it was discovered that the practices are more than just the result of DevOps implementation and include supporting practices and principles such as agile software development methodology, code review, quality assurance test automation, and so on. Since challenges such as transitioning from a traditional development approach to a DevOps approach, resistance to new culture and technology, lack of organizational commitment to DevOps culture changes, lack of skill, lack of experience, knowledge gap, and adaptation to new tools and technology were still observed in the findings of this study.

5.3. Recommendations for Practice

This recommendation is based on the results of this research analysis. It is provided as part of a research project on the practices and challenges of DevOps implementation in the Ethiopian software development industry. The study's findings are intended to improve Ethiopian software development companies' experience with DevOps implementation and to guide best practices in the improvement of their DevOps implementation. As well as assisting in the elimination of

DevOps challenges and the realization of the intended DevOps implementation benefits. As a result, the researcher has forwarded the following recommendations.

- **People Dimension**: Ethiopian software development firms must identify and train teams with automation or capable technical skills. When there is a shortage of professional experience manpower, management must recruit employees with the necessary skills as an action item when implementing DevOps.

- **Culture Dimension**: Another important practice is culture, which influences how teams collaborate and share responsibility for their application's end users. It is critical to begin a mutual exchange with Dev and Ops in order to break down the wall between the teams, and both teams will mutually learn from each other. This will be determined by the working culture of the employees, as well as their society and country.

- **Technology and Tools Dimension:** Other practices in DevOps implementation include technology and tools. Technology and tools aid in DevOps implementation practices such as configuration management, DevOps process environment configuration, code and application testing, infrastructure as code authoring, and environment deployment. Companies must, however, exercise caution when selecting these tools and technologies because improper selection will expose DevOps implementation practices to unnecessary challenges. Choosing the right tools, for example, is a significant challenge. There are far too many tools in the DevOps implementation process, all of which serve different functions in different situations. You must find the right tool for the right situation when implementing DevOps.

- **Process Dimension**: One of the DevOps implementation practices is process dimension, which alters all of a company's software development processes. Software firms should improve their software development processes and ensure that DevOps approaches and practices are implemented throughout the organization. Software companies may claim to use DevOps processes, but their operations are still traditional. Furthermore, project managers should consider how to manage DevOps implementation practices such as

continuous development, continuous testing, continuous deployment, continuous integration, continuous delivery, and feedback.

- Increased frequency of quality deployments, improved testing quality, quality software delivery, and increased collaboration between development and operations teams are all advantages of DevOps implementation. Positive impact was identified for key relationships between realized benefits. Benefits such as high autonomy, motivating collaboration, and feeling valued, for example, were discovered to positively contribute to improved team morale and engagement, whereas benefits such as improved code quality, natural communication, and knowledge sharing were discovered to positively contribute to improved deployment frequency.

5.4. Future Research Recommendation

DevOps implementation has frequently faced challenges in various situations because it is a combination of two processes such as development and IT operations. The most difficult challenge in the DevOps implementation process is cultural. The practices and challenges in this study are classified into four categories. In the future, more research will be needed in each dimension to establish best practice guidelines for successful DevOps implementation.

Furthermore, this study was limited to ten companies, but the author believes it is critical to conduct additional research, preferably with a large number of software companies, to identify best practices in the successful implementation of DevOps in order to assist the Ethiopian software development industry.

Finally, quantitative and qualitative studies are used in this study to identify DevOps implementation practices and challenges. Using continuous evaluation methods and feedback mechanisms, piloting longitudinal research helps to improve the quality, efficiency, usability, and content of proposed guidelines.

REFERENCES

A. Katal, V. Bajoria, and S. Dahiya. (2019). DevOps: Bridging the gap between development and operations. in Proc. 3rd Int. Conf. Comput. Methodolog. Commun. (ICCMC), Mar, pp. 1–7.

Airaj, Mohammed. (2016). Enable cloud DevOps approach for industry and higher education. Concurrency and Computation: Practice and Experience.

Balalaie, A., et al. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. IEEE Software 33(3), 42–52.

Bass, L., Weber, I., Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley.

Chen L. (2017). Continuous Delivery: Overcoming Adoption Challenges. Journal of Systems and Software. 128: 72–86.

Cito J, Leitner P, Fritz T & Gall HC. (2015). The Making of Cloud Applications: An Empirical Study on Software Development for the Cloud. Proc. 10th Joint Meeting on Foundations of Software Engineering. New York: ACM Press: 393–403.

Creswell, J. W., et al. (2003). Advanced mixed methods research designs. In A. Tashakkori & C. Teddlie (Eds.).

Creswell, J.W. (2012). Educational research: Planning, conducting, and evaluating quantitative and qualitative research. (4th Ed.). Boston, MA: Pearson.

Creswell, John W. (2014). Research Design: Qualitative, Quantitative, and Mixed Methods – in research (25-26).

Cukier, D. (2013). DevOps patterns to scale web applications using cloud services. In: Proc. of the 2013 Companion Publication for Conference on Systems, Programming, & Applications: Software for Humanity. pp. 143–152. SPLASH '13, ACM, New York, NY, USA (2013)

Debois, P. (2008). 'Agile infrastructure and operations: How infra-gile are you?', Agile 2008 Conference pp. 202–207.

Elberzhager, Frank & Arif, Taslim & Naab, Matthias & Süß, Inge & Koban, Sener. (2017). From Agile Development to DevOps: Going Towards Faster Releases at High Quality – Experiences from an Industrial Context. 33-44.

Erich, Floris & Amrit, Chintan & Daneva, Maya. (2014). Report: DevOps Literature Review.

F.M.A. Erich, C. Amrit, and M. Daneva. (2017). A Qualitative Study of DevOps Usage in Practice. Journal of Software: Evolution and Process. 29(6). e1885.

Ferreira Leite, Leonardo & Rocha, Carla & Kon, Fabio & Milojicic, Dejan & Meirelles, Paulo. (2019). A Survey of DevOps Concepts and Challenges.

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software. 123. 176–189.

Freeman, E. (2019). DevOps for Dummies (For Dummies (Computer/Tech)). For Dummies.

Golafshani, N. (2003). Understanding Reliability and Validity in Qualitative Research. The Qualitative Report, 8(4), 597-606.

Gunnar Menzel. (2015). DevOps - The Future of Application Lifecycle Automation.

Hamunen, Joonas. (2016). Challenges in adopting a DevOps approach to software development and operations.

Hart, Michael & Burke, John. (2020). An Exploratory Study on the DevOps IT Alignment Model. Interdisciplinary Journal of Information, Knowledge, and Management. 15. 127-154.

Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2019). From Agile to DevOps: Smart Skills and Collaborations. Information Systems Frontiers. 22(4). 927–945.

Humble, J. & Molesky, J. (2011). 'Why enterprises must adopt DevOps to enable continuous delivery.

Hüttermann M. (2012). Introducing DevOps. In: DevOps for Developers. Apress, Berkeley, CA.

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps? Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops.

Jez Humble and David Farley. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (1st. ed.). Addison-Wesley Professional.

Johannesson, P. and Perjons, E. (2014). An Introduction to Design Science. Springer International Publishing.

Jones S, Noppen J & Lettice F. (2016). Management Challenges for DevOps Adoption within UK SMEs. Proc. 2nd International Workshop on Quality-Aware DevOps. New York: ACM Press: 7–11.

K. Nybom, J. Smeds, and I. Porres. (2016). On the impact of mixing responsibilities between devs and ops. in International Conference on Agile Software Development.

Khan, Muhammad & Jumani, Awais & Mahar, Farhan & Siddique, Waqas & Shaikh, Asad. (2020). Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud. Indian Journal of Science and Technology. 13. 552-575.

Kim, G., Humble, J., Debois, P. & Willis, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, IT Revolution.

L. Riungu-Kalliosaari, S. Mäkinen, L.E. Lwakatare, J. Tiihonen, and T. Männistö. (2016). "DevOps Adoption Benefits and Challenges in Practice: A Case Study", In International Conference on Product-Focused Software Process Improvement, Springer, 2016, pp. 590-597.

L.E. Lwakatare, T. Karvonen, T. Sauvola, P. Kuvaja, H.H. Olsson, J. Bosch, and M. Oivo. (2016). Towards DevOps in the Embedded Systems Domain: Why is it so Hard?. In 49th Hawaii International Conference on System Sciences. IEEE. pp. 5437-5446.

Lindgren, E., Münch, J. (2015). Agile Processes, in Software Engineering, and Extreme Programming. 16th Int. Conf., XP 2015, Helsinki, Finland, May 25-29, Proceedings, chap. Software Development as an Experiment System: A Qualitative Survey on the State of the Practice, pp. 117–128. Springer International Publishing.

Lwakatare, L.E. (2017). DevOps adoption and implementation in software development practice: concept, practices, benefits, and challenges.

Lwakatare, Lucy Ellen & Kilamo, Terhi & Karvonen, Teemu & Sauvola, Tanja & Heikkilä, Ville & Itkonen, Juha & Kuvaja, Pasi & Mikkonen, Tommi & Oivo, Markku & Lassenius, Casper. (2019). DevOps in Practice: A Multiple Case study of Five Companies. Information and Software Technology.

Mishra, Alok & Otaiwi, Ziadoon. (2020). DevOps and software quality: A systematic mapping. Computer Science Review.

Mohd. Asfar, Mr. Raj kumar goel, and Mr. pradeep kumar. (2017). An approach of improve efficiencies through DevOps Adoption. International Research Journal of Engineering and Technology

Nagarajan, Anitha & Overbeek, Sietse. (2018). A DevOps Implementation Framework for Large Agile-Based Financial Organizations. Confederated International Conferences. CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I.

Neely, S. and Stolt, S. (2013). Continuous delivery? Easy! Just change everything (well, maybe it is not that easy). In: Proceedings of the 2013 Agile Conference. pp. 121–128. AGILE '13, IEEE Computer Society, Washington, DC, USA (2013).

P. Perera, M. Bandara, and I. Perera. (2016). Evaluating the impact of DevOps practice in Sri Lankan software development organizations. 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer). 2016. pp. 281-287.

Perera, Pulasthi & Silva, Roshali & Perera, Indika. (2017). Improve software quality through practicing DevOps. 1-6.

Pooja Batra, A. Jatain. (2020). DevOps: Current Practices, Challenges, and Implication.

R. Macarthy and J. Bass. (2020). An Empirical Taxonomy of DevOps in Practice. 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). 2020. pp. 221-228.

Rajkumar, M., Pole, A. K., Adige, V. S., & Mahanta, P. (2016). DevOps culture and its impact on cloud delivery and software development. 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring).

Rowse, M., & Cohen, J. (2021). A Survey of DevOps in the South African Software Context. In Proceedings of the 54th Hawaii International Conference on System Sciences (p. 6785).

Saima Rafi, Muhammad Azeem Akbar, Abeer Abdulaziz AlSanad, Lulwah AlSuwaidan, Halah Abdulaziz AL-ALShaikh, Hatoon S AlSagri. (2020). Decision-Making Taxonomy of DevOps Success Factors Using Preference Ranking Organization Method of Enrichment Evaluation. Mathematical Problems in Engineering. vol. 2022. 15 pages. 2022.

Sanjeev Sharma. (2017). The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise

V. Gupta, P.K. Kapur, and D. Kumar. (2017). Modeling and Measuring Attributes Influencing DevOps Implementation in an Enterprise Using Structural Equation Modeling. Information and Software Technology. 92. pp. 75-91.

Violino, B. (2016). Real-world devops failures -- and how to avoid them. InfoWorld.

Virginia Braun & Victoria Clarke. (2006). Using thematic analysis in psychology, Qualitative Research in Psychology, 3:2, 77-101

W.P. Luz, G. Pinto, and R. Bonifácio. (2018). Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice. In 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1-10.

Wang, C., & Liu, C. (2018). Adopting DevOps in Agile: Challenges and Solutions.

Wiedemann, Anna & Forsgren, Nicole & Wiesche, Manuel & Gewald, Heiko & Krcmar, Helmut. (2019). Research for practice: The DevOps phenomenon. Communications of the ACM. 62. 44-49.

Wiedemann, Anna & Schulz, Thomas. (2017). Key Capabilities of DevOps Teams and their Influence on Software Process Innovation: A Resource-Based View.

Yarlagadda, Ravi Teja, (2021). an Exploratory Study of DevOps & Its Future in USA International Journal of Innovations in Engineering Research and Technology.

Yiran and L. Yilei. (2017). The challenges and mitigation strategies of using DevOps during software development. Tech. Rep.

APPENDICES

Appendix 1: Letter of Request

አዲስ አበባ ዩኒቨርሲቲ
የተፈጥሮ ሳይንስ ኮሌጅ
የኢንፎርሜሽን ሳይንስ ት/ቤት

Addis Ababa University
College of Natural Science
School of Information Science

Date: December 21, 2021
Ref No. SIS/25/2021/14

To Whom It May Concern

Subject:- Student Mesfin Tafese

Dear Sir /Madam,

Student Mesfin Tafese (ID.No GSE/6282/12) is a graduate student at the School of Information Science, Addis Ababa University. He is currently conducting his M.Sc. Thesis research under the title "DevOps Implementation Practices and Challenges in the Ethiopian Software Development Industry."

I would like to thank you in advance for all the assistance that you would provide to the student.

With Regards

Tibebe Beshah (PhD)
Head, School of Information Science

✉: 1176 Email: information_cci_cns@aau.edu.et ☎: +251-(11)-122-91-91

Appendix 2: Questionnaire Survey

ADDIS ABABA UNIVERSITY

SCHOOL OF INFORMATION SCIENCE

DEPARTMENT OF INFORMATION SYSTEM

Dear Sir or Madam:

In partial fulfillment of the requirements for the Degree of Master of Science in Information System, I am undertaking research on "DevOps implementation practices and challenges in the Ethiopian Software Development industry" at Addis Ababa University. I have accordingly prepared this Survey questionnaire. The objective of the survey is to investigate current practices and challenges of DevOps implementation in the Ethiopian software development context.

This research is believed to produce results that can improve the understanding of the successful implementation of DevOps in software development companies. Kindly your honest responses to each question and statement are extremely valuable to the outcome of this research. The questionnaire survey will be used for academic purposes only. As a result, all responses will be kept in strict confidentiality.

Your dedication is most valued and appreciated and I would like to take this opportunity to thank you in advance for your kind participation, honest, and on-time response to the Questionnaires.

Thank you again!

Mesfin Tafese

Demographic related Questionnaires (please use X mark on the choice you Select)

1. **Basic Information**
2. Which organization are you from? _____
3. What is your current level of education?

   ☐ Diploma

   ☐ Degree

   ☐ Master's degree

   ☐ Ph.D.

   ☐ Other: _____

4. How many years have you worked in the software development and/or IT operation field?

   ☐ Less than two year

   ☐ 2- 5 years

   ☐ 6 - 10 years

   ☐ More than 10 years

DevOps Implementation Practices and Challenge questions

2. DevOps Implementation Basics

1. Have you implemented DevOps in your software company?

   ☐ Fully implemented

   ☐ Partially implemented

2. How have you implemented DevOps at your company? **You may select more than one.**

☐ Dev and Ops paring to form one team

☐ Formulating DevOps team

☐ Employing infrastructure-as-code practice

☐ Early incorporation of Ops in Dev projects

☐ Increase competence of Ops in Dev teams especially in outsourced Operations

☐ Need to increase the level of automation in testing, documenting the procedure for configuring and monitoring services

☐ Through organization structural changes e.g. incorporation of feature teams

3. DevOps Implementation Practices

Please indicate your level of agreement or disagreement (where **1 = Strongly Agree, 2 = Agree, 3 = Neutral 4 = Disagree and 5 = Strongly Disagree**) for the following questionnaires.

Questionnaires Items		1	2	3	4	5
Technology Dimension						
1	Is your company using deployment automation properly?	☐	☐	☐	☐	☐
2	Is your company using the required DevOps environments?	☐	☐	☐	☐	☐
3	Is your company using an Issue tracking tool?	☐	☐	☐	☐	☐
4	Is your company using Software configuration management tools?	☐	☐	☐	☐	☐
5	Is there enough DevOps technology capability in your company?	☐	☐	☐	☐	☐

Process Dimension						
6	Is your company using a continuous development process	☐	☐	☐	☐	☐
7	Is your company using a continuous testing process?	☐	☐	☐	☐	☐
8	Is your company using continuous deployment and delivery processes?	☐	☐	☐	☐	☐
9	Is your company using continuous integration, monitoring, and feedback processes?	☐	☐	☐	☐	☐
People Dimension						
10	Is your company using team organization?	☐	☐	☐	☐	☐
11	Is there continuous learning practice in your company?	☐	☐	☐	☐	☐
12	Is your company using competent and capable employees?	☐	☐	☐	☐	☐
13	Does your company has qualified employees with DevOps?	☐	☐	☐	☐	☐
Culture Dimension						
14	Are there communications and collaborations between each team in your company?	☐	☐	☐	☐	☐
15	Is there enough requirement understanding in your company?	☐	☐	☐	☐	☐
16	Is there a cultural understanding in your company?	☐	☐	☐	☐	☐
17	Are there innovation drivers in your company?	☐	☐	☐	☐	☐
18	Is your company using cultural capability practice?	☐	☐	☐	☐	☐

4. DevOps Implementation Challenges

Please indicate your level of agreement or disagreement (where **1 = Strongly Agree, 2 = Agree, 3 = Neutral 4 = Disagree and 5 = Strongly Disagree**) for the following questionnaires.

Questionnaires items		1	2	3	4	5
1	Understanding the DevOps Concept is challenging in your company.	☐	☐	☐	☐	☐
2	Getting DevOps capable members into a team is a challenge in your company.	☐	☐	☐	☐	☐
3	Obtaining management support for DevOps practices is a challenge in your company.	☐	☐	☐	☐	☐
4	Increasing the complexity of the build process in DevOps is a challenge in your company.	☐	☐	☐	☐	☐
5	Achieving compatibility between DevOps and legacy systems is a challenge for you.	☐	☐	☐	☐	☐
6	Finding experienced professionals to support DevOps practice is a challenge in your company.	☐	☐	☐	☐	☐
7	Changing company culture to support DevOps implementation is a challenge for you.	☐	☐	☐	☐	☐
8	Using DevOps technology and tools is a challenge in your company.	☐	☐	☐	☐	☐
9	Access to production systems needed by DevOps is a challenge in your company.	☐	☐	☐	☐	☐
10	Adapting organizational processes to DevOps is a challenge for you.	☐	☐	☐	☐	☐
11	Identifying the responsibility of DevOps between the developers and IT operations is a challenge in your company.	☐	☐	☐	☐	☐

12	Communication and collaboration between the development team and IT operation team is a challenge in your company to implement DevOps.	☐	☐	☐	☐	☐
13	Adapting employees' mindsets to achieve successful DevOps is a challenge in your company.	☐	☐	☐	☐	☐

14. Any other challenges you faced during DevOps implementation at your company?

1. _____
2. _____
3. _____
4. _____

15. What do you think are the solutions to address those challenges?

1. _____
2. _____
3. _____
4. _____

16. What are the appropriate ways you have been followed to implement DevOps successfully at your organization?

1. _____
2. _____
3. _____
4. _____

Appendix 3: Interview Outline

Interview questions

Introduction

1. Could you please introduce yourself and describe your relationship with DevOps implementation at your organization?
2. What does your understanding of DevOps in your organization?
3. How did your organization implement DevOps?

DevOps Implementation Practices

4. What are the common DevOps practices you use at your organization?
5. How do the development and operations teams collaborate and implement the DevOps culture at your organization?
6. Name some best practices which should be followed for DevOps implementation success at your organization.

DevOps Implementation Challenges

7. What challenges existed when implementing DevOps at your organization? Could you list it out?
8. What are the common challenges you faced to implement DevOps?
9. What kind of measures did you take to solve those challenges when you implement DevOps at your organization?
10. Are there any appropriate ways you followed to implement DevOps successfully at your organization? Could you list it out?

Appendix 4: Descriptive statistics of the survey mean and standard deviation

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
1. Is your company using deployment automation properly?	107	1	5	1.99	.966
2. Is your company using the required DevOps environments?	107	1	4	2.07	.934
3. Is your company using an Issue tracking tool?	107	1	5	1.73	.957
4. Is your company using Software configuration management tools?	107	1	4	1.93	.832
5. Is there enough DevOps technology capability in your company?	107	1	4	2.24	.989
1. Is your company using a continuous development process?	107	1	5	1.64	.884
2. Is your company using a continuous testing process?	107	1	4	1.80	.906
3. Is your company using continuous deployment and delivery processes?	107	1	4	1.71	.869
4. Is your company using continuous integration, monitoring, and feedback processes?	107	1	4	1.93	.918
1. Is your company using team organization?	107	1	4	1.55	.676
2. Is there continuous learning practice in your company?	107	1	4	1.64	.756
3. Is your company using competent and capable employees?	107	1	4	1.70	.755
4. Does your company has qualified employees with DevOps?	107	1	5	2.34	.961
1. Are there communications and collaborations between each team in your company?	107	1	4	1.62	.654
2. Is there enough requirement understanding in your company?	107	1	4	1.80	.758
3. Is there a cultural understanding in your company?	107	1	4	1.90	.800
4. Are there innovation drivers in your company?	107	1	5	2.09	.917
5. Is your company using cultural capability practice?	107	1	5	2.25	.766
Understanding the DevOps Concept is challenging in your company.	107	1	5	3.11	1.261
Getting DevOps capable members into a team is a challenge in your company.	107	1	5	2.79	1.139

Obtaining management support for DevOps practices is a challenge in your company.	107	1	5	2.76	1.148
Increasing the complexity of the build process in DevOps is a challenge in your company.	107	1	5	2.79	1.071
Achieving compatibility between DevOps and legacy systems is a challenge for you.	107	1	5	2.73	1.069
Finding experienced professionals to support DevOps practice is a challenge in your company.	107	1	5	2.42	1.064
Changing company culture to support DevOps implementation is a challenge for you.	107	1	5	2.47	1.093
Using DevOps technology and tools is a challenge in your company.	107	1	5	2.99	1.068
Access to production systems needed by DevOps is a challenge in your company.	107	1	5	2.83	1.112
Adapting organizational processes to DevOps is a challenge for you.	107	1	5	2.92	1.167
Identifying the responsibility of DevOps between the developers and IT operations is a challenge in your company.	107	1	5	2.88	1.105
Communication and collaboration between the development team and IT operation team is a challenge in your company to implement DevOps.	107	1	5	2.90	1.124
Adapting employees' mindsets to achieve successful DevOps is a challenge in your company.	107	1	5	2.59	1.090
Valid N (listwise)	107				

Appendix 5: Theme generation in open code