UWS UNIVERSITY OF THE WEST *of* SCOTLAND

COMPARATIVE ANALYSIS OF DEVOPS TOOLS: GUIDING ORGANIZATIONAL SELECTION FOR OPTIMAL SOFTWARE DELIVERY ACROSS VARYING SCENARIOS.

Oluwapelumi Samuel Owolabi

B00947298

Thesis for the partial fulfillment of the requirements for the Master's Degree in Information Technology

University of the West of Scotland School of Computing, Engineering and Physical Science

19/04/2024

UWS UNIVERSITY OF THE WEST *of* SCOTLAND

DECLARATION OF ORIGINALITY

I proudly declare that this MSc project is a unique and original study, a product of my diligent work, and has not been submitted for any other course or degree.

Signature ______________________________

UWS UNIVERSITY OF THE WEST *of* SCOTLAND

Library Form to Accompany MSc Project

To be completed in full

Surname: **Owolabi**	
First Name: Oluwapelumi	**Initials: OS**
Banner ID: **B00947298**	
Course Code: **POSTGIT**	
Project Supervisor: **Miriam Birch**	
Project Title: **Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios.**	
Session: 2023/2024	**Date of Submission: 19/04/2024**
Signature:	

DEDICATION & ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who made this thesis possible. My appreciation to God for life to see me through this project from the beginning to the end of the project. All things worked together for my good.

I extend my sincere appreciation to my supervisor, and moderator, Miriam Birch, and Santiago Matalonga, whose expertise, understanding, and patience added considerably to my graduate experience. Your guidance was invaluable in formulating the research content and methodology for this thesis. I also thank Dr Daune West for competently co-ordinating this master's project in an orderly manner. She's well appreciated for lending her expertise to ensure that this project ran smoothly. I also express my gratitude to Dr. Costas Iliopulos, the program co-ordinator for successfully overseeing my degree program as a whole. Thank you for holding it all together.

I am particularly grateful for the assistance given by the technical staff in the CEPS at the University of West Scotland. Their help was a milestone in the completion of this project. I also wish to acknowledge the support provided by my peers and colleagues; their insights and feedback were crucial in refining my analysis.

A special thanks to the professionals and experts in the field of DevOps who participated in the interviews and provided critical real-world insights that greatly enriched my research. Without their enthusiastic participation, this study would not have been feasible.

I must express my very profound gratitude to my family for providing me with unfailing mental and emotional support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

TABLE OF CONTENT

TABLE OF FIGURES AND TABLES

ABSTRACT

As the world of software development continues to evolve rapidly, so does the need for DevOps practices to ensure productivity and innovation (Gonçalves et al., 2020). This project, titled "Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios," seeks to unearth some of these tools while providing a lucid framework that organizations can use when choosing their best-fit tools. This study aims to close the knowledge gap on DevOps tools by deeply analyzing literature and evidence.

The study combines a qualitative approach, using interviews to gain insights from industry experts, with literature-based secondary research to explore the practical advantages, guiding principles, and implementation challenges of DevOps tools. Ethical considerations such as informed consent and data confidentiality will be considered during this study.

Among its outcomes are establishing an all-around system that will assist in utilizing various resources to improve software delivery processes through DevOps. Such a structure will enable a wider adoption of strategic DevOps across different organizational settings, thereby contributing more knowledge on effective practice within this area.

This research seeks to enhance practical implementation and deepen academic knowledge on selecting and using DevOps tools.

Keywords: *DevOps, Comparative Analysis, qualitative research, scalability, usability, security, framework, BPMN, Agile, Tools, framework, organization, efficiency, cost-effectiveness, thematic analysis*

To view framework diagram raw files, please visit -
https://github.com/PelumiOwolabi/UWS-Project

CHAPTER 1: RESEARCH BACKGROUND

1.1 Introduction

Today's software development and delivery environment has seen DevOps become a transformative approach where development (Dev) teams work closely with operations (Ops) personnel thus fostering collaboration, efficiency, as well overall agility throughout software lifecycle management (Hutterman, 2012; Lwakatare et al., 2016). This shift bridges gaps between those responsible for creating programs on one hand and individuals charged with deploying them or keeping systems running on another hand (Bass, Weber & Zhu 2015).

Adoption of integration techniques associated with continuous deployment such as those explained by Puppet Labs' State Of Devops Report 2015 have been found to enhance performance indicators like frequency change-lead time; MTTR etc where organizations can meet market needs faster reliably. However achieving these gains depends much upon proper selection and utilization of appropriate Development Operation Tools i.e systems which control versions among others that enable efficiency work while promoting collaboration spirit among different parties involved in software production process (Aljundi, 2018).

Although it is widely agreed upon that these tools are essential, the dynamic nature of DevOps presents challenges for organizations trying to choose; they must select tools that can not only serve current operational requirements but also meet future strategic goals given diverse organizational settings within which such entities operate (Azad & Hyrynsalmi 2023). The wide array of capabilities strengths and weaknesses possessed by each tool available under this category makes it difficult to decide which among them is most suitable.

This study aims to help organizations choose the best software to deliver in different situations by analyzing DevOps tools comparatively.

1.2 Problem statement

Despite recognizing DevOps practices as a way to improve software delivery, there needs to be more explicit guidance on choosing DevOps tools. This absence of a structured approach leads to businesses making inadequate tool selections, which can result in wasted time and resources and a failure to fully leverage the benefits of DevOps or DevOps implementation (Erich, Amrit, & Daneva, 2017; Senapathi, Buchan, & Osman, 2019).

This study aims to resolve the problem. The aim and objectives describe how this problem will be solved and improved.

1.3 Aim

The main aim of this research is to develop a framework that will assist organizations in choosing DevOps tools that optimize software delivery across different situations and increase operational effectiveness and creativity in software development processes. The objectives below are the guidelines for what should be covered in the body of this project and actions taken so that all these goals may be met.

1.4 Objectives

1. To understand the DevOps Landscape by conducting a literature review.
2. To valuate DevOps tools by conducting interviews and a comparative analysis of the tools.
3. To develop a selection framework using BPMN 2.0, a software used to design frameworks.
4. To validate the Framework by ensuring it matches project objectives.
5. To contribute to DevOps knowledge and practice

1.5 Research Question

The following research question guides the study:

What are the valuable features of DevOps tools that need to be given priority by organizations depending on various scenarios and how can these features help in selecting appropriate tools for specific organizational needs and contexts?

Other areas that helped answer this research question are:

1. What criteria are essential for selecting DevOps tools in varying organizational contexts?
2. How do specific DevOps tools align with the identified selection criteria, and how does this affect organizational practices and software delivery outcomes?
3. What framework can be developed to guide organizations in choosing suitable DevOps tools for maximum software delivery?

1.6 Justification

This project seeks to compare different DevOps tools and provide guidance on how different organisations can best select them to ensure optimal software delivery in diverse situational contexts.

The rationale behind this is rooted in recognising that the landscape of DevOps tools is rapidly evolving, presenting organisations with a daunting challenge.

While it is established that DevOps practices are needed for effective software delivery within companies, there is a lack of literature discussing the selection of the tools being used. This includes factors such as aligning company practices with tools, the cost, whether they can afford them, and whether the revenue generated will be more than the finances used to maintain the tools.

This research aims to bridge the information gap by combining primary data obtained from industry professionals via interviews and reviewing the tools by conducting a comparison. This will help reveal the rationale behind using them and develop a selection framework that offers a pragmatic tool for organizations, enhancing their ability to navigate the complex DevOps tool landscape effectively.

1.7 Structure of the Dissertation

Chapter 1 briefly summarizes what the project is about. Chapter 2 thoroughly reviews existing literature, delving into the history of DevOps, its principles, and current approaches to selecting DevOps tools. Chapter 3 explains the research methods used in the study, including data collection and analysis techniques.

Chapter four presents findings from the qualitative analysis based on common themes from selecting DevOps tools while considering various environments. Chapter five discusses results obtained through investigation vis-a-vis previous studies, indicating their relevance to organizational practice and future inquiry. Therefore, it summarises major discoveries alongside professional recommendations within this industry, plus gives alternative suggestions regarding further exploration and critical evaluation concerning this research. It also provides a selection framework across environments.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Technology has evolved into an indispensable component of modern business. Given global tech advancement and the demand for comfort in day-to-day activities, enterprises are faced with an endless need to simplify business processes, foster employee performance, and enhance customer experience. This is primarily possible through software solutions. However, sadly, only a few companies have successfully implemented software for problem-solving (Kristinsson, 2015; Pueraniemi, 2014).

Consequently, DevOps has been developed as a solution-oriented approach to facilitating effective software production and implementation to help businesses and groups achieve organisational goals and solve problem efficiently.

Against this backdrop, this chapter presents findings and discussions of academic reports, research, and interviews on DevOps tools and methodologies applied in software delivery.

To determine these, the following research questions were used:

i. How is DevOps explained in literature?

ii. What principles and practices constitute DevOps?

iii. What benefits can be attributed to the implementation of DevOps tools among different organizations?

2.2 Overview of Relevant Literature

This review aims to identify relevant studies carried out on DevOps tools and related research areas. Using the research questions above as a guide, the review focused on papers published within the past decade (2010 to 2023). After filtering available publications based on relevance and recency, the review identified 32 previous publications: 28 primary studies and 4 secondary reports.

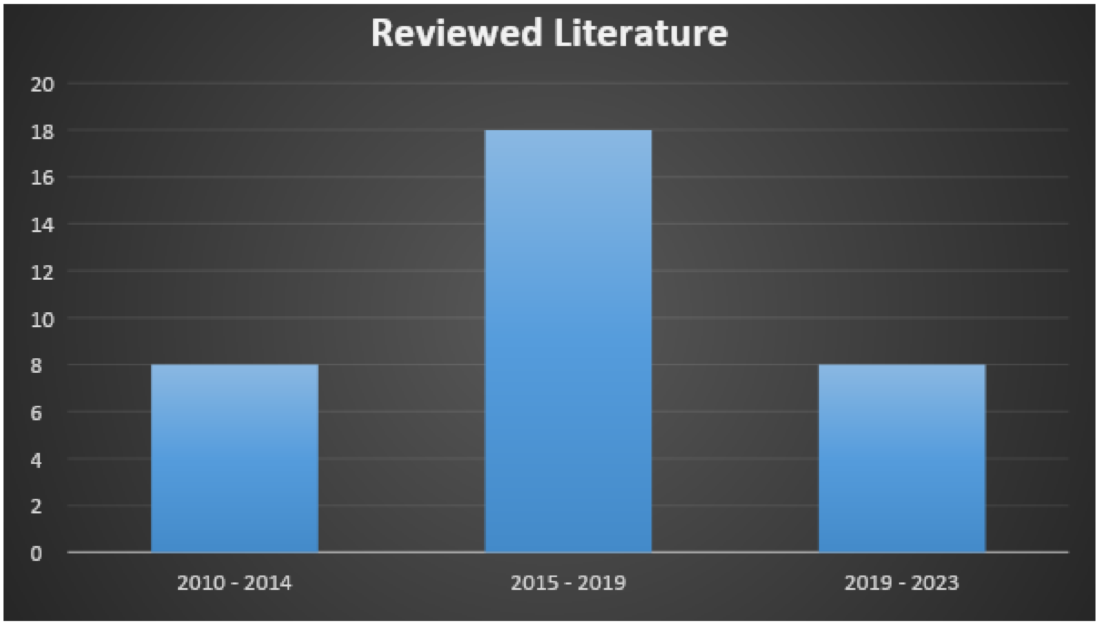


Figure 1: Overview of Relevant Literature

Information gathered from 19 (representing ~60%) the previous studies comprised interviews from individuals who had direct interactions with DevOps tools (developers, operations teams, DevOps experts, and employees in organisations where DevOps was adopted). In comparison, experimental studies comprised an estimate of ~40% of the papers reviewed.

Findings from this review established that organisations using DevOps for software delivery experienced notable improvements in several key areas, including client satisfaction, improved employee productivity, and a significant boost in overall organisational performance. DevOps was also found to boost quality assurance and software product delivery.

Following the gaps established in this review, this study will conduct a qualitative survey using semi-structured interviews. The sampled population consists of experts in the software development industry. All selected interviewees are professionals in DevOps practice with experiences ranging from 3 to 15 years. Interviewee responses have been collected and transcribed using Turboscribe and analysed using Thematic Analysis.

2.3 Understanding DevOps Tools

The term 'DevOps' represents the interplay between development and operations (Freeman, 2015). Beyond this, a literature review needs to establish more agreement among scholars, researchers, and industry experts on the meaning of DevOps. Like other buzzwords across the software development industry, everyone uses the term' DevOps,' but only some know what it is precisely (Freeman, 2015) or use it correctly (Kristinsson, 2015).

From a conceptual standpoint, DevOps has been applied as an approach to problem-solving in software delivery and, therefore, is best explained from the same perspective. The nature of DevOps makes it difficult to pin it to a definite definition. More so, given that DevOps has been applied to different contexts among various groups and companies to meet several different needs, the definition of DevOps is an open-ended question subject to the interpretation of the user (Huttermann, 2012).

However, a review of selected literature reveals that some scholars agree on what constitutes the basis of DevOps. Akshaya et al. (2015) present DevOps as a practice that requires the collaboration of 'operations and development' across the stages of a DevOps life cycle. Bass, Weber, and Zhu (2015), quoted in Bass, Champlin-Scharff, and Zhu (2016), explain DevOps as "a set of practices intended to reduce the time between committing a change being placed into normal production, while ensuring high quality" (p. 33).

Soni (2016) maintains that DevOps transcends the technical components involved in development but is, in fact, also a 'concept, culture, development and operational philosophy' (p 9). Similarly, Freeman (2019) defines DevOps as 'an engineering culture of collaboration, ownership, and learning with the purpose of accelerating the software development circle' (p. 7). This implies that DevOps is a system of multiple constituents combining human input (developers) with operations, including tools, processes, and methods.

Kristinsson (2015) echoes this thought as follows:

> “If DevOps was human I would say that the DevOps methods are the body and the limbs, while the humans putting the methods into action are the brain, the hearts and the veins going through the body.”

Therefore, for the purpose of this review, it is established that DevOps is a practice with multiple components – humans, philosophies, practices, methods, tools and processes – working together from development to production of software.

2.3.1 Origin of DevOps

Patrick Debois first used the term 'DevOps' in 2009 at a DevOps Days Conference in Belgium (Hutterman, 2012). As a practice, DevOps developed as a response to the challenges faced by extensive web services in deploying immediate service updates (Crowley et al., 2018). Its origin is often traced to Agile, but evidence from the literature suggests that the origin of DevOps predates the Agile revolution of 2021 (Freeman, 2019; Lwakatare et al., 2016).

Pugh (2010), as quoted by Kristinsson (2015), reports that DevOps is an offshoot of Agile, but the Agile movement in itself was developed from Lean thinking (p 3). Lean thinking is a practice modelled on the philosophy of waste reduction. In essence, unnecessary parts of a process should be removed to make the process less cumbersome and more effective. This philosophy, which had been in practice for over a decade, was adopted into software development "as a complementary methodology incorporated … to control a software development cycle" (Kristinsson, 2015. p 3).

Lean Think formed the foundation for the Agile movement (Shore & Warden, 2008, pp. 9 -10), which developed when a group of I.T. professionals partnered to eliminate barriers between operations and development. The result became the agile manifesto, which incorporated four values and twelve principles focused on developing 'people-centric software' by placing people over software, individuals over processes and tools, and customer collaboration over negotiation (Shore & Warden, 2008. p 8).

Lwaktare et al. (2016) demonstrate the relationship between DevOps and Agile, as well as DevOps and Lean. Their findings established that Lean Think principles inform the development and implementation of DevOps further than Agile. More so, Lean's emphasis on systems thinking promotes connectedness between developers and operations. On the other hand, Agile software principles are compulsory pillars for the success of DevOps, and they both thrive on collaboration, a fundamental value needed to enhance operations (pp. 10 – 14).

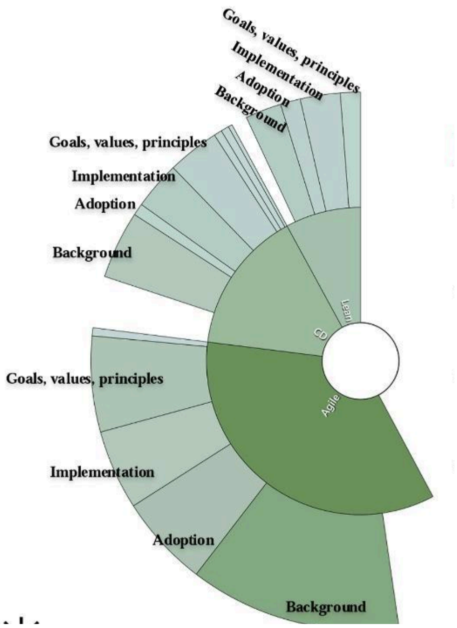


Figure 2: DevOps and Lean and Agile (Lwaktare et al. (2016)

Evidence shows that the Lean Think and Agile movements have heavily influenced DevOps, as practised today. Since 2009, when both Development and Operations were first annexed, DevOps has adopted end-to-end processes from Lean Thinking, shorter software delivery life cycles, and overall changes in infrastructure and systems from the Agile Movement (Deming, 1982; Highsmith & Cockburn, 2001; Sharma & Coyne, 2015).

Exploring Waterfall and Agile Models as Precursors of DevOps

Applications of the Waterfall and Agile models confirm that both are effective software development models despite the differences (Bob, 2021). However, the growing need for more efficiency and ease of software delivery necessitated the adoption of more modern, productive, and efficient software development models. In a comparative study of the Waterfall and Agile models, Balaji and Murugaiyan (2012) established the pros and cons of adopting either model and the major differentiating factors and requirements for use.

The waterfall model is "a sequential design process often used in software development processes, in which progress is seen as flowing downwards through the phases" (Balaji and

Murugaiyan, 2012. p. 1). Being the earlier of the two, the waterfall model operates on a sequence of stages in which an earlier stage's output becomes the next stage's output. This model emphasises the associative relationship and interconnectedness within the software development life cycle stages. This means that the requirement for the cycle's next phase should be clearly stated before commencement (Huttermann, 2012). While the downward flow of the progress maintains the general quality of the development, ending a phase before starting another makes it difficult to effect significant changes to previous phases of the cycle.

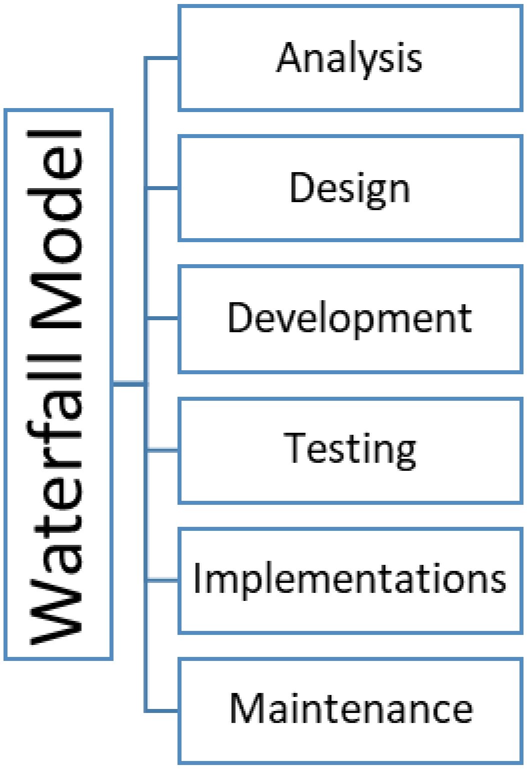


Figure 3 Waterfall Model (Balaji and Murugaiyan, 2012)

The waterfall model progresses linearly (Freeman, 2019, p. 80). This feature makes it easier for developers to follow progress and monitor software development. Waterfall also supports lean resources, making it easier for organisations or DevOps teams to commit only needed resources to the development process while limiting waste. Adetokunbo (2013) maintains that the waterfall model is most useful in structured systems where revisiting earlier stages for alterations is prohibited and unnecessary (p. 429).

Despite its benefits, DevOps teams reported challenges with the waterfall model. Balaji and Murugaiyan (2012) argue that the linear progression of the waterfall model is as much a benefit as an inhibition. In application, testers often detect errors with an earlier phase much later in the development cycle, and it becomes difficult to rework the phase, often resulting in a 'badly structured system' (p. 27). Since the processes and data in the Waterfall Model are separated, attempting to rework a phase will alter the entire code, making upgrades expensive and cumbersome (Adetokunbo, 2013. p. 430).

Setbacks from the Waterfall Model necessitated the introduction of Agile. Around the 1990s, organisations needed a flexible, feature-driven framework that could align with the changing atmosphere of the software ecosystem (Crowley et al., 2018). In 2001, 17 software engineers met to develop what would evolve into the principles of the Agile Model. As an improvement on the preceding Waterfall Model, the core focus of Agile was on continuous software delivery, simplified features, collaborative self-organised teams, and dynamism, among others. According to Crowley et al. (2018), the practices in the Agile Model have influenced and formed the foundations of DevOps processes (p. 30).

At the centre of the Agile Model is a swift and continuous activity, especially as a response to a novel demand. Balaji and Murugaiyan (2012) describe the Agile model as a model with "an adaptive team which is able to respond to the changing requirements." Through continuous integration/continuous delivery (CI/CD), Agile is essential for delivering effective and updated software solutions to meet new challenges and ensure a smooth user experience (p. 28).

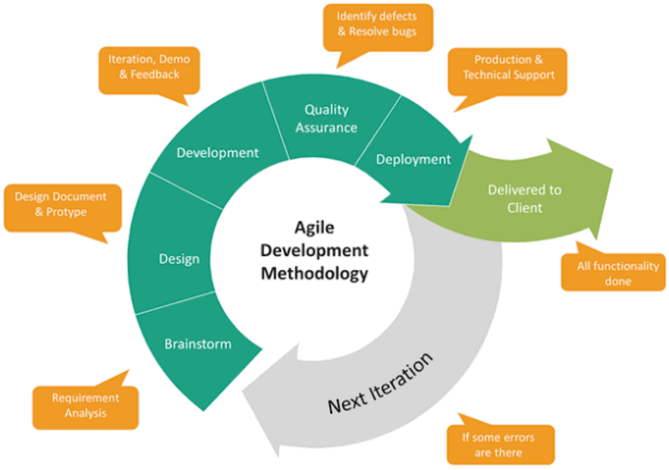


Figure 4 Agile Model SDLC (Gray, 2020)

In practice, the agile model upholds values such as continuous redefinition of roles for developers and operations experts. This implies that Agile antagonises the top-down flow of roles, which is popular with the waterfall model. Instead, the team selects and distributes tasks depending on experiences encountered during the software development (Meyer, 2014). The absence of a boss-like designation of roles results from removing a clear set of requirements at the beginning of the project. According to Meyer (2014), the agile argument is that extensive conclusive planning at the onset of development often turns out useless because of refactoring needs.

> "Requirements cannot be captured at the beginning of a project, because users do not know what they want. Even if one managed to write a requirements document, it would be useless because requirements will change through the project… Building a design upfront is a waste of time because we do not know what will work and what will not" (Meyer, 2014. p. 3).

Therefore, the agile model is described as a customer-centric software development model with a collaborative, self-organised team undertaking time-bound iterative development activities while focusing only on the software's essential functions (Huttermann, 2012). Huttermann (2012) further argues that Agile facilitates a seamless partnership between developers, testers, and

Quality Assurance (Q.A.) experts because there is a shared goal and understanding of the continuous changes in the software code.

Haines et al. (2017) investigated the agile model within the application context among software developers, product managers, and management team members to test its use probability. They found that "agile methods increase productivity on the products for customers, but lacking communication is a common negative factor that stands out among all the cons." Schwaber and Beedle (2002) echo this thought on communication challenges, stating that while constant feedback and communication are possible among teams working on small projects, they are much more tasking when used on large projects, given the likely multiple sprints involved.

In summary, the evolution of strategies and methods that resulted in DevOps as it is now forms a necessary backdrop for the current study. This study will investigate what respondents agree to be the necessary foundational practices that facilitate the success of DevOps and identify what features have consolidated into DevOps as practised now.

2.3.2 DevOps Principles and Values

For Hutterman (2012), DevOps draws inspiration from the same Agile principles that contributed to its development. DevOps extends these principles from essential software to other aspects of software delivery, such as customer satisfaction (Peuraniemi, 2014). Similarly, DevOps adopts components of Lean to make deployment easy for customers and management easier for I.T. experts and developers in the system.

Like Agile development and Lean, DevOps principles are value-driven and incorporate continuity, automation, flexibility, collaboration, and holistic thinking (Hutterman, 2012; Pueraniemi, 2014; Kristinsson, 2015). Hutterman (2012) identifies respect for one another, commitment to shared values, collective ownership, and shared values as foundational values that promote collaboration among teams (p. 27).

Communication between developers and operations specialists in DevOps is essential to facilitate collaboration. This helps team members develop understanding and respect for one another and commit to shared goals while ensuring that every member plays their role in the software delivery process (Peuraniemi, 2014).

The iteration focuses on mitigating repetitive problems in the system and strengthening weak areas. According to Hutterman (2012), continuous iteration ensures quality assurance in development and operations. Similarly, automating software development facilitates ease of use for customers and creates opportunities for developers to focus on code writing. Automating processes increases software's self-sufficiency, reducing human errors and boosting productivity.

The principle of holistic thinking emphasises empowering the DevOps team to approach and tackle problems with a consideration for all components involved, including the end user. DevOps is, therefore, deemed successful if it combines efficient processes with technology and human resources (Hutterman, 2012, p. 41).

DevOps teams try to combine the essential principles into their processes and activities with the goal of driving DevOps values in their teams. A perfect combination of the DevOps principles holds value for the business, team members, and customers involved in the software. These values are summarised as business, human, and customer (Kristinsson, 2015, pp. 5 - 8).

2.3.3 The DevOps Life Cycle

A complete DevOps cycle involving a combination of several different stages is necessary for effective software delivery. Broadly, the DevOps cycle involves all stages of software delivery, from building to eventual delivery and feedback. Like other aspects of DevOps, scholars have scattered opinions on the number of stages involved.

Regardless, continuity forms an underlying factor across all stages of software development. Yarlagadda (2021) identifies a looped system of cycles connected by collaborative and repetitive actions in the software development cycle. The DevOps cycle incorporates repetitive stages such as planning, building, integration and deployment, monitoring, operating, and feedback.

For Kristinsson (2015), there are three main stages of the DevOps life cycle, which are further delimited into four phases by Gokarna (2023) as follows:

i. Plan and Measure

ii. Develop and Test

iii. Release and Deploy

iv. Monitor and Optimize

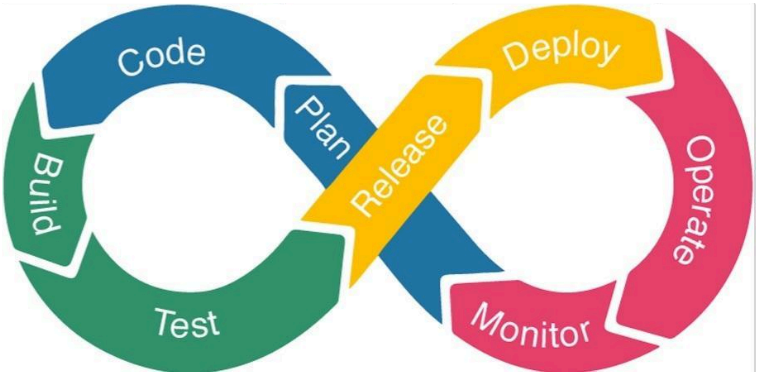


Figure 5 DevOps Life Cycle (Yarlagadda, 2021)

The DevOps lifecycle starts when an idea to solve an identified problem comes to life. In the planning phase, the vision is mapped out so that the DevOps team can fully comprehend the components and expectations of the projected software. Following the planning stage is the actual continuous development. DevOps actions are broken into implementable actions using several source code tools such as Grade, Ant, and SVN, among others (Gokarna, 2023). During development, continuous testing is done to check for the effectiveness of the software. Testing helps to ascertain that the developed software performs as expected. Also, through testing, it is possible to detect errors and bugs in the software code that could inhibit its eventual implementation and use.

According to Gokarna (2023), the four phases in the DevOps lifecycle further expand into six stages to accommodate all possible areas in software development and delivery, including continuous business planning, continuous development and integration, continuous testing, continuous release and deployment, continuous monitoring, and continuous feedback and optimisation.

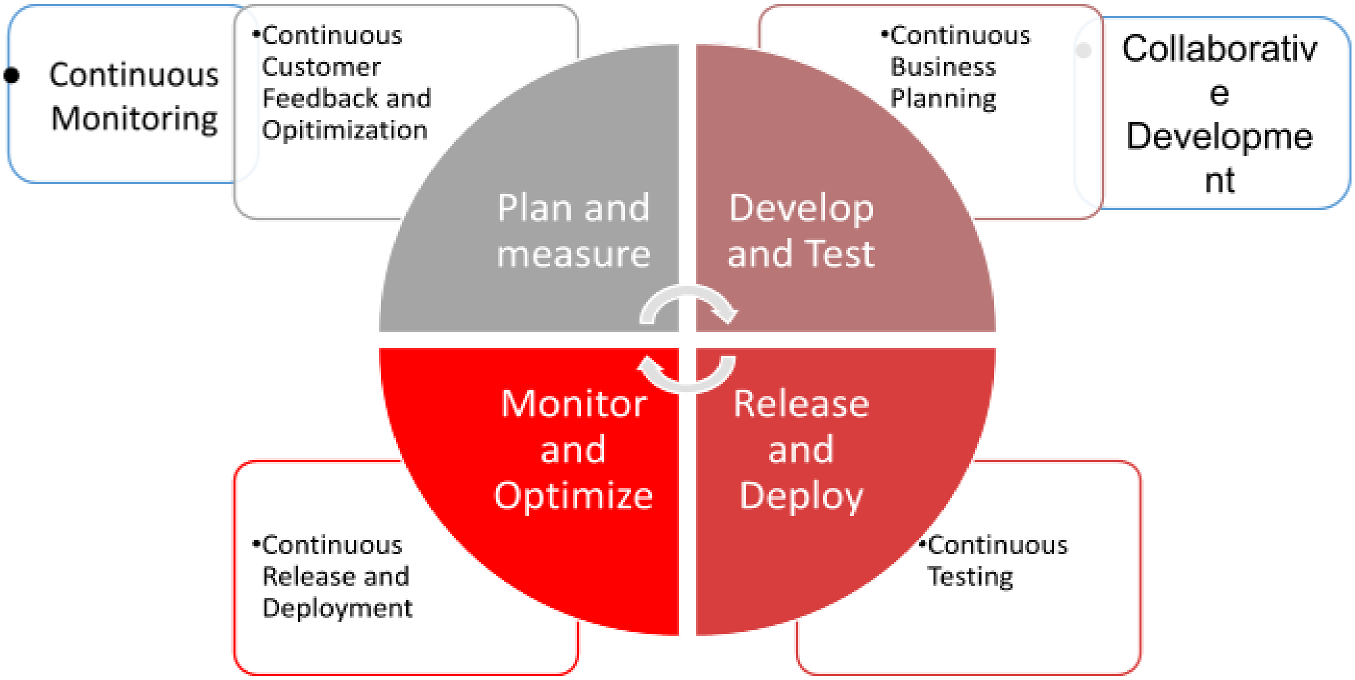

Continuous Monitoring
•Continuous Customer Feedback and Opitimization
•Continuous Business Planning
Collaborativ e Developme nt
Plan and measure
Develop and Test
Monitor and Optimize
Release and Deploy
•Continuous Release and Deployment
•Continuous Testing

Figure 6 DevOps Lifecycle (Gokarna, 2023)

The interlinked systems involved in the software development process imply no definite end to the DevOps stages (Yarlagadda, 2023). Therefore, continuous testing, for example, could follow even after deployment and release. More so, user feedback could necessitate further testing, and as a result, the code requires constant monitoring and optimisation for the best outcome possible (Kristinsson (2015).

Freeman (2019) holds that continuous integration and continuous delivery (CI/CD) only eliminate some challenges that could arise from software use. However, they position the DevOps team to identify better and resolve such disruptions quickly (p. 10).

> “But teams who work in a DevOps methodology find solutions faster through better coordination, more open accessibility, shared learning, and better performance monitoring… The business side of your organization

> will see fewer customer complaints, faster delivery of new features, and improved reliability" (p. 11).

DevOps requires the right combination of people, tools, and processes to oversee every stage of its life cycle for a successful implementation. Collaborative partnership is a crucial element needed to ensure the software delivery process's efficient and smooth running (Gorkana, 2023; Yarlagadda, 2021; Bob, 2021). Setting a clear lifecycle from the start helps the development and operations teams involved in software delivery navigate the process much more accessible through effective collaboration and a mutual sense of goal (Kristinsson, 2015, p. 24).

Therefore, this study will build on existing knowledge by determining what components of the DevOps process respondents believe are essential for the success of DevOps in software delivery.

2.3.4 Why DevOps: Understanding the Benefits of Adopting DevOps for Software Delivery

Implementing DevOps holds values that benefit customers, software developers, and the business. Crowley et al. (2018) identified the core benefits of DevOps as follows:

> "… higher organizational performance, faster and more efficient delivery, services that are more reliable and higher quality products. Customer satisfaction is increased… The work experience of the employees is enhanced."

In a 'State of DevOps' report based on a survey involving 25,000 technical experts, the evidence established that organisations with high-performing DevOps are twice as likely to surpass their market share, productivity goals, and profitability targets (Puppet Labs, 2015). The study also found that such organisations were 60x less likely to experience failures while recording 200x shorter lead times (Crowley et al., 2018; Brown et al., 2016).

Similarly, DevOps has been found to integrate development and operations efficiently enough to improve the rate of software delivery cycles and aid the delivery of quality software products aimed at specific problem-solving (Fitzgerald & Stol, 2018). By facilitating mutual understanding, smooth communication, and understanding between teams, DevOps improves productivity in development and operations processes (Mohamed, 2015) while reducing production costs by 20% (Elbert et al., 2016).

Where customer demand is high, DevOps makes it easy for I.T. teams to meet customer needs and deliver value quickly and efficiently (Sharma & Coyne, 2015). Automation also plays a role in DevOps, helping teams detect problems, make fast data-driven decisions, and respond to customer complaints promptly. Therefore, businesses can boost customer satisfaction using optimised resources (Hussaini, 2015; Duvall, 2012).

Overall, the 'State of DevOps' survey reported a positive atmosphere in the business environment by boosting motivation and ensuring collaboration rather than blame among developers and operations experts on one hand and I.T. experts and customers on the other (Crowley et al., 2018 Puppet Labs, 2014).

This study will attempt to cross-verify the benefits of DevOps in businesses based on findings from the literature. Respondents will assess the benefits they believe DevOps has brought to their business, organisation, or project.

2.4 DevOps in Use: An Empirical Review of Selected Literature on DevOps

This section presents and discusses the findings of previous direct studies on applying DevOps. The goal of this review is to identify current gaps in the literature, specifically as applied to software delivery.

Several studies have been conducted to determine the impact of DevOps on software delivery. Senapathi et al. (2019) conducted a case study on the implementation of DevOps in a New Zealand-based product development organisation. Six software engineers were tasked to monitor the gradual implementation of selected DevOps practices and principles over six months.

This review investigated the outcome of DevOps tools when applied in non-software situations to ascertain their applicability further. In an empirical assessment, Mohammed (2015) revealed that information management systems (MIS) can be facilitated through DevOps strategies. The core focus was determining whether DevOps can improve the delivery speed of MIS without sacrificing product quality.

Similarly, Erich et al. (2017) conducted an interview-based study of DevOps usage across six randomly selected software organisations. The results showed positive reactions to DevOps and further testified that DevOps implementation improved business operations. These studies, however, are relatively broad in orientation and do not extend to in-depth investigations of specific DevOps tools.

Rzig et al. (2022) provide the most streamlined study on DevOps by focusing on Machine Learning (ML) tools and analysing their contributions and benefits. They analysed 4031 ML projects hosted on GitHub (Rzig et al., 2022) and provided empirical evidence that adopting DevOps in machine learning (ML) increases productivity, especially in development.

In summary, the studies' results revealed that the case organisations experienced noticeable benefits in business operations, including increased collaboration among development and operations teams. DevOps was also found to boost quality assurance and the frequency of deployments.

2.5 A Review of Selected DevOps Tools and Frameworks

This section presents secondary findings and discussions of reports on selected DevOps tools in application. Each phase of the DevOps life cycle requires specific tools for successful and efficient execution. Therefore, several researchers have investigated selected DevOps tools over time to document empirical results on their efficacy.

Many different DevOps tools are classified along different factors based on their functions in the development life cycle, including collaboration, configuration management, planning, automation, security, testing, and many others (Crockett, 2023). Consequently, there is no exhaustive list of DevOps tools, and the scope of this review only expands to some available tools in use. Instead, it presents a summative look into the following DevOps tools randomly selected from different groups based on function:

Types	DevOps Tools
Version Control	Git and SVN
CI/D	Jetkins and Gitlab

Configuration Management	Ansible and Puppet
Containerization and Orchestration	Docker and Amazon ECS
Monitoring and Logging	Prometheus and Grafana
Infrastructure as Code (IaC)	Terraform and Azure Resource Manager
Testing tools	Selenium and JUnit
Artifact Repository	Nexus Repository Manager and JFrog Artifactory
Security	Aqua Security and Sonaqube

Table 1 DevOps Tools and Frameworks

In a systematic review, Aljundi (2018) collected reports from an investigation of 50 DevOps tools selected from the classifications above, ranging from integration to repository and cloud security. Similarly, Azad and Hyrynsalmi (2023) examined what constitutes the critical success factors (CFSs) of DevOps tools based on their performance in contextual use. According to Ghantous and Gill (2017), cataloguing the tools and their effectiveness can help organisations determine what specific tools suit their current project needs.

2.5.1 Version Control Tools

Version control tools are adequate for software development change tracking and code control. Where change is iterative, as is common in DevOps, developers risk getting lost while implementing changes to the source code, possibly suggested by collaborators and partners working on the same project. Blischak et al. (2016) provide evidence of Git as an efficient version control system that allows developers to experiment with new ideas without the risk of losing a previous version. A minor downside, however, is that while Git is most suitable for plain text files such as protocols, eLab notebooks, and manuscripts, it gives little room to commit 'large data files that hardly change' (Blischak, 2016. p. 11).

Conversely, Subversion offers extensive features such as copy-modify-merge and working copies, making it suitable and efficient for multiple source codes and projects (Collins-Sussman et al., 2004).

2.5.2 Continuous Integration and Continuous Deployment Tools

Continuous Integration and Continuous Deployment facilitate optimising the software delivery process (Wabgrant, 2022). With DevOps, software delivery involves a series of changes to the development process until the required outcome is achieved. CI/CD requires automatic release to delivery and automatic deployment for production. Automating the process makes Continuous Integration and Continuous Deployment much more manageable for developers and operations experts. It also helps to identify issues early, implement feedback, effect changes, and improve the overall software quality (Nikhil, 2017).

Therefore, CI/CD tools such as Jenkins and Gitlab have been proven effective in the project delivery pipeline. In a comparative study, Virtanen (2021) reviewed the effectiveness of CI/CD services, including Jenkins and GitLab. Jenkins was discovered to be a more effective platform for automatic CI and CD.

While Gitlab is a broader platform with more development functions, Jenkins makes automatic integration easier, allowing for faster testing, building, deployment, and delivery (Mohanan, 2023). However, Singh (2022) found that Gitlab equally facilitates all four components of CI/CD, including testing, releasing, deploying, and operating (p. 15). A minor challenge with Gitlab is that its operations can be slow if the process involves heavy data images (Singh, 2022. p. 50).

2.5.3 Configuration Management

Ansible and Puppet efficiently automate the software configuration network, contributing to faster configuration and increased consistency (Wabgrant & Radic, 2022). They operate on two distinct sets of properties. In comparison, Ansible is based on Linux and macOS; Puppet uses Linux, macOS, and Windows, which makes it more accessible to users. Similarly, the source code of Ansible is written in Python with the YAML (Yet Another Markup Language) script language while Puppet is written in Ruby.

With a focus on consistency, Wabgrant and Radic (2022) compared Ansible and Puppet and evaluated their performance through experiments. Overall, they found that the significant performance distinction between configuration management tools lies in the volume of configuration changes involved. With lower configuration changes, Ansible performed better than Puppet. Conversely, Puppet performed more optimally when more enormous configuration changes were made (Wabgrant & Radic, 2022, p. 15).

2.5.4 Containerization and Orchestration

Containerisation of applications has continued to gain momentum in software development thanks to its multiple benefits (Anne, 2023). Containerising software helps it run independently, with zero interference and compatibility problems. Container orchestration is the "management of container lifecycles, particularly in large, dynamic environments" (Kaiser et al., 2022. p. 84863). Container orchestration involves automating the essential work needed to help the containerised services operate.

According to Kumudavalli and Venkatesh (2021), Docker is the most popular containerisation technology. It allows the packaging of application software into compact containers containing all components needed to help the application run smoothly, including the application source code and associated dependencies (Kaiser et al., 2022).

As a container orchestration service, Amazon ECS helps developers manage and scale containerised software. Altaleb and Khalaf (2022) maintain that the unique point of Amazon ECS as a Container as a Service (CaaS) tool is that it supports Docker, thereby reducing management complexity problems, particularly in "storage, servers, security and networking" (p. 2). Moreover, the scalability of ECS applications reduces operation costs, an essential feature for developers working on cloud-based projects.

2.5.5 Monitoring and Logging

Logging can be tedious in DevOps because it involves statements that can be read while the program runs. The log statements must display the program's state and actions at each point (Freeman, 2019). This helps the developer(s) identify and navigate the changes made to the code without combing through the entire code at each point in time.

Monitoring is more diagnostic in orientation; it is essential to check the software's "performance, availability, security" (Freeman, 2019, p. 154) and other vital components after launch.

Prometheus and Grafana are monitoring and logging tools that provide information about the application or software's performance. Leppanen (2021) conducted a data visualisation and monitoring process using the combination of Prometheus and Grafana.

According to Leppanen (2021), Prometheus contains optional features needed to monitor software successfully, including an alert manager, exporters, libraries, and supporting plugins. However, most of the Prometheus features operate on third-party issuers. Similarly, Grafana supports several data sources, including Prometheus, which are primarily built-in features, making it easy for the developer(s) to assess application performance (p. 24).

2.5.6 Infrastructure as Code (IaC)

Developers often prepare blueprints with actionable specifications for orchestration to remove unnecessary steps and save time. This tactic results in a practice called Infrastructure as Code (IaC) (Artac et al., 2017). IaC facilitates and speeds up software operations since several components of the DevOps process are configured and automated using the same standard language.

Terraform optimises the software delivery process by helping developers provision and manage infrastructure efficiently. Since Infrastructure as Code makes management possible, developers spend less time and effort on code management and less time on less important details. Similarly, Azure supports scripting in code and makes changes easy for developers to read and track.

2.5.7 Testing

Testing in DevOps refers to subjecting a software product to processes such as reviews and validation to ensure it meets expected standards and achieves the desired goal. The stages of testing in DevOps are differently classified, but most testing tools incorporate unit testing, integration testing, and end-to-end testing.

Given that most processes in DevOps are iterative, testing is best automated. Therefore, automated testing is considered much more practicable when tools such as Selenium and JUnit are used. A further review of Selenium reveals that it is a preferred testing tool in the software development industry. Bruns et al. (2009) hold that Selenium is a viable choice because it caters to the need for lean, automatic, and progressive testing since web applications and software products tend to evolve. Moreover, Selenium can be implemented with diverse frameworks for functional tests, including JUnit, Ant, and Maven.

2.5.8 Artifact Repository

The evolving need for software application upgrades necessitates an artefact repository. Given that the optimal success of any software product depends on continuous development and integration, 'storage' is necessary to keep updated artefacts safe for subsequent deployment. Artefacts may include source codes, prototypes, container images, configuration files, and binary software.

Nexus Repository Manager and JFrog Artifactory are standout artifact repositories with excellent features. In a comparative review, the Sonatype Nexus Repository was more robust than the JFrog Artifactory. It allows for customisation, faster deployment, and integration with supporting tools.

2.5.9 Security

DevOps security has garnered more concern over time, and understandably so because the process from software ideation to delivery is elaborate. There are many changes in features, and several developers are involved at one point in the life cycle or another. In the process, the source code or other third-party kits may contain bugs that are likely to expose software users (clients or organisations) to security threats.

While software developers are not primarily charged with security duties, they owe a responsibility to the organisation and potential code users to adopt security measures in the software delivery process. DevOps security tools such as Aqua Security and Sonarqube reduce software products' vulnerabilities.

The Aqua Security architecture is equipped with capabilities to detect threats in libraries and dependencies. It also prompts developers on suspicious incidents and activities in the package. The threat with regular DevOps pipelines has increased the use of severless security. Where serverless software is used, Aqua Security has features and capabilities to provide an extra layer of security. Similarly, Sonarqube is a reliable software security tool. Sonarqube can be automated to scan and detect potential vulnerabilities in the code at each build and testing level.

In summary, DevOps tools are inexhaustible, and this review only attempts to investigate some available tools. Moreover, emerging needs and concerns in DevOps continue to facilitate the development of more efficient tools across all lifecycle stages. Therefore, the present study aims to identify features of the selected tools that could guide organisations in choosing DevOps tools to achieve optimal software delivery.

The reviewed literature has advanced knowledge of DevOps tools and applications, but only to the extent of broad organisation-based use. Therefore, it contributes a foundation for this study and leaves a gap for further research, particularly on how DevOps may be optimised for software development in specific organisational scenarios.

2.6 Current Challenges in the Software Delivery Pipeline

The emergence of DevOps has proven to be a massive game-changer in the software development industry. Evidence exists of businesses and organisations witnessing significant operations boosts through innovative software input.

Regardless, the road to software delivery is not hitch-free. Every software product undergoes a series of stages and processes from development to launch. Scholars, researchers, and software developers have reported significant challenges the software delivery pipeline faces. In a secondary investigation, Shahin et al. (2017) highlighted some factors reported across the literature as setbacks in continuous software delivery. Issues identified ranged from infrastructural to behavioural inhibitions among organisations and directly among developers working on the projects. They include costs of operation, resistance to change among stakeholders in organisations, and excessive dependencies in design, among others. The figure below presents a comprehensive but non-conclusive list of challenges in software delivery:

Type	Challenge
Resources	High costs
	Lack of experienced and skilled DevOps teams

	Excessive workload on available teams
	Absence of updated tools, methods and technologies
Organizational	Poor organizational culture
	Lack of shared goals
	Poor communication among team members
	Resistance to emerging software development technologies
	Lack of trust among team members
	Excessively large
Architecture	Lack of a working architecture for software development
	Lack of compatibility among components
	Dependencies in design
	Excessive changes in database, etc.
Security	Exposure to third-party risks

Table 2 Challenges in Software Deployment and Delivery Pipeline (Shahin, et. al. 2017)

Following these findings, the current study will ask respondents to identify how many of the challenges they have faced or are currently facing in using DevOps to develop and deliver solution-oriented software in their organisations. Understanding the common challenges faced in software delivery will aid the development of a guiding framework for DevOps experts.

2.7 Gaps in Literature

Previous studies have established and emphasised the positive aspects of DevOps tools. Results from different empirical research align with the benefits of DevOps implementation across organisations and businesses. However, more literature needs to focus on what features of DevOps tools are best for specific software needs and scenarios among organisations.

Therefore, this study aims to identify what DevOps features will support optimal software delivery in chosen scenarios and should be prioritised in the DevOps process. This research is original in its scope, approach, and contribution to DevOps scholarship. As part of an extensive comparative analysis, the study will assess selected DevOps tools to ascertain how well they embody the founding principles of DevOps and whether they offer benefits along with the three core values of DevOps.

This investigation aims to determine a working framework for selecting and implementing cloud-based DevOps tools among organisations in different scenarios. A framework will simplify the DevOps tools selection process among organisations and, more importantly, facilitate business operations through optimal tool selection and implementation.

To achieve this objective, semi-structured interviews with DevOps experts will be conducted. It is noteworthy that most previous studies in DevOps have equally adopted interviews. However, to substantiate the qualitative findings from the interviews in this study, further comparisons between selected tools across crucial dimensions will be made. The research will evaluate the suitability of selected DevOps tools for use in different organisational scenarios, focusing on vital dimensions such as features, scalability, cost, and compatibility. Combining interviews and comparative studies will bridge the loophole that previous literature research has created by relying solely on interviews (Erich et al., 2017; Senapathi et al., 2019) to determine the efficacy and performance of DevOps tools.

Findings from this study will bridge the existing gap in software development literature by determining whether any DevOps tools can be applied for optimal software delivery. This study will also help develop a framework to guide software developers and organisations in developing, implementing, or adopting the most efficient software tools to drive organisational goals.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

The methodology of this research project revolved around conducting a comprehensive comparison of cloud-based DevOps tools to derive insights practical for software delivery in various organizational contexts. The primary academic aim was to evaluate and contrast the effectiveness of these tools across different dimensions.

The present study has categorized the comparison of DevOps tools into several significant dimensions that are crucial in determining their viability and efficacy across different organizational settings. Some of these dimensions comprise:

Features: These include examining the range and type of functions the tools provide, how they meet DevOps-specific requirements, and how they affect the software life cycle.

Usability: This entails analyzing a tool's ease of use, learning curve, availability of user support, and overall user experience provided by the user interface.

Scalability: To determine whether the tools can meet increase in demands that will aid the organization to have 100% uptime or close to it.

Integration Capabilities: Looking at how the software can be easily integrated with other existing software used in organizations as well as compatibility with existing workflows and technologies

Cost: The upfront costs attached to each solution and their ongoing costs will also be assessed to evaluate their cost-effectiveness, followed by an understanding of how organizations might fit them into their finances.

Security: Such security features will cover measures related to data protection, conformity to relevant standards, and promoting secure development.

3.2 Research Design

The research design incorporated qualitative elements. In the qualitative segment, in-depth interviews provided detailed insights into user experiences with DevOps tools. This approach was supported by secondary research, including succinct comparisons between the leading DevOps software within their industry categories. The qualitative approach was chosen to effectively answer the question "Why?" Applying this approach will enable more in-depth responses than polar responses, making this study much easier to understand. The qualitative approach helped to reveal the underlying reasons behind tool selection and practices.

As DevOps tools do not operate in a space but in organizational practices, a qualitative approach helps to capture these practices and contexts, as shown in the results. These contexts, in turn, impact the effectiveness and suitability of these tools when we understand how these tools fit into the broader scheme of things. This, in turn, helps with informed recommendations that are contextually relevant.

The more in-depth insights this approach yields, the better the framework is being built for better understanding.

3.3 Data Collection

3.2.1 In-Depth Interviews:

The researcher conducted in-depth conversations with experts in DevOps. These discussions covered various topics, including the capabilities, user-friendliness, growth potential, compatibility with existing systems, and financial implications of the tools they used. Three professionals were interviewed, and while they are all DevOps Engineers, they have specific roles they play in their organization. Their roles gave interesting responses, and factors such as organizational practices, company mission and vision, and job position influenced the responses provided.

The choice of interviews was necessary for this study as it aided in uncovering the motivation behind tool preferences, the challenges faced by professional users, and other organizational contexts that influence the choice of these tools.

3.2.2 Secondary Research:

Secondary research was undertaken in the literature review, reviewing the top two DevOps software in each category and comparing them according to their abilities. This was done to support the data obtained from the interviews.

Conducting secondary research is also necessary to support the primary, as not all the tools mentioned in the comparison were mentioned in the interviews. In the security arena, for instance, categories like that are being outsourced to a third-party organization or unit that helps the DevOps Engineers perform their jobs better.

The secondary goal was also to give insight into the category and the needs that the tools in these categories are meant to meet.

3.4 Data Analysis

3.4.1 Thematic Analysis:

Thematic analysis was applied to assess the collected data. Themes within the interview data were analysed to identify patterns and gather valuable information about the tools used by the professionals. Thematic analysis was chosen because it is well-suited to structuring and extracting meaningful patterns from qualitative data. It is also flexible, aids in producing detailed insights, and helps in comprehensive data presentation.

3.5 Rigor and Validity

3.5.1 Pilot Studies:

A Pilot study was conducted on a sample before the primary data collection phases to refine the interview protocol and ensure clarity in interpreting collected information.

During this process, I discovered that one of the questions needed to be more specific and clear to the participant. It was based on the features of the tools, and it was corrected.

Vague question: In your opinion, what are the most valuable features of the cloud-based DevOps tools your team is using?
Corrected question: What features of these tools do you find most valuable as an individual and a team?

The data gathered from these pilot studies was later converted to one of the main interview responses, as contacting the participant proved difficult. It will also serve to repeat the same questions and responses. In light of that, necessary corrections were made during the pilot interview process.

3.5.2 Triangulation:

The research process combined findings from in-depth interviews with data extracted from technical documentation, literature, and industry reports to enhance the study's reliability and validity.

3.6 Presentation of Results

3.6.1 Tables and images

The study's findings were presented through detailed comparisons, incorporating reports from studies in the literature review and using tables and images to convey the results clearly.

3.6.2 Error Acknowledgment:

Any errors or limitations encountered during the research process were acknowledged, and corrective measures were taken to improve the study's reliability.

3.7 Anticipated Areas for Critical Reflection in the Final Report

3.7.1 Challenges and Iterative Improvements:

- The final report elaborates on challenges faced during data collection and the iterative improvements made based on ongoing analysis and participant feedback. It also addressed the implications of potential biases on the study's outcomes, ensuring a comprehensive and reflective conclusion to the research.

3.8 Ethical Considerations

Before the commencement of the interviews, ethical approval was sought from the university management, and it was approved. The analysis was conducted with strict adherence to ethical standards, ensuring the confidentiality and anonymity of the participants. All data was treated with the utmost respect for the participants' privacy, and quotes included in the findings were carefully selected and anonymized to prevent identification.

CHAPTER 4: DATA ANALYSIS

4.1 Introduction

This chapter provides concise insight into the interviews conducted among DevOps experts to assess DevOps tools using qualitative methods. Relevant themes were created and analysed using Atlas Ti (Version 24). Each of the objectives is discussed with its associated themes to align with the study objectives.

4.2 Evaluation of DevOps Tools through interviews and comparative analysis of the tools.

The discussion covered the assessment and comparison of DevOps tools, focusing on vital aspects such as their most valuable features, their application across different organisational contexts and tasks, the difficulties encountered when using these tools, and the strategies employed to address these challenges.

4.2.1 Theme 1: Most Valuable Features of DevOps Tools

As participants established in Figure 6, User Friendliness is the most valuable feature of a DevOps tool. Other features include problem-solving, an exemplary user interface, easy integration with an existing system, automation, and security.

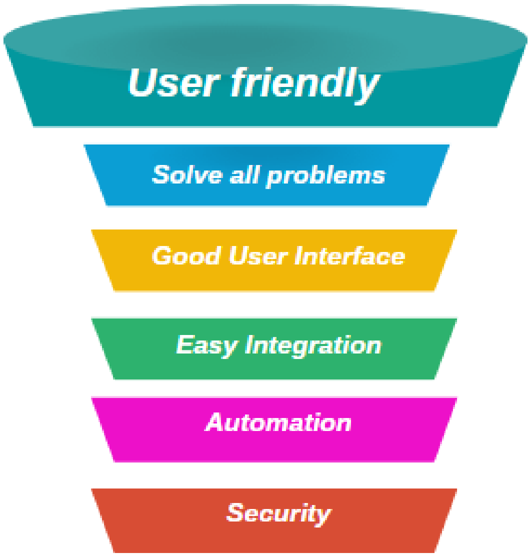


Figure 6: Features of DevOps Tools

Here are some excerpts:

“... I think the major key thing I would consider is being user-friendly and able to solve all my problems” (Participant 1, DevOps Engineer)

“...In fact, for you to even grasp what exactly they are trying to do, it becomes like you are studying- ***It must be user-friendly. Yeah, very important. And the funny thing about these is that you have to sync with your team because what is user-friendly for Mr. B might not be user-friendly for A.*** *So that user-friendly environment, that's what we look into” (Participant 2, Solution Architect / DevOps Engineer)*

“...If there's a tool that can help with automation. Because that's one of the goals. *When you are doing something repeatedly, try to find a way to automate that. Automation and security, because most of these tools are connected to production environment”* (Participant 3, Site Reliability Engineer)

4.2.2 Theme 2: Explored DevOps Tool Utilisation across Diverse Organizational Landscapes and Work Scopes

Exploring the different perspectives of the participants in the study based on the DevOps tool adopted in their organisations revealed numerous DevOps tools based on the organisation's needs. For a payment processing company in Scenario A, Docker is used for containerisation, Jenkins is used for continuous integration and deployment, and Prometheus and Grafana are used for monitoring purposes. Furthermore, Elasticsearch, Logstash, and Kibana (ELK Stack) are used for centralised logging, and Nexus is used for artefact storage. However, in Scenario B, which is a Digitalisation organisation that focuses on software building and workspace automation - Jenkins, Docker, and Kubernetes are being used for integrations and deployments, and AgroCD is for streaming deployments and providing architectural insights, GitHub is for code repositioning, while in Scenario C which is for an organisation that design, build, secure and scale APIs - Terraform is being used for infrastructure provisioning, Amazon EKS for container orchestration in order to automatically provide, deploy, scale and manage containerised applications, and occasional use of Terraform, Teragrunt, and Ansible for infrastructure automation. Still, in Scenario C, the participant revealed that Golang and Python were tools used and pre-empted to be development tools used for programming.

Notably, the use of Jenkins and Docker was established in Scenario A and Scenario B, while the use of GitHub was established in Scenario B and C.

As outlined in Table 3, various organisations employ distinct DevOps tools tailored to their specific organisational requirements.

Scenario A - Payment processing organisation	**Scenario B - Business Digitalisation, Application Integration, Software Building, and Workspace automation organisation**	**Scenario C- Design, building, securing, and scaling APIs organisation.**
Infrastructure and Deployment Uses Docker for Containerisation Uses Jenkins for continuous Integration and Deployment (CI-CD)	**Continuous Integration and Deployment (CICD):** Uses Jenkins, Docker, Kubernetes for continuous integrations and deployments. Exploring AgroCD as a new deployment tool for streamlining deployments and providing architectural viewpoints.	**Infrastructure Management** Uses Terraform for infrastructure provisioning. Uses Amazon EKS (Managed Kubernetes Service) for container orchestration. Occasional use of Terraform, Teragrunt, and Ansible for infrastructure automation.

Monitoring and Logging Uses Prometheus and Grafana for monitoring purposes. Uses Elasticsearch, Logstash, and Kibana (ELK Stack) for Centralised Logging	**Version Control and Collaboration:** Uses GitHub for code repository management and collaborative development.	**Development Tools** Uses Golang and Python are the programming languages Uses GitHub for code repository management and collaborative development**.**
Artifact Management: Uses Nexus for artifact storage	**Automation:** Rarely use Ansible	

Table 3: Comparison of various DevOps tool across Diverse needs (Source: Author, 2024)

Below are some of the excerpts used in developing the table above:

"...Okay, we use Terraform, EKS, that's Amazon Humanities, Managed Humanities Service. And then, Terraform, Teragrunt, Ansible, occasionally. Okay. And then, Golang, Python. Those are like, some of the major tools I use. GitHub too." (Participant 3, Site Reliability Engineer)

"...For example, some of the tools we make use of in our organization, like cloud-based tools, like Jenkins, Docker, Kubernetes, for continuous integrations and deployments, like CICD. These are the handy tools that we make use, you know. So those are the tools that I use day in, day out…" (Participant 2, Solution Architect / DevOps Engineer)

"Currently, we use Docker for containerization. For monitoring, we are using Prometheus and Grafana. Okay. For centralized logging, we are using ELK Stack. So currently, for our build process, which is the CI-CD pipeline, we are using Jenkins. Then when it comes to our monitoring, we use centralized monitoring, which we were using for Prometheus and Grafana…" (Participant 1, DevOps Engineer)

4.2.3 Theme 3: Challenges and Pain Points in Using DevOps Tools

As shared by study participants, increases in cloud cost, knowledge gap, learning curve, difficulty in integrating the DevOps tools with legacy systems and ensuring consistent performance across different departments were challenges faced in using the DevOps tools. Other identified challenges include more support and adequate documentation of open-source DevOps tools. Furthermore, at the organisational level, the acceptance of new implementation by the management staff could also be a pain point. Here are excerpts that speak to the challenges as mentioned earlier:

"All right. Some challenges are mostly for open source tools. You have very little management and support. So, that's one of it. The second one is cost. Because as cloud costs keep increasing, some of those costs are going to be pushed to you. Okay. So, firstly, lack of support and adequate documentation in some cases, and then secondly, cost. Okay…" (Participant 3, Site Reliability Engineer)

"...Integration with legacy systems. You understand? There's some systems that are, sorry to say, they are obsolete, right? We need to learn where this is. So integration with legacy systems, ensuring consistent performance across different departments, right? So these are some of the challenges. Like I'm working for one, well, I'm partnering with one organization, a very well-known company in Nigeria, right? They have, okay, see they push their code almost every day. So we need to check, okay, what are you guys pushing? Okay, let us run this test. Let's test this place, you know? So you have to close monitoring across different departments…" (Participant 2, Solution Architect / DevOps Engineer)

"One of the pain points is when it comes to knowledge sharing, having knowledge about how those tools work, their functionality, and everything else. So that one, knowledge gap. Yeah. This could be a key factor. But a pain point is the organization accepting the new implementation you want to use. If the board, the head, the CTO, or your line management agrees with what you want to do, those are the key pain points" (Participant 1, DevOps Engineer)

4.2.4 Theme 4: Strategies in Tackling some identified challenges / pain points in use of DevOps.

Having established above the cloud cost, knowledge gap, learning curve, difficulty in integrating the DevOps tools with legacy systems and ensuring consistent performance across different departments, lack of support and inadequate documentation of open source DevOps tools, and the acceptance of new implementation by the management staff. The network analysis in Figure 7 below revealed the strategies adopted by these organisations in tackling some of the challenges. For instance, the organisation regularly updates its DevOps tools and security patch; on knowledge gaps, an individual or team makes an effort to bridge the knowledge gap by learning; At the same time, in tackling the learning curve, integration and consistent performance, the organisation regularly updates and maintain their DevOps and ensures thorough training of team members.

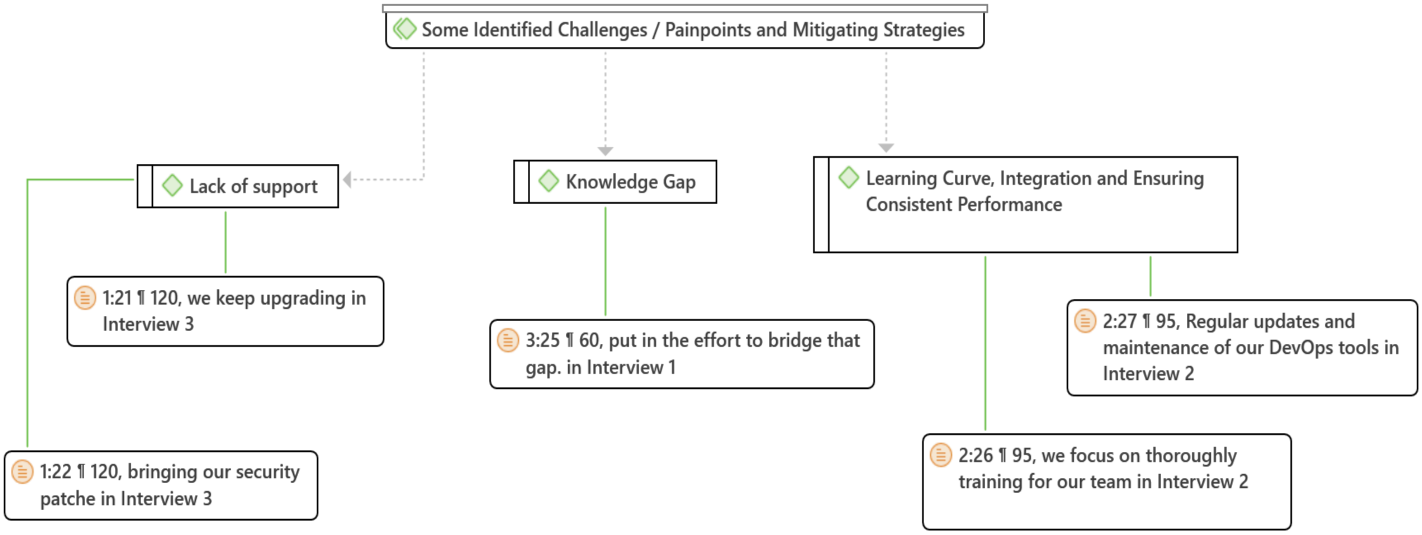

Some Identified Challenges / Painpoints and Mitigating Strategies
Lack of support
Knowledge Gap
Learning Curve, Integration and Ensuring Consistent Performance
1:21 ¶ 120, we keep upgrading in Interview 3
3:25 ¶ 60, put in the effort to bridge that gap. in Interview 1
2:27 ¶ 95, Regular updates and maintenance of our DevOps tools in Interview 2
1:22 ¶ 120, bringing our security patche in Interview 3
2:26 ¶ 95, we focus on thoroughly training for our team in Interview 2

Figure 7: A network analysis identified challenges and mitigation

4.3 A Qualitative Insight on the Selection Framework for DevOps Tools.

The interviews revealed some perceived selection criteria for DevOps tools. Research was identified as a selection criterion, with a focus on the tool's usability/user friendliness and cost effectiveness. Another criterion identified is the team's ability to have the required skills in using the skills and knowing if there are open source tools available.

Furthermore, the ease of integration, scalability, and community support were also identified as selection criteria. Below are excerpts from one of the above-raised points:

"All right, so, mostly, that decision comes about after a bit of research, right? So, research is focused on the usability, right? How can we use this application? And then, the cost effectiveness, because one of the goals of SRE is how to run software reliably, in a secured manner, and that software is costing the least amount of money. So, we look at, like, software that would cost us less, right, when we are doing research on what to use, and what is secure, what has a bit of support when we need to, if we are doing product software, right? And then, we look at the team, and how fast the team can get skilled in using that particular software. So, we don't make everything complicated, because at the end of the day, the team has to manage whatever drawbacks you get, whenever you have issues…" (Participant 3, Site Reliability Engineer)

"All right, so one of the criteria for selecting our tools includes the ease of integration, scalability, community support, and cost effectiveness, all right, because most of the time, community support is very important, because you might be using a particular tools, and if you don't have that support base, most especially AWS and Microsoft are two, they are a big company, but they have good communities, most especially Microsoft, all right…" (Participant 2, Solution Architect / DevOps Engineer)

“I said one of the key things that was put in place when selecting these solutions was user-friendliness, serving the purpose, the essential purpose, and why we need that tool. Being able to solve our problems, yeah. Okay…” (Participant 1, DevOps Engineer)

4.4 DevOps Knowledge and Practice

This details findings on the knowledge and practices of DevOps tools under the following themes: adoption of cloud-based DevOps Tools, effectiveness of DevOps tools between start-ups and established Organisations, best practices for selecting, implementing, and optimising DevOps Tools, and recommendations to organisations on the adoption of DevOps tools.

4.4.1 Theme : Adoption of cloud-based DevOps Tools

All three participants interviewed established their organisation's interest in adopting cloud-based DevOps tools. Participant 1 said, *"So- Yeah, we have plans in place to move, yeah. Because most of these old legacy tools have reached the end of their life. So I feel that for any organization that is planning to start up, I would advise going on cloud. Just move all your services to the cloud. The way they're going, yeah, the cloud is the future. Because most organizations now are adopting that cloud system"*. Other excerpt is shared below:

"I mean, we're always preparing for these changes. So in terms of software, in terms of the DevOps tools we are using, lots of these tools come with different versions, and you have to learn different versions to enjoy some of the benefits" (Participant 3, Site Reliability Engineer).

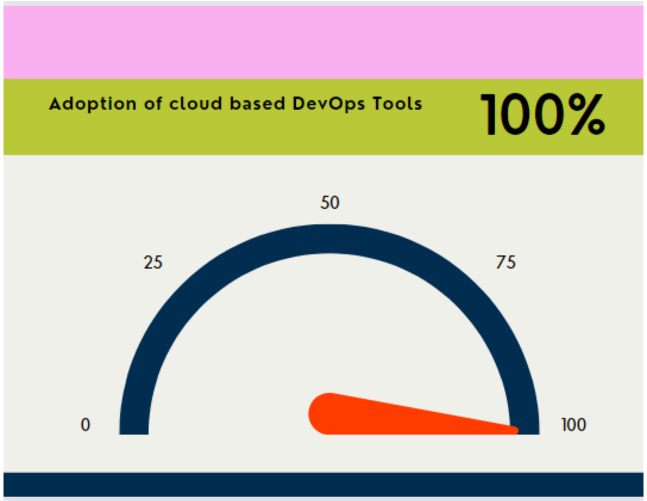


Figure 8: Adoption rate of cloud based DevOps Tools

Although cloud-based features give more uptime and could generate more revenue or funds for the organisation, failure to adopt the future trend of cloud-based DevOps tools could lead to longer time and troubleshooting of the system, as identified by Participant 1: *"So, most of the time, we focus on less important things, on things that are not our core responsibility. But if you are on the cloud, you won't have time for that. You will have time to focus on what really matters"*.

4.4.2 Effectiveness of DevOps Tools between Start Up and Established Organisations

Findings from the interviews revealed that start-ups try to keep their tools simple and cheap, ensuring that they are cost-effective compared to established organisations that use complex APIs and sophisticated shipped systems. Another perception was that start-ups often prioritise agility while established organisations focus more on scalability and integration of the DevOps tools with existing systems. However, cost-effectiveness was more profound for the start-ups.

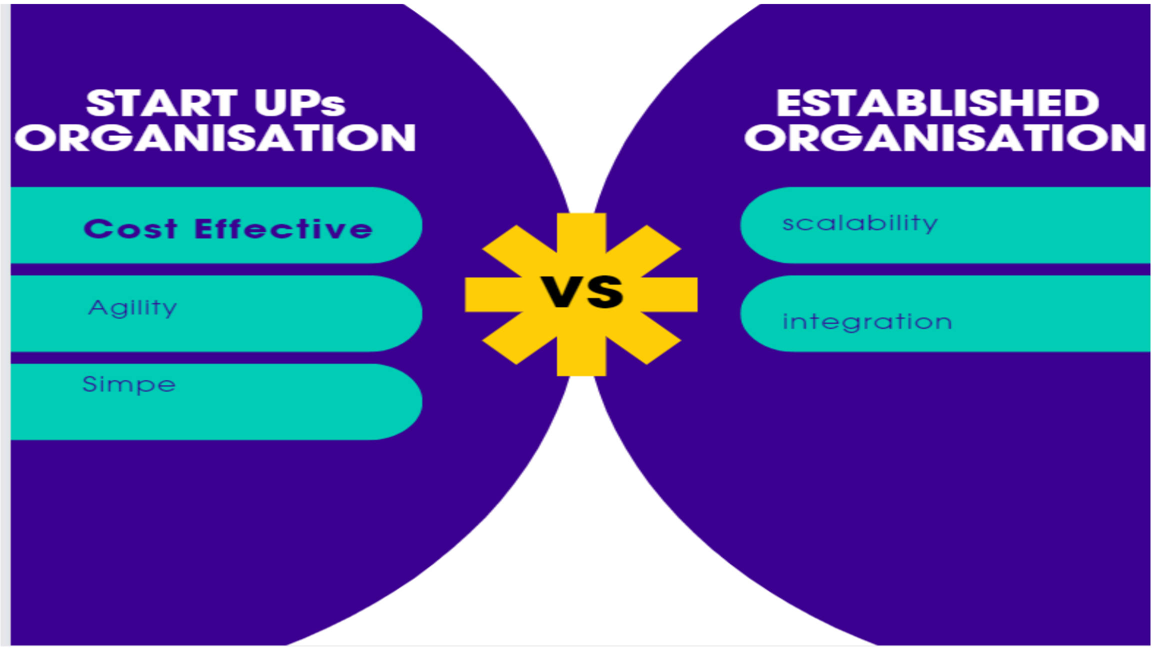


Figure 9: Factors influencing choice of DevOps Tools between start up and established organisation

4.4.3 Best Practices for selecting, implementing and optimising DevOps Tools

One of the best practices identified by participants is the use of a calculating system provided by tools such as the Azure Calculator or the AWS calculator. These calculators help estimate the amount to be spent in a day, month, and year, and they are usually free.

CHAPTER 5: EVALUATION & DEVELOPMENT OF FRAMEWORK

5. 1 Introduction

This chapter critically compares the findings from the interviews and existing literature to the study objectives, delving into insights shared by participants regarding the research questions

5.2 Selection Framework for DevOps Tool using BPMN Model

The DevOps Tool Selection Framework is a comprehensive and iterative approach that assists organisations in systematically identifying and adopting the most suitable DevOps tools. Using the Business Process Model and Notation (BPMN) 2.0 methodology, the framework provides steps to help an organization go through the complexities of assessing, comparing and choosing the best tool that fits their operational requirements and strategic goals.

Starting with a thorough user research phase allows a multilayered evaluation process where decision gateways are incorporated to ensure every tool is rigorously vetted against such critical organisational criteria as scalability, integration ease, community support, cost-effectiveness, user-friendliness, and open-source availability. Below is the tool selection framework:

Selection Framework for DevOps Tool using BPMN Model

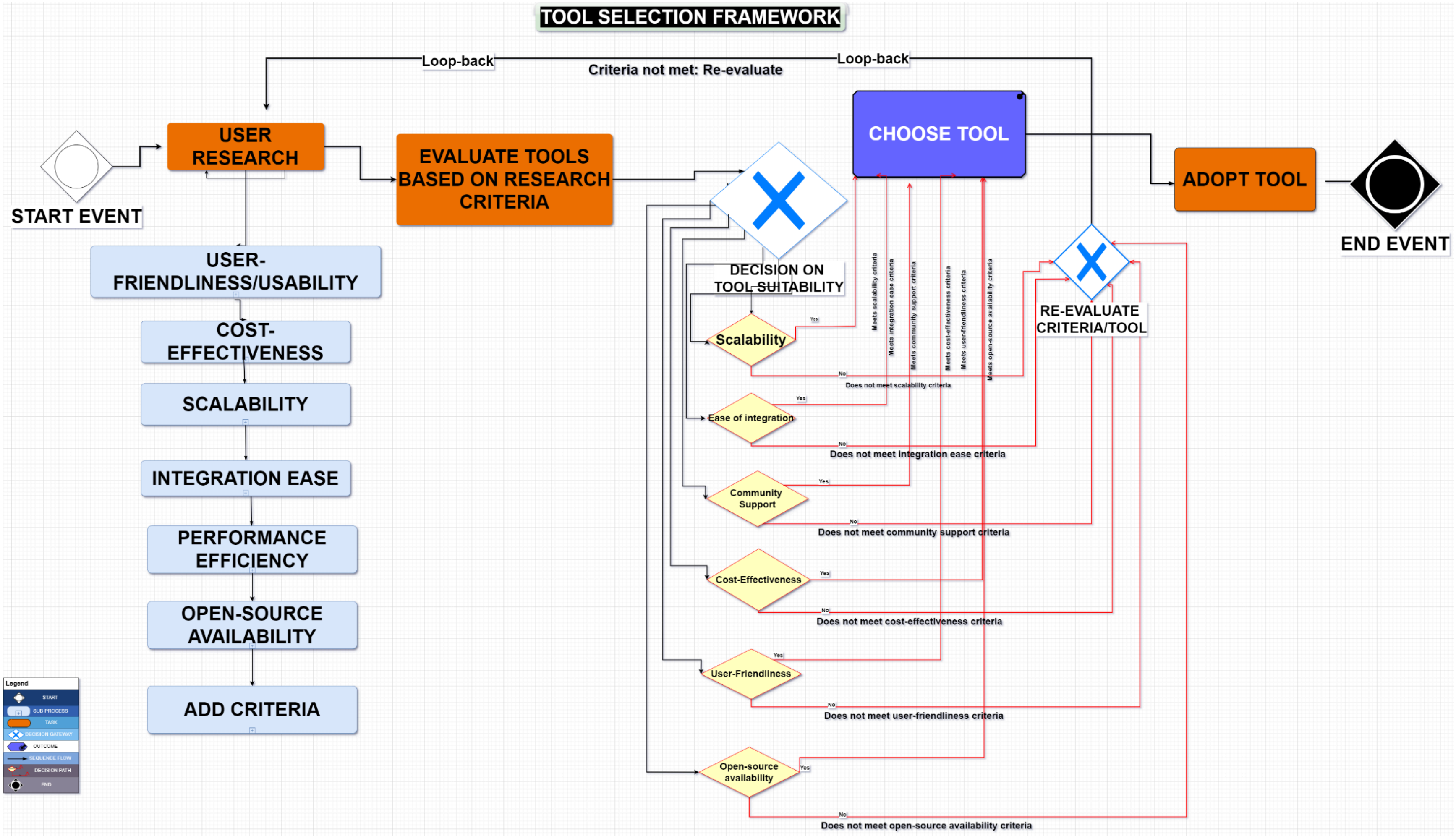

TOOL SELECTION FRAMEWORK
Loop-back
Criteria not met: Re-evaluate
Loop-back
START EVENT
USER RESEARCH
EVALUATE TOOLS BASED ON RESEARCH CRITERIA
CHOOSE TOOL
ADOPT TOOL
END EVENT
USER-FRIENDLINESS/USABILITY
COST-EFFECTIVENESS
SCALABILITY
INTEGRATION EASE
PERFORMANCE EFFICIENCY
OPEN-SOURCE AVAILABILITY
ADD CRITERIA
DECISION ON TOOL SUITABILITY
RE-EVALUATE CRITERIA/TOOL
Scalability
Yes
No
Does not meet scalability criteria
Ease of integration
Yes
No
Does not meet integration ease criteria
Community Support
Yes
No
Does not meet community support criteria
Cost-Effectiveness
Yes
No
Does not meet cost-effectiveness criteria
User-Friendliness
Yes
No
Does not meet user-friendliness criteria
Open-source availability
Yes
No
Does not meet open-source availability criteria
Meets scalability criteria
Meets integration ease criteria
Meets community support criteria
Meets cost-effectiveness criteria
Meets user-friendliness criteria
Meets open-source availability criteria
Legend
START
SUB PROCESS
TASK
DECISION GATEWAY
OUTCOME
SEQUENCE FLOW
DECISION PATH
END

Figure 8: Selection Framework for DevOps Tool using BPMN Model.

5.2.1 Framework Guide

This framework is designed to guide organizations through selecting the right DevOps tools. The framework uses a systematic approach that evaluates different potential tools against specific criteria to ensure the best fit for an organization. Below is a guide on how it works:

Start Event

The process starts with the '*Start event*,' which triggers the tool's selection. This is symbolic of an organisation evaluating tools to enhance its DevOps practices.

User Research

In the '*User Research'* phase, the company gathers detailed, in-depth information about various DevOps tools, including data on each tool's feature functionality, user reviews, pricing models, and compatibility with existing systems, among other factors.

Evaluate Tools Based on Research Criteria

When the above research is done, it brings us to the Evaluation phase where collected information is compared to pre-defined evaluation criteria that act as filters to disregard information that does not match business needs.

Decision on Tool Suitability

The Decision Gateway is the crucial juncture where each tool is tested against specific criteria:

Scalability: Can the tool handle the organization's growth and adjust to increasing demands?

Ease of Integration: How well does the tool blend with the current technical ecosystem?

Community Support: Is there a robust community or vendor support for the tool?

Cost-Effectiveness: Does the tool offer a favourable return on investment considering its total cost of ownership?

User-Friendliness: Will the team be able to use the tool without a steep learning curve? I.e. is it easy to use?

Open-Source Availability: Is the tool open-source, which may affect cost, customizability, and support?

For each criterion, a binary decision is made—yes or No. A "Yes" means the tool fulfils the criterion, and the process advances to the next decision point. A "No" triggers a loopback for further action.

LoopBack for Re-evaluation

The process continues if a tool fails to meet any of the criteria. Instead, there is a LoopBackLoopback—the process circles back to the evaluation phase. This is critical as it allows the organisation to reassess its criteria, perhaps adjusting them, or to collect more information about the tools to make a more informed decision.

Choose Tool

If a tool passes all the criteria, it reaches the *Choose Tool* phase. Here, stakeholders can gather and make decisions based on what they have learned about their company's requirements before agreeing on the one that best suits them.

Adopt Tool

After having selected it, the Adopting The Tool stage follows, where integration preparation configurations and training take place. This ensures the effective implementation of the chosen tool across the organizational workflow.

End Event

This is when *End Event* signifies the completed selection process. At this point, an agreement was reached upon choosing one particular instrument, which is presently being integrated into operations, thereby transforming an organisation's DevOps practices into improved ones.

NB: It is important to note that each criterion is not fixed. The criteria can be easily adjusted depending on what matters most to each team or organization. The framework was built so that if a tool does not meet a specific requirement but meets other requirements, it can still be used if the team/organisation can do without the requirement. Both the criteria and the tool can be adjusted according to the organization’s preferences. Hence, the criteria or tool needs to be re-evaluated.

Through this elaborate framework, the BPMN diagram acts as a visual guide showing stakeholders how decisions are made and actions are taken. It represents a scrupulous mechanism that eliminates subjective decision-making and ensures the selected tool aligns with the organization’s strategic and operational objectives. It is a step for building an efficient, cohesive, and effective DevOps environment.

5.3 Summary of Findings

Interviews revealed that the most valuable feature of a DevOps tool is user-friendliness. Other emerging features included the ability to solve user problems, a good interface, ease of integration with other DevOps tools, automation, and security. It was further established that various DevOps tools are used in different organisations based on need.

While various tools are being used, users face challenges, including increased cloud cost, knowledge gap, learning curve, and difficulty integrating the tools with legacy systems and ensuring they perform optimally. Other concerns of the user of DevOps tools include the need for more support and inadequate documentation for the open-source tools compared to the paid/licensed version of these tools. Aside from technical issues with the tools, the DevOps engineer needs help from the organisation's management in accepting new implementations. As a mitigation strategy, the DevOps Engineer ensures that the DevOps tools are being used regularly and ensures more security patches, bridging the gaps in knowledge through individual and team learning.

In selecting DevOps tools for optimal software delivery, research was identified as a selection criterion, with a concentration on the usability/user-friendliness of these tools and cost-effectiveness. Another criterion identified is the team's ability to have the required skills in using the skills and knowing if there are open source tools available. Other criteria mentioned were ease of integration, scalability, and community support.

All the DevOps engineers established that their organisation is gradually adopting cloud-based DevOps tools. Start-ups aim to maintain simplicity and affordability in their tool selection, ensuring cost-effectiveness compared to established organisations, which often utilise complex APIs and sophisticated integrated systems. Another observation is that start-ups prioritise agility, while established organisations prioritise scalability and seamless integration of DevOps tools with existing systems. However, start-ups particularly emphasise cost-effectiveness.

Participants identified the use of built-in calculators provided by tools such as the Azure Calculator or the AWS Calculator as a best practice. These calculators assist in estimating daily, monthly, and yearly expenses and are typically free of charge.

5.4 Recommendations

1. Organisations should clearly define the problem they want to solve to identify the necessary tools.

2. Organisations should find a tool that enables them to build software securely, reliably and cost-friendly

3. Keep the tools as simple as possible.

4. Developers of the tools should prioritise user-friendliness as a key feature.

5. DevOps engineers should develop management and mitigation strategies to optimise available resources.

6. Regular training, workshops, and knowledge-sharing sessions should be organised and invested in to bridge knowledge gaps.

CHAPTER 6: CONCLUSION

6.1 Summary of Study Purpose and Approach

DevOps tools selection process was critically explored in this master's research thesis; this aimed at breaking it down in different contexts for organizations. This journey was precipitated by a gap that exists in structured approaches, which guide companies through the varieties of DevOps tools available. This study has used qualitative research design to undertake expert interviews and then compare these findings with those from other authors, meticulously creating a checklist for selecting appropriate ones.

6.2 Summary of Main Findings

It is worth noting that whereas there are many DevOps tools on the market today that offer various functionalities, ultimately selecting any one among them mostly focuses on several key attributes such as: usability, scalability, affordability and ease of integration. These constituted focal points upon which decisions were made by organization so as to improve operational efficiency and make it consistent with strategic intent.

6.3 Developing a Framework and Integrating BPMN 2.0

In response to the research results, a DevOps Tool Selection Framework has been uniquely devised using the BPMN 2.0 model as its backbone. Besides making the selection process operationalized, this framework contains an iterative loop-back mechanism that ensures tool selection is not a one-time event but rather an ongoing process that caters to changing needs within organizations.

6.4 Theoretical Significance and Practical Importance

In theory, this paper makes its contribution into science concerned with DevOps by providing an analytic lens through which the best tools can be selected or assessed. In practice, it could serve as an instrument for decision-making by firms thereby transforming their software delivery channels. It goes beyond what should be done and right into why organizations must choose any particular tool based on its functional properties versus what they need it for.

6.5 Methodological Reflections

This study therefore reiterates the value of qualitative research methods in uncovering a variation of insights resulting from an analysis method applied, in this case being thematic analysis. Utilizing the thematic analysis, patterns and themes were extracted from qualitative data through coding, hence enriching the study in terms of its depth and practical use.

6.6 Implications for Practice and Academic Context

Professionally, this framework offers a practical and stepwise approach to dealing with the complicated nature of DevOps tools. At the same time, this opens up fresh research avenues for aspiring scholars to examine in future leading to research on how tool selection affects software delivery and organizational agility.

6.7 Recommendations for Future Research

DevOps is a rapidly changing field, where new tools are regularly emerging while others are being updated. Future studies should investigate whether the framework can adapt over time as well as remain relevant amidst constant evolution of technology. The integration of quantitative techniques may also complement qualitative insights thereby providing a more rounded perspective of the selection process.

CHAPTER 7: CRITICAL REFLECTION ON THIS RESEARCH PROCESS.

Critical Reflections on Comparative Analysis of DevOps Tools

When I began this exploration, my aim was to analyze the complex landscape of DevOps tools so as to assist organizations in making the best choice for better software delivery. This critical reflection is an inward look at the methods used, what has been learnt from it, problems encountered and professional growth during this learning process.

Research Initiation and Alignment with Objectives

This research was initiated due to a gap in formalized approaches to choosing devops tools. It is aimed at assisting the organizations not only to rationalize their selection but also develop criteria they can use when assessing the suitability of each tool in its operations and strategic environment. In retrospect, the research objectives seem well aligned with the methodology that follows it up thus providing integrity for this study.

Methodological Appraisal

The fact that a qualitative approach based on indepth interviews complemented by comparative tool analysis was chosen was dictated by nature of questions under investigation. While this method produced rich contextual insights, it introduced challenges pertinent to qualitative research like bias and difficulties in ensuring total saturation with themes. However, upon further re-examination, it seems that while qualitative approach may have been invaluable; using mixed-methods would have led to stronger data triangulation.

Findings Review and Framework Construction

These findings shed light on various aspects such as subtle preferences and decision-making drivers for organizations selecting DevOps tools which served as foundation for developing the BPMN 2.0 based framework. Nonetheless, this reflective thinking causes me question whether this framework can be applied in other environments or not? While the framework was validated within the scope of this study, its performance in varying global contexts and organizational sizes could be further examined.

Challenges Associated With Adapting Research Process

Some challenges arose throughout our participation including finding experts willing to take part through interviewee recruitment all way to assuring relevancy and applicability of developed framework One of the experts couldn't be present due to busy schedule but I was able to meet the minimum I presented initially. In order to tackle such hurdles some flexible ways had to be adopted thus promoting innovativeness and persistence within any research.

Personal Learning and Skill Development

My exposure to experts and analysis of the resultant data has sharpened my critical reasoning abilities, communication aptitudes, and qualitative analytical skills. The process also made me aware of how complex it is to choose appropriate tools and devise systems that will be in line with the organizational culture. This experience underscored the importance of lifelong learning and flexibility—qualities that are indispensable in the constantly changing DevOps landscape.

Professional Growth: Thoughts on Future Application

Obviously, this research has contributed significantly to my professional development. This study has given me insights into strategic implications of DevOps practices which I could use for my future activities in IT management. Additionally, this work has further strengthened my belief in the role of rigorous studies supporting informed decision making within corporate environments.

Broader Impact and Considerations

I am realizing now that through broader reflection, this research may have impacts on DevOps communities at large. However, I understand that technological advancements would necessitate continuous review processes and iterations within the framework. What I have perceived here is that research is not a destination but rather a continuous journey.

Concluding Statements

This critical thinking has given me an opportunity to fully scrutinize the process and end results of research. It identified successes and areas that would benefit from different procedures. At the end, I learnt a lot; it has been a very valuable learning type which is showing how academic researches are related to field technology applications and practice.

CHAPTER 8: REFERENCES

Artec, M., Borovssak, T., Di Nitto, E., Guerreiro, M., & Tamburri, D. (2017). DevOps: Introducing infrastructure as code. In IEE/ACM Engineering Companion (ICSE-C), 497 – 498.

Aljundi, M.K. (2018). Tools and Practices to Enhance DevOps Core Values. School of Business and Management. Lappeeranta University of Technology.

Ashkaya, H.L., Nisarga J. S., Vidya J., & Veena, K. (2005). A Basic Introduction to DevOps Tools. International Journal of Computer Science and Information Technologies, 6(3).

Ataleb, B., & Abo Khalaf, M. (2022). Scalability and Economy of Amazon Lambda, EKS, and ECS. Blekinge Institute of Technology.

Azad, N., & Hyrynsalmi, S. (2023). DevOps Critical Success Factors: A Systematic Literature Review. Information and Software Technology, 157.

Akshaya, H.L., Jagadish, N.S., Vidya, J., & Veena, K. (2015). A Basic Introduction to DevOps Tools. International Journal of Computer Science and Information Technologies, 6(3), 2349-2353.

Balaji, S., & Murugaiyan, M.S. (2020). Waterfall Vs. V-Model Vs. Agile: A Comparative Study on SDLC. International Journal of Information Technology and Business Management, 2(1).

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

Bruns, A., Kornstadt, A., & Wichmann, D. (2009). Web application tests with selenium. IEEE Software, 26(5), 88 – 91.

Blischak, J.D., Davenport, E.R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. PLoS Computational Biology, 12(1), e1004668.

Catherine Crowley, Laura McQuillan, & Conor O’ Brien. (2018). Understanding DevOps: Exploring the Origins, Composition, Merits, and Perils of a DevOps Capability. 4th International Conference on Production Economics and Project Evaluation.

Collins-Sussman, B., Fitzpatrick, & Pilato, C.M. (2004). Version Control with Subversion: Draft Revision 9189. TBA.

Crockett, E. (2023). Top 20 Tools for DevOps. Retrieved from https://www.datamation.com/applications/devops-tools-20-top-tools-for-successful-devops/

Duvall, P.M. (2012). DevOps in the Cloud. Addison-Wesley.

Elbert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software, 33(3), 94 – 100.

Erich, F.M.A., Amirit, C., & Daneva, M. (2017). A Qualitative Study of DevOps Usage in Practice. Journal of Software Evolution and Process.

Ezinne, A. (2023). Containers: what is containerization and container orchestration? Retrieved from https://dev.to/ezinne-anne/containers-what-is-containerization-and-container-orchestration-38pe

Fitzgerald, B., & Stol, K. (2018). The Future of Software Development Methods. In The Routledge Companion to Management Information Systems (pp. 125 – 137). Routledge.

Freeman, E. (2019). DevOps for Dummies. John Wiley and Sons, Inc.

Gray, S. (2020). Agile Software Development Life Cycle. Retrieved from https://serenagray2451.medium.com/agile-software-development-life-cycle-b3ed0f0f7212

Gorkana, M. (2023). DevOps Phases across Software Development Lifecycle. IBM India Pvt Ltd.

Ghantous, G.B., & Gill, A. (2017). DevOps: Concepts, Practices, Tools and Challenges. PACIS 2017 Proceedings, 96.

Haines, T., Idemudia, E.S., & Mahesh, S.R. (2017). The Conceptual Model for Agile Tools and Techniques. American Journal of Management, 17(3), 77 – 88.

Husaini, S. (2015). A Systematic Approach to Reinforce Development and Operations Functions in Delivering an Organizational Program. Procedia Computer Science, 61, 261 – 266.

Hutterman, M. (2012). Beginning DevOps for Developers. In M. Hutterman (Ed.), DevOps for Developers (pp. 1-10). Apress.

Kaiser, S., Haq, S.T., S.A., & Kormaz, T. (2022). Container Technologies for ARM Architecture: A Comprehensive Survey of the State-of-the-Art. IEEE Access, 84853 – 84881.

Kristinsson, R. (2015). Software Development with DevOps. HAGAA-HELIA. University of Applied Sciences.

Kumudavalli, M.V., & Venkatesh, G. (2021). Cloud computing based monolithic to containerization using elastic containers services for phylogenetic analysis. In Third International Conference on Intelligent Communication, Technologies and Virtual Mobile Networks (ICICV) (pp. 1540 – 1543).

Leppanen, T. (2021). Data visualization and monitoring with Grafana and Prometheus. Information and Communications Technology. Turku University of Applied Sciences.

Lwakatare, L.E., Kuvaja, P., & Markku, O. (2016). Relationship of DevOps to Agile, Lean and Continuous Deployment. In PROFES (pp. 1-10). University of Oulu.

Meyer, B. (2014). Agile!: The Good, the Hype and the Ugly. Russia.

Mohammed, I.A. (2015). An Empirical Study of The Importance of DevOps Strategies and Approaches in Information Management Systems. International Journal of Current Science, 5(1), 38 – 52.

Mohanan, R. (2023). GitLab vs. Jenkins: Which is the Best CI/CD Tool in 2023? Retrieved from https://www.spiceworks.com/tech/devops/articles/gitlab-vs-jenkins/

Morris, K. (2016). Infrastructure as Code: Managing Servers in the Cloud. Retrieved from http://oreikky.com/catalog/errata.csp?isbn=9781491924358

Nkhil, P. (2017). Learning Continuous Integration with Jenkins (2nd Edition). Packt.

Rzig, D.E., Hassan, F., Tucker, A., Siala, C., & Houerbi, A. (2022). Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects.

Schwaber, K., & Beedle, M. (2002). Agile Software Development with Scrum. Prentice-Hall.

Senapathi, M., Buchan, J., & Osman, H. (2019). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering (pp. 57 – 67).

Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access, 5, 3909 - 3949.

Sharma, S. (2017). The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise. John Wiley & Sons.

Sharma, S., & Coyne, B. (2015). DevOps for Dummies (2nd Ed.). John Wiley and Sons Inc.

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies: A Conceptual Study. International Journal on Computer Science and Engineering (IJCSE), 4(5).

Singh, V. (2022). Developing a CI/CD Pipeline with Gitlab. Information and Communications Technology. Turku University of Applied Sciences.

Virtanen, J. (2021). Comparing Different CI/CD Pipelines. Hameen University of Applied Sciences.

APPENDICES

APPENDIX A - PROJECT SPECIFICATION & PROJECT PROCESS DOCUMENTATION

PROJECT SPECIFICATION

University of the West of Scotland

School of Computing, Engineering and Physical Sciences

MSc Project Specification

Student name: Oluwapelumi Owolabi

Banner ID: B00947298

Email: b00947298@studentmail.uws.ac.uk

Project being undertaken on part-time or full-time basis: Full Time

MSc Programme/stream: Information Technology

MSc Programme Leader: Costas Iliopoulos

Project Title:

Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios.

Research Question to be answered:

What are the key features of DevOps tools that organizations should prioritize based on different scenarios, and how can these features guide the selection of the best tools for specific organizational needs and contexts?

Outline (overview) and overall aim of project:

This dissertation aims to conduct a comprehensive comparative study of various cloud-based DevOps tools, assessing their features, scalability, compatibility with other tools in terms of integration, and cost. The research will also delve into the effectiveness of these tools in diverse scenarios, such as organizational size and deployment duration. The findings will be used to create a framework for the selection and implementation of cloud-based DevOps tools based on the specific requirements of organizations.

The overall aim of this project is to see if a framework that will aid organizations in making an informed decision about the right tools to used based on the different scenarios, can be developed.

By comparing these tools across key dimensions, the research aims to offer practical guidelines tailored to organizational needs, ensuring optimal tool selection and implementation strategies in various contexts.

Objectives (list of tasks to be undertaken to achieve overall aim of the project and to answer the research question posed):

- Conduct a comprehensive analysis of key dimensions – Features, scalability, cost, and compatibility/integration with other tools, for prominent cloud-based DevOps tools.
- Conduct primary research via interviews with 3-4 users of DevOps tools and obtain other comparisons from secondary research.
- Evaluate the performance of identified tools in diverse scenarios. (Organization sizes, and deployment processes)
- Develop a framework for selecting and implementing cloud-based DevOps tools in organizational settings.

Relationship of proposed project to MSc programme/stream:

My project topic relates closely to my MSc Information Technology program in that it allows me to delve into a specific sector within the IT industry which is broad.

My choice of specializing in cloud-based tools for DevOps allows me to deepen my knowledge and expertise in a specific aspect of IT, which is essential in the diverse and evolving IT landscape.

The project directly aligns with software development and the Technologies for Business Intelligence module, as it involves in-depth exploration and analysis of technologies integral to optimizing information technology processes within an organizational context. By evaluating cloud-based DevOps tools, the study directly addresses the integration of technological solutions for streamlining development and operational workflows, which is a crucial aspect of business intelligence in contemporary IT environments.

Through the investigation of features, scalability, compatibility, and cost of these tools, the project extends the knowledge acquired in the Business Intelligence module to a practical, real-world scenario. Additionally, the study encompasses considerations of organizational needs and size, deployment, and the stage of the organization, thus offering a holistic perspective on implementing technologies for business intelligence.

Indicative reading list (references to be correctly presented) and resources:

Bass, L., Weber, I. and Zhu, L., 2015. DevOps: A software architect's perspective. Addison-Wesley Professional.

Beyer, B., Jones, C., Petoff, J. and Murphy, N.R., 2016. *Site reliability engineering: How Google runs production systems*. " O'Reilly Media, Inc.".

Freeman, E., 2019. DevOps for dummies. John Wiley & Sons. [Online]

Gruver, G. and Mouser, T., 2015. Leading the transformation: Applying agile and DevOps principles at scale. IT Revolution.

Humble, J. and Farley, D., 2010. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education.

Humble, J. and Kim, G., 2018. Accelerate: The science of lean software and DevOps: Building and scaling high-performing technology organizations. IT Revolution

Kim, G., Humble, J., Debois, P., Willis, J. and Forsgren, N., 2021. The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations. IT Revolution.

Kim, W.G., Behr, K. and Spafford, G., 2013. The Phoenix Project: a novel about IT, DevOps, and helping your business. Oregon: IT Revolution Press Portland, 172. [Online]

Matthias, K. and Kane, S.P., 2015. *Docker: Up & Running: Shipping Reliable Containers in Production*. " O'Reilly Media, Inc.".

Morris, K., 2016. Infrastructure as code: managing servers in the cloud. " O'Reilly Media, Inc.". [Online]

Newman, S. 2015. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.

Schwartz, M., 2016. *The art of business value*. It Revolution.

Sharma, S., 2017. *The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise*. John Wiley & Sons.

Shroff, G., 2010. Enterprise cloud computing: technology, architecture, applications. Cambridge university press.

Marking scheme:

Introduction: 5%

Literature review: 15%

Methodology: 15%

Data Analysis: 20%

Evaluation and Development of framework: 25%

Conclusions: 10%

Critical self-evaluation: 10%

Supervisor:

Miriam Birch

Moderator:

Santiago Matalonga

Programme Leader:

Costas Iliopoulos

Date specification submitted:

25/01/2024

Please complete the ‘ethics’ form below for all projects.

School of Computing, Engineering and Physical Sciences

MSc PROJECT – REQUIREMENT FOR ETHICAL APPROVAL

SECTION 1: TO BE COMPLETED BY THE STUDENT

Does your proposed research involve: research with human subjects (including requirements gathering and product/software testing), access to company documents/records, questionnaires, surveys, focus groups and/or other interview techniques? Does your research entail any process which requires ethical approval? (please enter √ in the appropriate box)

YES	√	**You must submit an application for approval to the Ethics Review Manager**
NO		You do not need to submit an application to the Ethics Review Manager

Name of Student (Print name): Oluwapelumi Owolabi

Signature: Pelumi Owolabi

Date: 19/01/2024

SECTION 2: TO BE COMPLETED BY THE PROJECT SUPERVISOR

I understand that the above project requires ethical approval (*please delete as appropriate).

Supervisor (print name): Dr Miriam Birch

Signature: Miriam Birch

Date: 26/01/2024

IMPORTANT: please note that by signing this form all signatories are confirming that any potential ethical issues have been considered and, where necessary, an application for ethical approval has been/will be made via the Ethical Review Manager software.

Any project requiring ethical approval but which has not been given approval will not be accepted for marking.

Ethical approval cannot be sought in retrospect.

PROJECT PROCESS DOCUMENTATION

MSc PROJECT (COMP11024)

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 1 **Date/Time: 17/01/2024 10 am**

Agenda for meeting:

What I need to do for the project

- Completing the Project spec form
- Applying for ethical approval
- Completing the process documentation form
- Managing your project / project plan

Discussion of agenda items:

- What I need to do for the project

  To decide on a project topic and start draw out a plan with deadlines on them using the Gantt Chart. To also ensure that my project has elements of research in them, both literature and analysis. To also have a marking scheme.

- Completing the Project spec form

  The project specification form states in specifics what I will be doing in my project, what it entails, how I will achieve it, the justification for the choosing that topic, research methods, if I'll need ethical approval and marking scheme.

- Applying for ethical approval

  If my research will include conducting a survey, then I’ll need to get ethical approval from the University management and I have t do it fast so I can get a response early as well.

- Completing the process documentation form

  This is necessary as it helps keep track of each meeting with my supervisor. I can always make reference to what has been discussed.

- Managing your project / project plan

  Preparing a plan to know where I should be in each week and to be conscious of the deadlines.

Summary of agreed action plan:

To complete the project spec form

To look at the PowerPoint and samples to get help with what should be in each section.

To ask myself - What is the problem / opportunity to be addressed? Why is my project worth doing? What am I going to do and why.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 2 **Date/Time: 24/01/2024 10 am**

Agenda for meeting:

- Review of the details of the project specification by student:

    1. Research questions
    2. Objectives
    3. References
    4. Methodology
    5. Marking scheme

- Supervisor’s review of project specification.
- Next steps

Discussion of agenda items:

- The ways by which I want to achieve my project tasks: This is through interview and secondary research.
- Discussion of project output and how I’m going to achieve it.
- Research questions: Guidance was given to me as I’ll have to rephrase the research question and the project title because the project title is not exactly specific as against the intention behind my project.
- Methodology - I should confirm if an interview is possible and state it in clear terms on my project specification. Also, I shouldn’t write anything I won’t be able to achieve.
- I am also to state that I will be building a framework that will help companies determine the right tools to use depending on their situation.

- Marking Scheme: I should review the percentages given to evaluation, presentation, conclusion and recommendations and think of what to write in those sections.
- References: More references should be added and be done in alphabetical order.

Summary of agreed action plan:

- Rework project specification: Be specific in areas where ambiguous, change choice of words.
- Fill out the Ethical approval form on Aula
- To go and confirm if the methods to achieve my research are feasible. (Interview, Case Study and Lab work)
- Send the revised form to Dr Daune West, copying Miriam and Santiago.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 3 **Date/Time: 31/01/2024 10 am**

Agenda for the meeting:

- Project specification
- Review of ethical review form.
- Discussion of research and Design and method results
- Next steps

Discussion of agenda items:

- **Project specification:** The project specification was filled last week, sent to Dr West and also submitted on Aula.

- **Review of ethical review form:** I filled the ethical review form for approval by the committee. Before doing that, I needed approval from my supervisor. She made some comments on the form on what I should adjust and areas I missed. She asked me to go back to refill the form and then resubmit to her for approval.

- **Discussion of research and Design and method results:** The feedback on my results for the Research and Design Methods module was discussed and I was asked by my supervisor to use the feedback from the result to better influence my literature review.

- **Next steps:** Review the ethical approval form and start my literature review and background research.

Summary of agreed action plan:

- Edit some parts of the form: Be specific in areas where ambiguous, change choice of words. Also, to fill out a part I missed.

- Start my literature view and background research.

- Fill out documentation for today's meeting.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 4 **Date/Time: 07/02/2024 10 am**

Agenda for meeting:

- **Review of Interim Report**

Discussion of agenda items:

- We went through my interim report as we're halfway through week 7. My supervisor asked me to link my literature review to my project. She also asked me to state what I'm doing, why I am doing it, why I am doing it the way I am doing it, the relevance of the previous literature/studies to my project and how I intend to achieve my project as well.

Summary of agreed action plan

- To go back to my document and apply the changes suggested.
- To submit interim report.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 5 **Date/Time: 21/02/2024 10 am**

Agenda for meeting:

Research Methodology

Discussion of agenda items:

- Prior to this meeting, the researcher had already submitted the interim report to Aula and his supervisor.

- The research methodology was discussed, and the researcher is at the point where ethical approval is needed in his research to move forward. His supervisor advised that he goes ahead with the pilot studies and secondary studies.

- Any new discoveries or recommendations will be implemented on the interview questions.

Summary of agreed action plan:

- Researcher should conduct pilot studies and carry on with project.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 6 **Date/Time: 28/03/2024 10 am**

Agenda for meeting:

- Project update

Discussion of agenda items:

- At this point, I had already conducted the pilot studies on one person in the industry field. Eventually, I had to convert the data collected from the pilot study to a primary data since the professional from the pilot study won't be available in the future for the main interview.
- New adjustments from the pilot studies were applied to the interview questions going forward.

Summary of agreed action plan:

- Schedule meeting times with industry professionals for data collection ahead of ethical approval as I was hinted that I might get ethical approval anytime soon.
- Supervisor advised on steps to Improve methodology, and that's to ensure that I am specifically clear and document all my findings.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 7 **Date/Time: 13/03/2024 10 am**

Agenda for meeting:

- Project update

Discussion of agenda items:

- At this point, I have gotten ethical approval from the ethics committee, and I can go ahead with the data collection from my interviews.
- I am awaiting feedback from interim report. I received it just some minutes to today's meeting, so I am yet to go through it.
- We discussed the progress of my project, and I was advised to keep working on my project.

Summary of agreed action plan:

- Proceed with interviews.
- Apply feedback from interim report on dissertation.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi	**Supervisor: Miriam Birch**
Meeting Number: 8	**Date/Time: 20/03/2024 10 am**

Agenda for meeting:

- Update on my project.
- Interim report discussion

Discussion of agenda items:

- We discussed on the feedback from my supervisor and moderator on my interim report. I was advised to apply the necessary changes on made on my interim report.
- Questions were also asked about the progress of my project. The main area of concern for correction is to make sure I don't speak in generic terms, but to be rather specific. Also, that I should find ways to link my literature review and methodology to my research objectives.

- The project was taking longer than expected as it was quite difficult getting a hold of one of the industry professionals.

Summary of agreed action plan:

- Apply corrections from interim report feedback on my project.
- Take next steps to my project: Continuing with the interviews.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 9 **Date/Time: 27/03/2024 10 am**

Agenda for meeting:

- Interim report feedback
- Update on my project

Discussion of agenda items:

- We discussed on the feedback from my supervisor and moderator on my interim report. I was advised to apply the necessary changes on made on my interim report.
- Questions were also asked about the progress of my project. The main area of concern for correction is to make sure I don’t speak in generic terms, but to be rather specific. Also, that I should find ways to link my literature review and methodology to my research objectives.

Summary of agreed action plan:

- Apply corrections from interim report feedback on my project
- Take next steps to my project: Continuing with the interviews.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 10 **Date/Time: 27/03/2024 10 am**

Agenda for meeting:

- Update on the project

Discussion of agenda items:

- At this point, I have just completed the third last interview required for my data collection. There's supposed to be a fourth but the meeting couldn't hold due to a busy schedule. I was already far behind on my project but I could meet up with the timeline if I acted fast.

Summary of agreed action plan:

- Start analysing data collected from interviews.
- Submit final report draft after analysis

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 11 **Date/Time: 10/04/2024 10 am**

Agenda for meeting:

- Update on the project

Discussion of agenda items:

- Prior to the meeting, I sent my supervisor a copy of my project. We discussed on what I have done so far and in the report I sent, I had done up to chapter 4. My chapter 5 wasn't done, which included my framework. The document I sent to her included my introduction up till the draft of my chapter 4 with the plan to finish it up during the week. After, the meeting, my supervisor later sent me feedback on the draft and I applied the changes suggested.

Summary of agreed action plan:

- Apply the changes suggested – Emphasis on making my literature review cohesive as it the sections weren't flowing into each other.

- Develop my framework and complete my chapter 4, 5 & 6.

PROJECT PROCESS DOCUMENTATION

Student: Oluwapelumi Owolabi **Supervisor: Miriam Birch**

Meeting Number: 12 **Date/Time: 17/04/2024 10 am**

Agenda for meeting:

- Review of Final report

Discussion of agenda items:

- We went through my final report briefly, with special attention on my framework. Some questions were asked by my supervisor on understanding my framework and I was advised to make complex areas clear.

- We discussed my next steps which is to make my framework clearer and easy to read, the documents needed for submission and what should be included in my appendices. I was also advised to ensure that my project is in line with my marking guide.

Summary of agreed action plan:

- Finish up on my final report and submit
- Prepare for my Viva with my supervisor and moderator on the 25th of April.

APPENDIX B - INTERVIEW QUESTIONS & RESPONSES, AND CONSENT FORMS

Introduction

1. Can you briefly give an overview of your name, organization name and function, role, and responsibilities?

Background:

2. How would you describe your organization's current software development and operations practices?

DevOps Tool Usage:

3. What specific cloud-based DevOps tools does your organization use?
4. What criteria did your organization use to select the DevOps tools being used?

Features

5. What features of these tools do you find most valuable as an individual and a team?

6. How would you rate the usability of these tools in your daily workflow, considering factors like user interface design, integration ease, and any other factors you may think about?

Scalability and Compatibility

7. How has your organization tested the DevOps tools concerning scalability, and can you share any experiences related to their scalability?

8. Have you encountered compatibility issues when integrating these tools with existing systems or processes?

Challenges and Pain Points:

9. What challenges or pain points have you and your team faced while using cloud-based DevOps tools?

10. How has your organization addressed or mitigated these challenges?

Effectiveness in Different Scenarios:

11. In your experience, how effective have these tools been for small teams compared to larger ones?

12. Can you share examples of how the DevOps tools have performed in simple versus complex deployment pipelines?

Startups vs. Established Enterprises:

13. Have you worked in a big or small organization? Is your organization a startup, or has it been long established?

14 From your perspective, how do the needs and preferences for DevOps tools differ between startups and established enterprises?

Cost Considerations:

15. How does your organization evaluate the cost-effectiveness of the selected DevOps tools?

16. Have you observed any cost-related challenges or unexpected expenses associated with these tools?

Recommendations and Best Practices:

17. Based on your experience, what recommendations would you provide to organizations looking to adopt cloud-based DevOps tools?

18. Are there any best practices you would recommend for selecting, implementing, and optimizing these tools?

Future Trends and Adaptability:

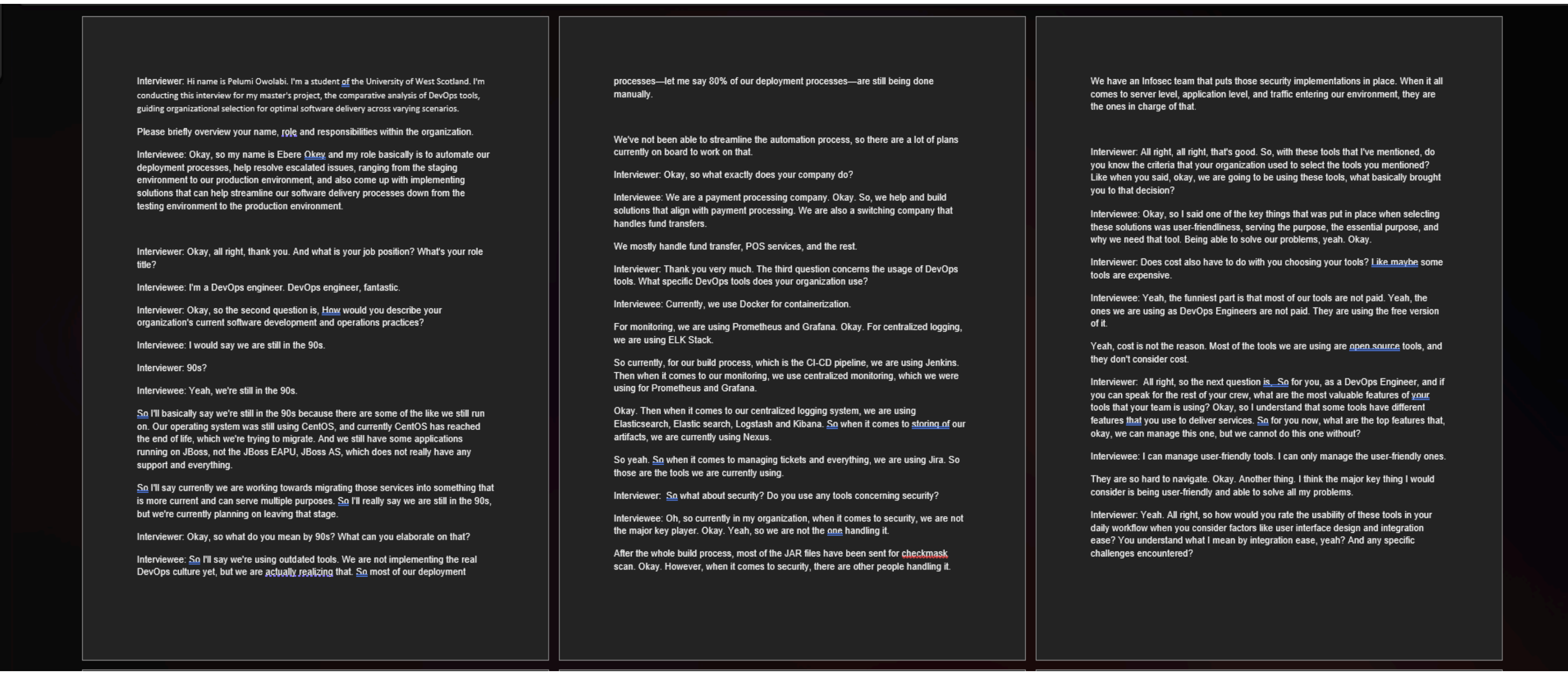
19. How do you see the future trends in cloud-based DevOps tools evolving, and how is your organization preparing for these changes?

20. How adaptable are your current DevOps practices and tools to emerging technologies and methodologies?

Closing:

21. Do you want to share additional information or insights regarding your experience with cloud-based DevOps tools?

RESPONDENT 1



Interviewer: Hi name is Pelumi Owolabi. I'm a student of the University of West Scotland. I'm conducting this interview for my master's project, the comparative analysis of DevOps tools, guiding organizational selection for optimal software delivery across varying scenarios.

Please briefly overview your name, role and responsibilities within the organization.

Interviewee: Okay, so my name is Ebere Okey and my role basically is to automate our deployment processes, help resolve escalated issues, ranging from the staging environment to our production environment, and also come up with implementing solutions that can help streamline our software delivery processes down from the testing environment to the production environment.

Interviewer: Okay, all right, thank you. And what is your job position? What's your role title?

Interviewee: I'm a DevOps engineer. DevOps engineer, fantastic.

Interviewer: Okay, so the second question is, How would you describe your organization's current software development and operations practices?

Interviewee: I would say we are still in the 90s.

Interviewer: 90s?

Interviewee: Yeah, we're still in the 90s.

So I'll basically say we're still in the 90s because there are some of the like we still run on. Our operating system was still using CentOS, and currently CentOS has reached the end of life, which we're trying to migrate. And we still have some applications running on JBoss, not the JBoss EAPU, JBoss AS, which does not really have any support and everything.

So I'll say currently we are working towards migrating those services into something that is more current and can serve multiple purposes. So I'll really say we are still in the 90s, but we're currently planning on leaving that stage.

Interviewer: Okay, so what do you mean by 90s? What can you elaborate on that?

Interviewee: So I'll say we're using outdated tools. We are not implementing the real DevOps culture yet, but we are actually realizing that. So most of our deployment processes—let me say 80% of our deployment processes—are still being done manually.

We've not been able to streamline the automation process, so there are a lot of plans currently on board to work on that.

Interviewer: Okay, so what exactly does your company do?

Interviewee: We are a payment processing company. Okay. So, we help and build solutions that align with payment processing. We are also a switching company that handles fund transfers.

We mostly handle fund transfer, POS services, and the rest.

Interviewer: Thank you very much. The third question concerns the usage of DevOps tools. What specific DevOps tools does your organization use?

Interviewee: Currently, we use Docker for containerization.

For monitoring, we are using Prometheus and Grafana. Okay. For centralized logging, we are using ELK Stack.

So currently, for our build process, which is the CI-CD pipeline, we are using Jenkins. Then when it comes to our monitoring, we use centralized monitoring, which we were using for Prometheus and Grafana.

Okay. Then when it comes to our centralized logging system, we are using Elasticsearch, Elastic search, Logstash and Kibana. So when it comes to storing of our artifacts, we are currently using Nexus.

So yeah. So when it comes to managing tickets and everything, we are using Jira. So those are the tools we are currently using.

Interviewer: So what about security? Do you use any tools concerning security?

Interviewee: Oh, so currently in my organization, when it comes to security, we are not the major key player. Okay. Yeah, so we are not the one handling it.

After the whole build process, most of the JAR files have been sent for checkmask scan. Okay. However, when it comes to security, there are other people handling it.

We have an Infosec team that puts those security implementations in place. When it all comes to server level, application level, and traffic entering our environment, they are the ones in charge of that.

Interviewer: All right, all right, that's good. So, with these tools that I've mentioned, do you know the criteria that your organization used to select the tools you mentioned? Like when you said, okay, we are going to be using these tools, what basically brought you to that decision?

Interviewee: Okay, so I said one of the key things that was put in place when selecting these solutions was user-friendliness, serving the purpose, the essential purpose, and why we need that tool. Being able to solve our problems, yeah. Okay.

Interviewer: Does cost also have to do with you choosing your tools? Like maybe some tools are expensive.

Interviewee: Yeah, the funniest part is that most of our tools are not paid. Yeah, the ones we are using as DevOps Engineers are not paid. They are using the free version of it.

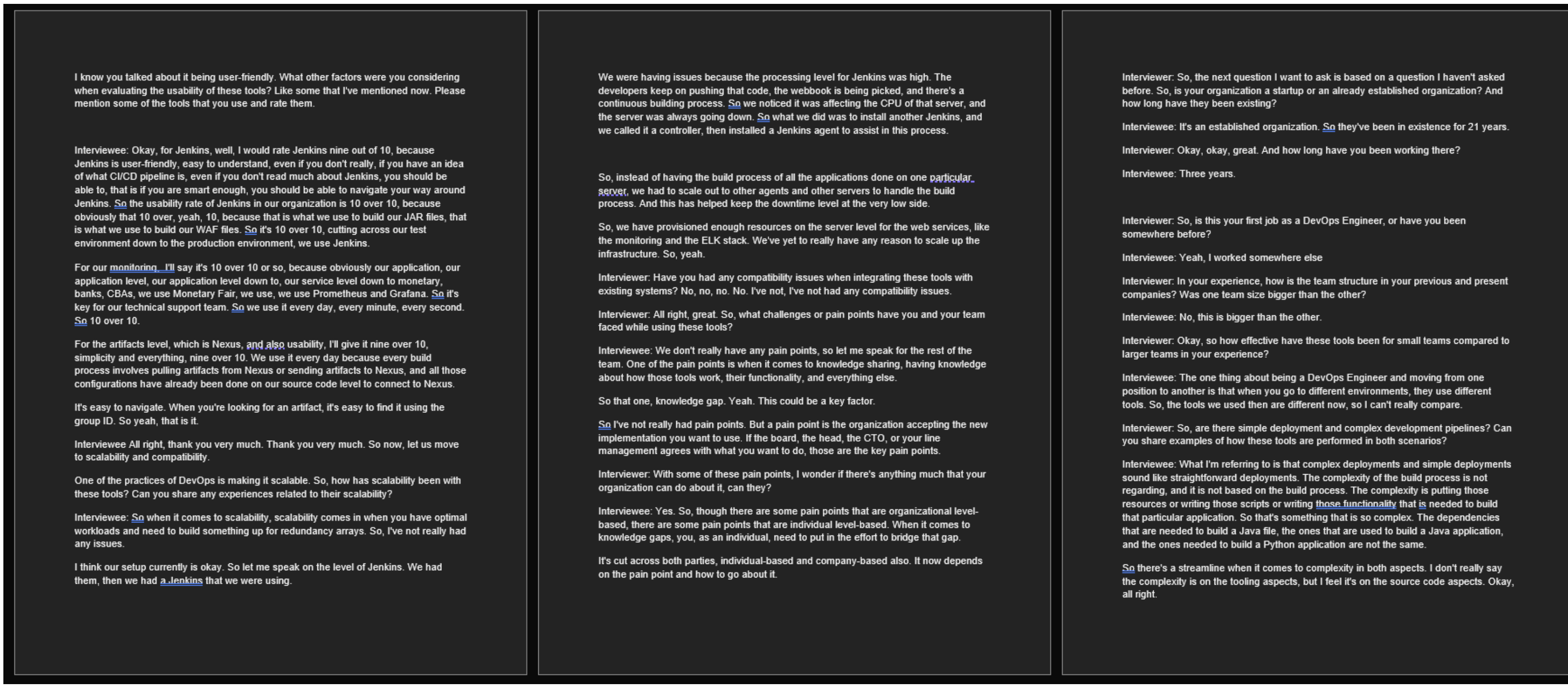
Yeah, cost is not the reason. Most of the tools we are using are open source tools, and they don't consider cost.

Interviewer: All right, so the next question is. So for you, as a DevOps Engineer, and if you can speak for the rest of your crew, what are the most valuable features of your tools that your team is using? Okay, so I understand that some tools have different features that you use to deliver services. So for you now, what are the top features that, okay, we can manage this one, but we cannot do this one without?

Interviewee: I can manage user-friendly tools. I can only manage the user-friendly ones.

They are so hard to navigate. Okay. Another thing. I think the major key thing I would consider is being user-friendly and able to solve all my problems.

Interviewer: Yeah. All right, so how would you rate the usability of these tools in your daily workflow when you consider factors like user interface design and integration ease? You understand what I mean by integration ease, yeah? And any specific challenges encountered?



I know you talked about it being user-friendly. What other factors were you considering when evaluating the usability of these tools? Like some that I've mentioned now. Please mention some of the tools that you use and rate them.

Interviewee: Okay, for Jenkins, well, I would rate Jenkins nine out of 10, because Jenkins is user-friendly, easy to understand, even if you don't really, if you have an idea of what CI/CD pipeline is, even if you don't read much about Jenkins, you should be able to, that is if you are smart enough, you should be able to navigate your way around Jenkins. So the usability rate of Jenkins in our organization is 10 over 10, because obviously that 10 over, yeah, 10, because that is what we use to build our JAR files, that is what we use to build our WAF files. So it's 10 over 10, cutting across our test environment down to the production environment, we use Jenkins.

For our monitoring, I'll say it's 10 over 10 or so, because obviously our application, our application level, our application level down to, our service level down to monetary, banks, CBAs, we use Monetary Fair, we use, we use Prometheus and Grafana. So it's key for our technical support team. So we use it every day, every minute, every second. So 10 over 10.

For the artifacts level, which is Nexus, and also usability, I'll give it nine over 10, simplicity and everything, nine over 10. We use it every day because every build process involves pulling artifacts from Nexus or sending artifacts to Nexus, and all those configurations have already been done on our source code level to connect to Nexus.

It's easy to navigate. When you're looking for an artifact, it's easy to find it using the group ID. So yeah, that is it.

Interviewee All right, thank you very much. Thank you very much. So now, let us move to scalability and compatibility.

One of the practices of DevOps is making it scalable. So, how has scalability been with these tools? Can you share any experiences related to their scalability?

Interviewee: So when it comes to scalability, scalability comes in when you have optimal workloads and need to build something up for redundancy arrays. So, I've not really had any issues.

I think our setup currently is okay. So let me speak on the level of Jenkins. We had them, then we had a Jenkins that we were using.

We were having issues because the processing level for Jenkins was high. The developers keep on pushing that code, the webbook is being picked, and there's a continuous building process. So we noticed it was affecting the CPU of that server, and the server was always going down. So what we did was to install another Jenkins, and we called it a controller, then installed a Jenkins agent to assist in this process.

So, instead of having the build process of all the applications done on one particular server, we had to scale out to other agents and other servers to handle the build process. And this has helped keep the downtime level at the very low side.

So, we have provisioned enough resources on the server level for the web services, like the monitoring and the ELK stack. We've yet to really have any reason to scale up the infrastructure. So, yeah.

Interviewer: Have you had any compatibility issues when integrating these tools with existing systems? No, no, no. No. I've not, I've not had any compatibility issues.

Interviewer: All right, great. So, what challenges or pain points have you and your team faced while using these tools?

Interviewee: We don't really have any pain points, so let me speak for the rest of the team. One of the pain points is when it comes to knowledge sharing, having knowledge about how those tools work, their functionality, and everything else.

So that one, knowledge gap. Yeah. This could be a key factor.

So I've not really had pain points. But a pain point is the organization accepting the new implementation you want to use. If the board, the head, the CTO, or your line management agrees with what you want to do, those are the key pain points.

Interviewer: With some of these pain points, I wonder if there's anything much that your organization can do about it, can they?

Interviewee: Yes. So, though there are some pain points that are organizational level-based, there are some pain points that are individual level-based. When it comes to knowledge gaps, you, as an individual, need to put in the effort to bridge that gap.

It's cut across both parties, individual-based and company-based also. It now depends on the pain point and how to go about it.

Interviewer: So, the next question I want to ask is based on a question I haven't asked before. So, is your organization a startup or an already established organization? And how long have they been existing?

Interviewee: It's an established organization. So they've been in existence for 21 years.

Interviewer: Okay, okay, great. And how long have you been working there?

Interviewee: Three years.

Interviewer: So, is this your first job as a DevOps Engineer, or have you been somewhere before?

Interviewee: Yeah, I worked somewhere else

Interviewer: In your experience, how is the team structure in your previous and present companies? Was one team size bigger than the other?

Interviewee: No, this is bigger than the other.

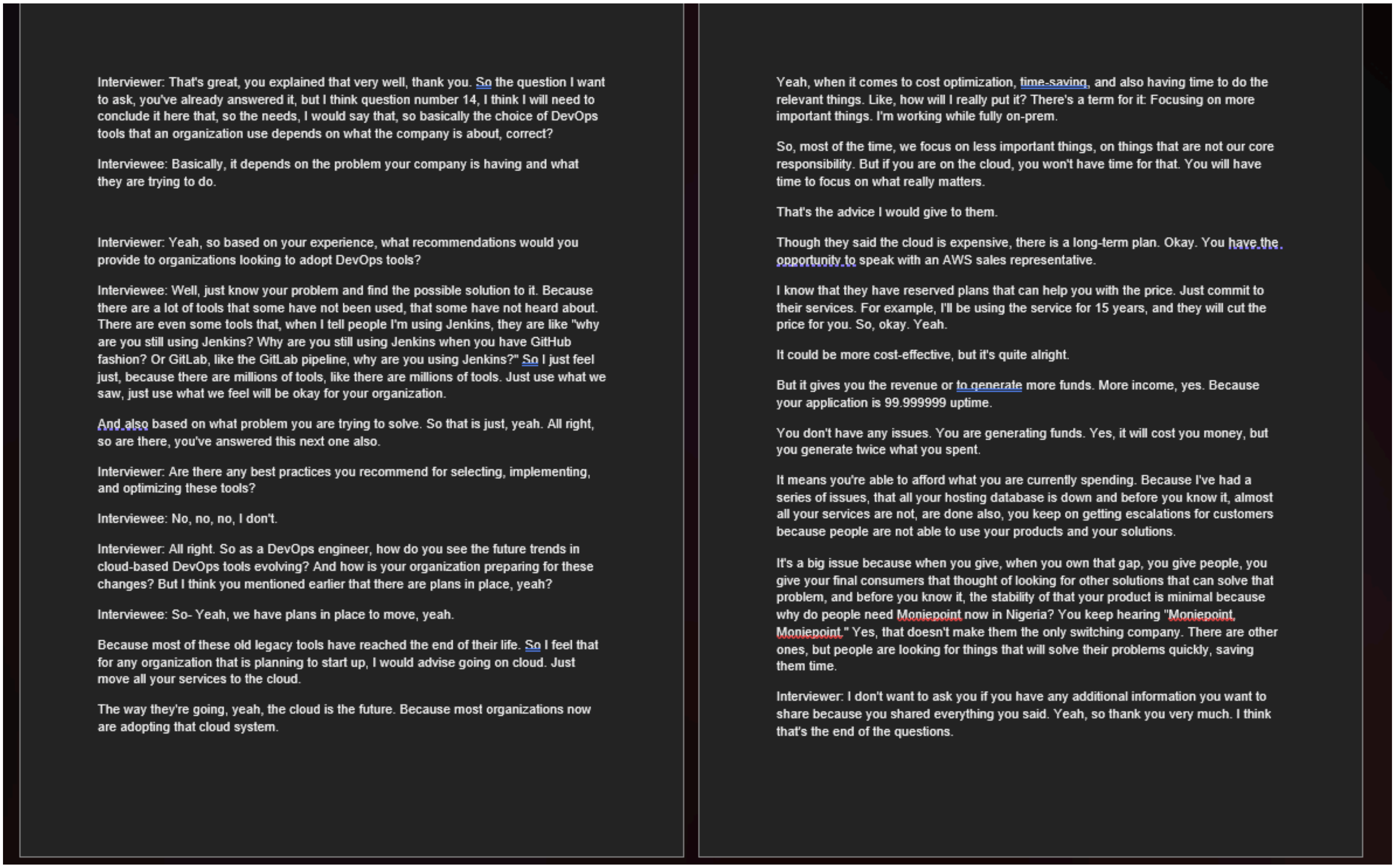
Interviewer: Okay, so how effective have these tools been for small teams compared to larger teams in your experience?

Interviewee: The one thing about being a DevOps Engineer and moving from one position to another is that when you go to different environments, they use different tools. So, the tools we used then are different now, so I can't really compare.

Interviewer: So, are there simple deployment and complex development pipelines? Can you share examples of how these tools are performed in both scenarios?

Interviewee: What I'm referring to is that complex deployments and simple deployments sound like straightforward deployments. The complexity of the build process is not regarding, and it is not based on the build process. The complexity is putting those resources or writing those scripts or writing those functionality that is needed to build that particular application. So that's something that is so complex. The dependencies that are needed to build a Java file, the ones that are used to build a Java application, and the ones needed to build a Python application are not the same.

So there's a streamline when it comes to complexity in both aspects. I don't really say the complexity is on the tooling aspects, but I feel it's on the source code aspects. Okay, all right.



Interviewer: That's great, you explained that very well, thank you. So the question I want to ask, you've already answered it, but I think question number 14, I think I will need to conclude it here that, so the needs, I would say that, so basically the choice of DevOps tools that an organization use depends on what the company is about, correct?

Interviewee: Basically, it depends on the problem your company is having and what they are trying to do.

Interviewer: Yeah, so based on your experience, what recommendations would you provide to organizations looking to adopt DevOps tools?

Interviewee: Well, just know your problem and find the possible solution to it. Because there are a lot of tools that some have not been used, that some have not heard about. There are even some tools that, when I tell people I'm using Jenkins, they are like "why are you still using Jenkins? Why are you still using Jenkins when you have GitHub fashion? Or GitLab, like the GitLab pipeline, why are you using Jenkins?" So I just feel just, because there are millions of tools, like there are millions of tools. Just use what we saw, just use what we feel will be okay for your organization.

And also based on what problem you are trying to solve. So that is just, yeah. All right, so are there, you've answered this next one also.

Interviewer: Are there any best practices you recommend for selecting, implementing, and optimizing these tools?

Interviewee: No, no, no, I don't.

Interviewer: All right. So as a DevOps engineer, how do you see the future trends in cloud-based DevOps tools evolving? And how is your organization preparing for these changes? But I think you mentioned earlier that there are plans in place, yeah?

Interviewee: So- Yeah, we have plans in place to move, yeah.

Because most of these old legacy tools have reached the end of their life. So I feel that for any organization that is planning to start up, I would advise going on cloud. Just move all your services to the cloud.

The way they're going, yeah, the cloud is the future. Because most organizations now are adopting that cloud system.

Yeah, when it comes to cost optimization, time-saving, and also having time to do the relevant things. Like, how will I really put it? There's a term for it: Focusing on more important things. I'm working while fully on-prem.

So, most of the time, we focus on less important things, on things that are not our core responsibility. But if you are on the cloud, you won't have time for that. You will have time to focus on what really matters.

That's the advice I would give to them.

Though they said the cloud is expensive, there is a long-term plan. Okay. You have the opportunity to speak with an AWS sales representative.

I know that they have reserved plans that can help you with the price. Just commit to their services. For example, I'll be using the service for 15 years, and they will cut the price for you. So, okay. Yeah.

It could be more cost-effective, but it's quite alright.

But it gives you the revenue or to generate more funds. More income, yes. Because your application is 99.999999 uptime.

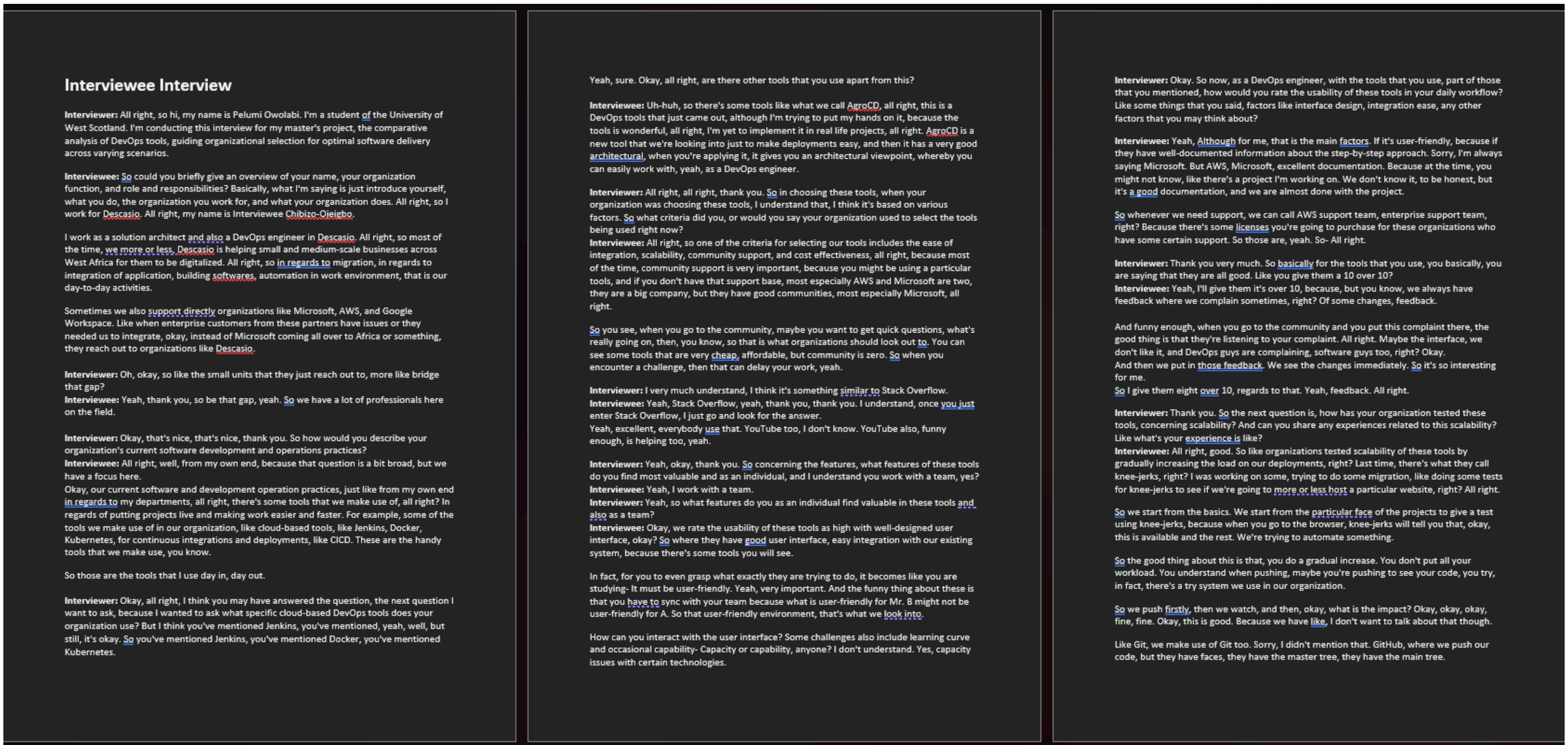
You don't have any issues. You are generating funds. Yes, it will cost you money, but you generate twice what you spent.

It means you're able to afford what you are currently spending. Because I've had a series of issues, that all your hosting database is down and before you know it, almost all your services are not, are done also, you keep on getting escalations for customers because people are not able to use your products and your solutions.

It's a big issue because when you give, when you own that gap, you give people, you give your final consumers that thought of looking for other solutions that can solve that problem, and before you know it, the stability of that your product is minimal because why do people need Moniepoint now in Nigeria? You keep hearing "Moniepoint, Moniepoint." Yes, that doesn't make them the only switching company. There are other ones, but people are looking for things that will solve their problems quickly, saving them time.

Interviewer: I don't want to ask you if you have any additional information you want to share because you shared everything you said. Yeah, so thank you very much. I think that's the end of the questions.

RESPONDENT 2



Interviewee Interview

Interviewer: All right, so hi, my name is Pelumi Owolabi. I'm a student of the University of West Scotland. I'm conducting this interview for my master's project, the comparative analysis of DevOps tools, guiding organizational selection for optimal software delivery across varying scenarios.

Interviewee: So could you briefly give an overview of your name, your organization function, and role and responsibilities? Basically, what I'm saying is just introduce yourself, what you do, the organization you work for, and what your organization does. All right, so I work for Descasio. All right, my name is Interviewee Chibizo-Ojeigbo.

I work as a solution architect and also a DevOps engineer in Descasio. All right, so most of the time, we more or less, Descasio is helping small and medium-scale businesses across West Africa for them to be digitalized. All right, so in regards to migration, in regards to integration of application, building softwares, automation in work environment, that is our day-to-day activities.

Sometimes we also support directly organizations like Microsoft, AWS, and Google Workspace. Like when enterprise customers from these partners have issues or they needed us to integrate, okay, instead of Microsoft coming all over to Africa or something, they reach out to organizations like Descasio.

Interviewer: Oh, okay, so like the small units that they just reach out to, more like bridge that gap?
Interviewee: Yeah, thank you, so be that gap, yeah. So we have a lot of professionals here on the field.

Interviewer: Okay, that's nice, that's nice, thank you. So how would you describe your organization's current software development and operations practices?
Interviewee: All right, well, from my own end, because that question is a bit broad, but we have a focus here.
Okay, our current software and development operation practices, just like from my own end in regards to my departments, all right, there's some tools that we make use of, all right? In regards of putting projects live and making work easier and faster. For example, some of the tools we make use of in our organization, like cloud-based tools, like Jenkins, Docker, Kubernetes, for continuous integrations and deployments, like CICD. These are the handy tools that we make use, you know.

So those are the tools that I use day in, day out.

Interviewer: Okay, all right, I think you may have answered the question, the next question I want to ask, because I wanted to ask what specific cloud-based DevOps tools does your organization use? But I think you've mentioned Jenkins, you've mentioned, yeah, well, but still, it's okay. So you've mentioned Jenkins, you've mentioned Docker, you've mentioned Kubernetes.

Yeah, sure. Okay, all right, are there other tools that you use apart from this?

Interviewee: Uh-huh, so there's some tools like what we call AgroCD, all right, this is a DevOps tools that just came out, although I'm trying to put my hands on it, because the tools is wonderful, all right, I'm yet to implement it in real life projects, all right. AgroCD is a new tool that we're looking into just to make deployments easy, and then it has a very good architectural, when you're applying it, it gives you an architectural viewpoint, whereby you can easily work with, yeah, as a DevOps engineer.

Interviewer: All right, all right, thank you. So in choosing these tools, when your organization was choosing these tools, I understand that, I think it's based on various factors. So what criteria did you, or would you say your organization used to select the tools being used right now?
Interviewee: All right, so one of the criteria for selecting our tools includes the ease of integration, scalability, community support, and cost effectiveness, all right, because most of the time, community support is very important, because you might be using a particular tools, and if you don't have that support base, most especially AWS and Microsoft are two, they are a big company, but they have good communities, most especially Microsoft, all right.

So you see, when you go to the community, maybe you want to get quick questions, what's really going on, then, you know, so that is what organizations should look out to. You can see some tools that are very cheap, affordable, but community is zero. So when you encounter a challenge, then that can delay your work, yeah.

Interviewer: I very much understand, I think it's something similar to Stack Overflow.
Interviewee: Yeah, Stack Overflow, yeah, thank you, thank you. I understand, once you just enter Stack Overflow, I just go and look for the answer.
Yeah, excellent, everybody use that. YouTube too, I don't know. YouTube also, funny enough, is helping too, yeah.

Interviewer: Yeah, okay, thank you. So concerning the features, what features of these tools do you find most valuable and as an individual, and I understand you work with a team, yes?
Interviewee: Yeah, I work with a team.
Interviewer: Yeah, so what features do you as an individual find valuable in these tools and also as a team?
Interviewee: Okay, we rate the usability of these tools as high with well-designed user interface, okay? So where they have good user interface, easy integration with our existing system, because there's some tools you will see.

In fact, for you to even grasp what exactly they are trying to do, it becomes like you are studying- It must be user-friendly. Yeah, very important. And the funny thing about these is that you have to sync with your team because what is user-friendly for Mr. B might not be user-friendly for A. So that user-friendly environment, that's what we look into.

How can you interact with the user interface? Some challenges also include learning curve and occasional capability- Capacity or capability, anyone? I don't understand. Yes, capacity issues with certain technologies.

Interviewer: Okay. So now, as a DevOps engineer, with the tools that you use, part of those that you mentioned, how would you rate the usability of these tools in your daily workflow? Like some things that you said, factors like interface design, integration ease, any other factors that you may think about?

Interviewee: Yeah, Although for me, that is the main factors. If it's user-friendly, because if they have well-documented information about the step-by-step approach. Sorry, I'm always saying Microsoft. But AWS, Microsoft, excellent documentation. Because at the time, you might not know, like there's a project I'm working on. We don't know it, to be honest, but it's a good documentation, and we are almost done with the project.

So whenever we need support, we can call AWS support team, enterprise support team, right? Because there's some licenses you're going to purchase for these organizations who have some certain support. So those are, yeah. So- All right.

Interviewer: Thank you very much. So basically for the tools that you use, you basically, you are saying that they are all good. Like you give them a 10 over 10?
Interviewee: Yeah, I'll give them it's over 10, because, but you know, we always have feedback where we complain sometimes, right? Of some changes, feedback.

And funny enough, when you go to the community and you put this complaint there, the good thing is that they're listening to your complaint. All right. Maybe the interface, we don't like it, and DevOps guys are complaining, software guys too, right? Okay.
And then we put in those feedback. We see the changes immediately. So it's so interesting for me.
So I give them eight over 10, regards to that. Yeah, feedback. All right.

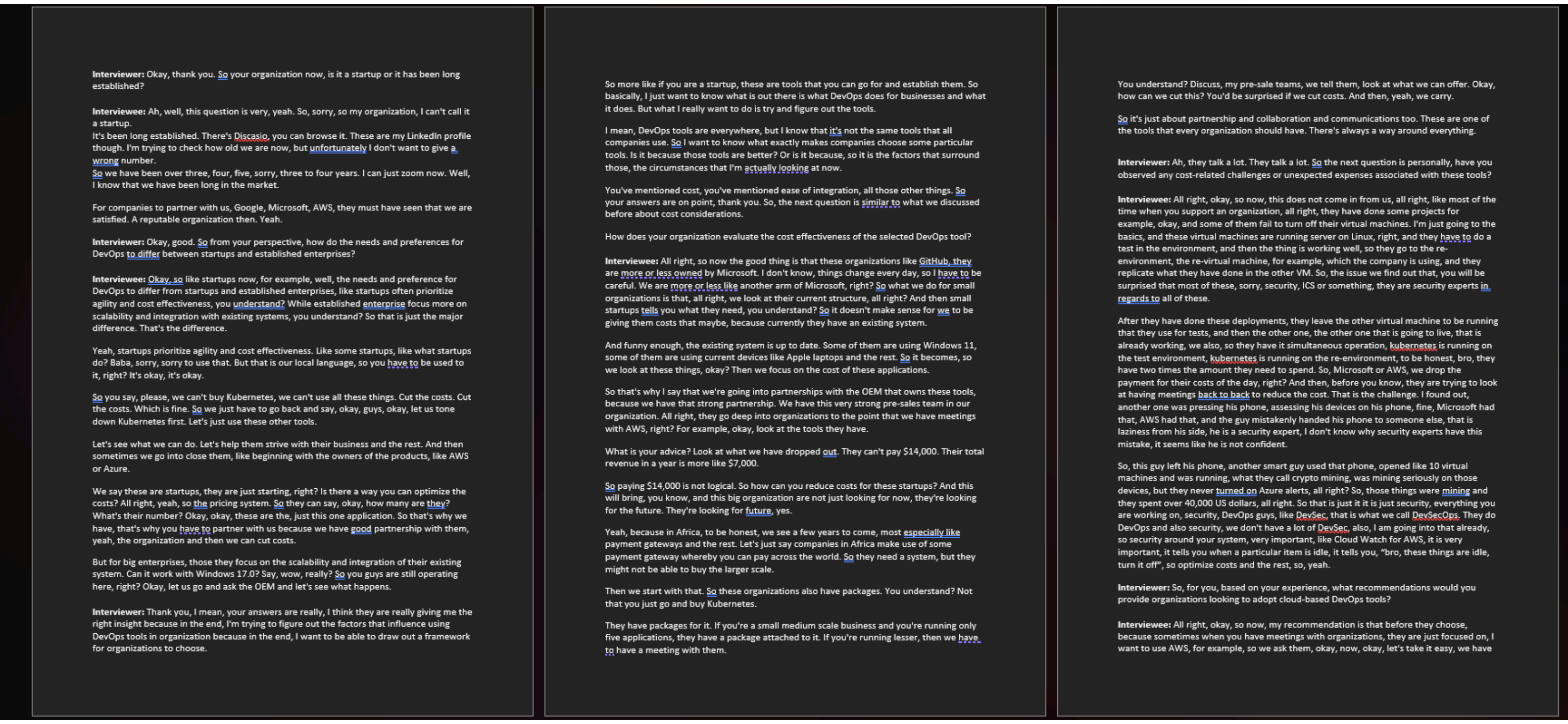
Interviewer: Thank you. So the next question is, how has your organization tested these tools, concerning scalability? And can you share any experiences related to this scalability? Like what's your experience is like?
Interviewee: All right, good. So like organizations tested scalability of these tools by gradually increasing the load on our deployments, right? Last time, there's what they call knee-jerks, right? I was working on some, trying to do some migration, like doing some tests for knee-jerks to see if we're going to more or less host a particular website, right? All right.

So we start from the basics. We start from the particular face of the projects to give a test using knee-jerks, because when you go to the browser, knee-jerks will tell you that, okay, this is available and the rest. We're trying to automate something.

So the good thing about this is that, you do a gradual increase. You don't put all your workload. You understand when pushing, maybe you're pushing to see your code, you try, in fact, there's a try system we use in our organization.

So we push firstly, then we watch, and then, okay, what is the impact? Okay, okay, okay, fine, fine. Okay, this is good. Because we have like, I don't want to talk about that though.

Like Git, we make use of Git too. Sorry, I didn't mention that. GitHub, where we push our code, but they have faces, they have the master tree, they have the main tree.



Interviewer: Okay, thank you. So your organization now, is it a startup or it has been long established?

Interviewee: Ah, well, this question is very, yeah. So, sorry, so my organization, I can't call it a startup.
It's been long established. There's Discasio, you can browse it. These are my LinkedIn profile though. I'm trying to check how old we are now, but unfortunately I don't want to give a wrong number.
So we have been over three, four, five, sorry, three to four years. I can just zoom now. Well, I know that we have been long in the market.

For companies to partner with us, Google, Microsoft, AWS, they must have seen that we are satisfied. A reputable organization then. Yeah.

Interviewer: Okay, good. So from your perspective, how do the needs and preferences for DevOps to differ between startups and established enterprises?

Interviewee: Okay, so like startups now, for example, well, the needs and preference for DevOps to differ from startups and established enterprises, like startups often prioritize agility and cost effectiveness, you understand? While established enterprise focus more on scalability and integration with existing systems, you understand? So that is just the major difference. That's the difference.

Yeah, startups prioritize agility and cost effectiveness. Like some startups, like what startups do? Baba, sorry, sorry to use that. But that is our local language, so you have to be used to it, right? It's okay, it's okay.

So you say, please, we can't buy Kubernetes, we can't use all these things. Cut the costs. Cut the costs. Which is fine. So we just have to go back and say, okay, guys, okay, let us tone down Kubernetes first. Let's just use these other tools.

Let's see what we can do. Let's help them strive with their business and the rest. And then sometimes we go into close them, like beginning with the owners of the products, like AWS or Azure.

We say these are startups, they are just starting, right? Is there a way you can optimize the costs? All right, yeah, so the pricing system. So they can say, okay, how many are they? What's their number? Okay, okay, these are the, just this one application. So that's why we have, that's why you have to partner with us because we have good partnership with them, yeah, the organization and then we can cut costs.

But for big enterprises, those they focus on the scalability and integration of their existing system. Can it work with Windows 17.0? Say, wow, really? So you guys are still operating here, right? Okay, let us go and ask the OEM and let's see what happens.

Interviewer: Thank you, I mean, your answers are really, I think they are really giving me the right insight because in the end, I'm trying to figure out the factors that influence using DevOps tools in organization because in the end, I want to be able to draw out a framework for organizations to choose.

So more like if you are a startup, these are tools that you can go for and establish them. So basically, I just want to know what is out there is what DevOps does for businesses and what it does. But what I really want to do is try and figure out the tools.

I mean, DevOps tools are everywhere, but I know that it's not the same tools that all companies use. So I want to know what exactly makes companies choose some particular tools. Is it because those tools are better? Or is it because, so it is the factors that surround those, the circumstances that I'm actually looking at now.

You've mentioned cost, you've mentioned ease of integration, all those other things. So your answers are on point, thank you. So, the next question is similar to what we discussed before about cost considerations.

How does your organization evaluate the cost effectiveness of the selected DevOps tool?

Interviewee: All right, so now the good thing is that these organizations like GitHub, they are more or less owned by Microsoft. I don't know, things change every day, so I have to be careful. We are more or less like another arm of Microsoft, right? So what we do for small organizations is that, all right, we look at their current structure, all right? And then small startups tells you what they need, you understand? So it doesn't make sense for we to be giving them costs that maybe, because currently they have an existing system.

And funny enough, the existing system is up to date. Some of them are using Windows 11, some of them are using current devices like Apple laptops and the rest. So it becomes, so we look at these things, okay? Then we focus on the cost of these applications.

So that's why I say that we're going into partnerships with the OEM that owns these tools, because we have that strong partnership. We have this very strong pre-sales team in our organization. All right, they go deep into organizations to the point that we have meetings with AWS, right? For example, okay, look at the tools they have.

What is your advice? Look at what we have dropped out. They can't pay $14,000. Their total revenue in a year is more like $7,000.

So paying $14,000 is not logical. So how can you reduce costs for these startups? And this will bring, you know, and this big organization are not just looking for now, they're looking for the future. They're looking for future, yes.

Yeah, because in Africa, to be honest, we see a few years to come, most especially like payment gateways and the rest. Let's just say companies in Africa make use of some payment gateway whereby you can pay across the world. So they need a system, but they might not be able to buy the larger scale.

Then we start with that. So these organizations also have packages. You understand? Not that you just go and buy Kubernetes.

They have packages for it. If you're a small medium scale business and you're running only five applications, they have a package attached to it. If you're running lesser, then we have to have a meeting with them.

You understand? Discuss, my pre-sale teams, we tell them, look at what we can offer. Okay, how can we cut this? You'd be surprised if we cut costs. And then, yeah, we carry.

So it's just about partnership and collaboration and communications too. These are one of the tools that every organization should have. There's always a way around everything.

Interviewer: Ah, they talk a lot. They talk a lot. So the next question is personally, have you observed any cost-related challenges or unexpected expenses associated with these tools?

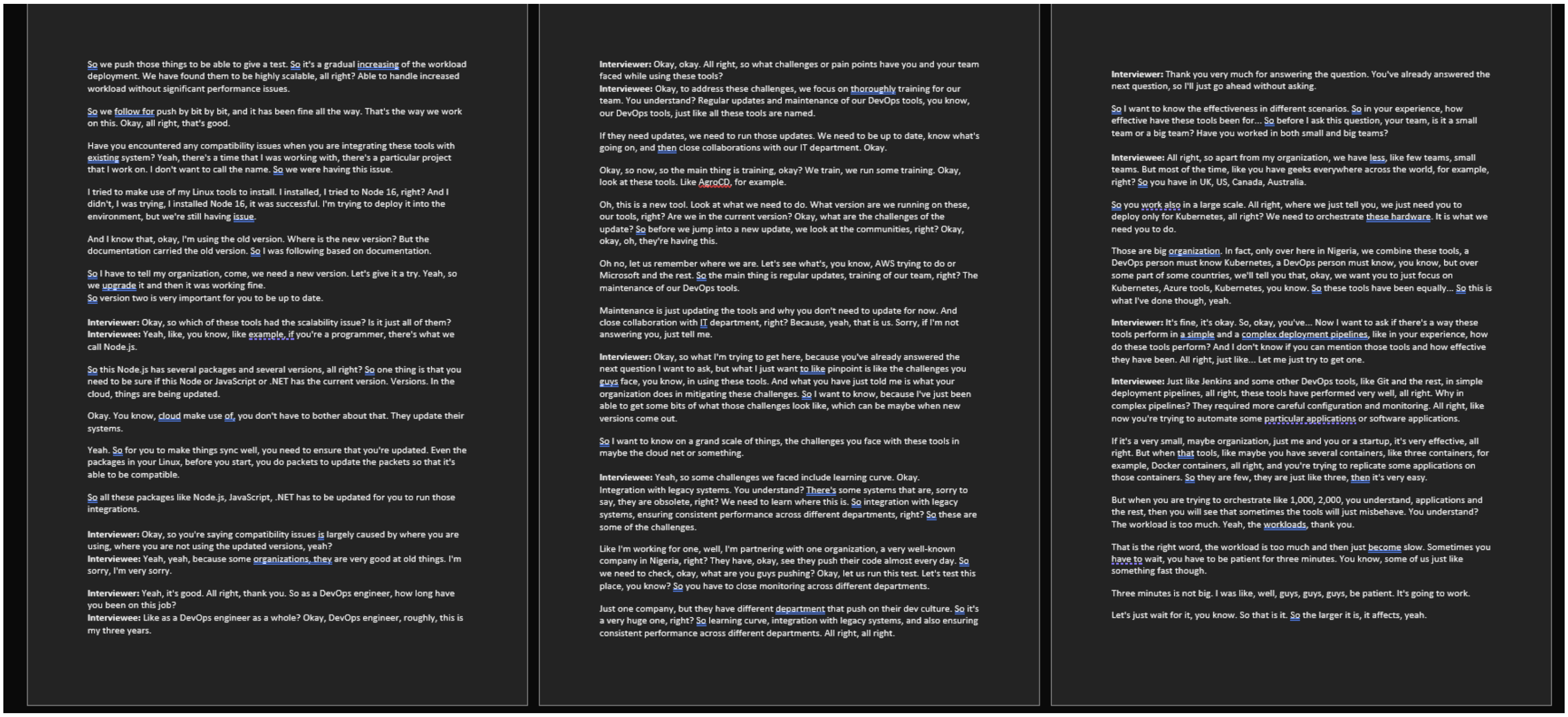
Interviewee: All right, okay, so now, this does not come in from us, all right, like most of the time when you support an organization, all right, they have done some projects for example, okay, and some of them fail to turn off their virtual machines. I'm just going to the basics, and these virtual machines are running server on Linux, right, and they have to do a test in the environment, and then the thing is working well, so they go to the re-environment, the re-virtual machine, for example, which the company is using, and they replicate what they have done in the other VM. So, the issue we find out that, you will be surprised that most of these, sorry, security, ICS or something, they are security experts in regards to all of these.

After they have done these deployments, they leave the other virtual machine to be running that they use for tests, and then the other one, the other one that is going to live, that is already working, we also, so they have it simultaneous operation, kubernetes is running on the test environment, kubernetes is running on the re-environment, to be honest, bro, they have two times the amount they need to spend. So, Microsoft or AWS, we drop the payment for their costs of the day, right? And then, before you know, they are trying to look at having meetings back to back to reduce the cost. That is the challenge. I found out, another one was pressing his phone, assessing his devices on his phone, fine, Microsoft had that, AWS had that, and the guy mistakenly handed his phone to someone else, that is laziness from his side, he is a security expert, I don't know why security experts have this mistake, it seems like he is not confident.

So, this guy left his phone, another smart guy used that phone, opened like 10 virtual machines and was running, what they call crypto mining, was mining seriously on those devices, but they never turned on Azure alerts, all right? So, those things were mining and they spent over 40,000 US dollars, all right. So that is just it it is just security, everything you are working on, security, DevOps guys, like DevSec, that is what we call DevSecOps. They do DevOps and also security, we don't have a lot of DevSec, also, I am going into that already, so security around your system, very important, like Cloud Watch for AWS, it is very important, it tells you when a particular item is idle, it tells you, "bro, these things are idle, turn it off", so optimize costs and the rest, so, yeah.

Interviewer: So, for you, based on your experience, what recommendations would you provide organizations looking to adopt cloud-based DevOps tools?

Interviewee: All right, okay, so now, my recommendation is that before they choose, because sometimes when you have meetings with organizations, they are just focused on, I want to use AWS, for example, so we ask them, okay, now, okay, let's take it easy, we have



So we push those things to be able to give a test. So it's a gradual increasing of the workload deployment. We have found them to be highly scalable, all right? Able to handle increased workload without significant performance issues.

So we follow for push by bit by bit, and it has been fine all the way. That's the way we work on this. Okay, all right, that's good.

Have you encountered any compatibility issues when you are integrating these tools with existing system? Yeah, there's a time that I was working with, there's a particular project that I work on. I don't want to call the name. So we were having this issue.

I tried to make use of my Linux tools to install. I installed, I tried to Node 16, right? And I didn't, I was trying, I installed Node 16, it was successful. I'm trying to deploy it into the environment, but we're still having issue.

And I know that, okay, I'm using the old version. Where is the new version? But the documentation carried the old version. So I was following based on documentation.

So I have to tell my organization, come, we need a new version. Let's give it a try. Yeah, so we upgrade it and then it was working fine.
So version two is very important for you to be up to date.

Interviewer: Okay, so which of these tools had the scalability issue? Is it just all of them?
Interviewee: Yeah, like, you know, like example, if you're a programmer, there's what we call Node.js.

So this Node.js has several packages and several versions, all right? So one thing is that you need to be sure if this Node or JavaScript or .NET has the current version. Versions. In the cloud, things are being updated.

Okay. You know, cloud make use of, you don't have to bother about that. They update their systems.

Yeah. So for you to make things sync well, you need to ensure that you're updated. Even the packages in your Linux, before you start, you do packets to update the packets so that it's able to be compatible.

So all these packages like Node.js, JavaScript, .NET has to be updated for you to run those integrations.

Interviewer: Okay, so you're saying compatibility issues is largely caused by where you are using, where you are not using the updated versions, yeah?
Interviewee: Yeah, yeah, because some organizations, they are very good at old things. I'm sorry, I'm very sorry.

Interviewer: Yeah, it's good. All right, thank you. So as a DevOps engineer, how long have you been on this job?
Interviewee: Like as a DevOps engineer as a whole? Okay, DevOps engineer, roughly, this is my three years.

Interviewer: Okay, okay. All right, so what challenges or pain points have you and your team faced while using these tools?
Interviewee: Okay, to address these challenges, we focus on thoroughly training for our team. You understand? Regular updates and maintenance of our DevOps tools, you know, our DevOps tools, just like all these tools are named.

If they need updates, we need to run those updates. We need to be up to date, know what's going on, and then close collaborations with our IT department. Okay.

Okay, so now, so the main thing is training, okay? We train, we run some training. Okay, look at these tools. Like AgroCD, for example.

Oh, this is a new tool. Look at what we need to do. What version are we running on these, our tools, right? Are we in the current version? Okay, what are the challenges of the update? So before we jump into a new update, we look at the communities, right? Okay, okay, oh, they're having this.

Oh no, let us remember where we are. Let's see what's, you know, AWS trying to do or Microsoft and the rest. So the main thing is regular updates, training of our team, right? The maintenance of our DevOps tools.

Maintenance is just updating the tools and why you don't need to update for now. And close collaboration with IT department, right? Because, yeah, that is us. Sorry, if I'm not answering you, just tell me.

Interviewer: Okay, so what I'm trying to get here, because you've already answered the next question I want to ask, but what I just want to like pinpoint is like the challenges you guys face, you know, in using these tools. And what you have just told me is what your organization does in mitigating these challenges. So I want to know, because I've just been able to get some bits of what those challenges look like, which can be maybe when new versions come out.

So I want to know on a grand scale of things, the challenges you face with these tools in maybe the cloud net or something.

Interviewee: Yeah, so some challenges we faced include learning curve. Okay. Integration with legacy systems. You understand? There's some systems that are, sorry to say, they are obsolete, right? We need to learn where this is. So integration with legacy systems, ensuring consistent performance across different departments, right? So these are some of the challenges.

Like I'm working for one, well, I'm partnering with one organization, a very well-known company in Nigeria, right? They have, okay, see they push their code almost every day. So we need to check, okay, what are you guys pushing? Okay, let us run this test. Let's test this place, you know? So you have to close monitoring across different departments.

Just one company, but they have different department that push on their dev culture. So it's a very huge one, right? So learning curve, integration with legacy systems, and also ensuring consistent performance across different departments. All right, all right.

Interviewer: Thank you very much for answering the question. You've already answered the next question, so I'll just go ahead without asking.

So I want to know the effectiveness in different scenarios. So in your experience, how effective have these tools been for… So before I ask this question, your team, is it a small team or a big team? Have you worked in both small and big teams?

Interviewee: All right, so apart from my organization, we have less, like few teams, small teams. But most of the time, like you have geeks everywhere across the world, for example, right? So you have in UK, US, Canada, Australia.

So you work also in a large scale. All right, where we just tell you, we just need you to deploy only for Kubernetes, all right? We need to orchestrate these hardware. It is what we need you to do.

Those are big organization. In fact, only over here in Nigeria, we combine these tools, a DevOps person must know Kubernetes, a DevOps person must know, you know, but over some part of some countries, we'll tell you that, okay, we want you to just focus on Kubernetes, Azure tools, Kubernetes, you know. So these tools have been equally… So this is what I've done though, yeah.

Interviewer: It's fine, it's okay. So, okay, you've… Now I want to ask if there's a way these tools perform in a simple and a complex deployment pipelines, like in your experience, how do these tools perform? And I don't know if you can mention those tools and how effective they have been. All right, just like… Let me just try to get one.

Interviewee: Just like Jenkins and some other DevOps tools, like Git and the rest, in simple deployment pipelines, all right, these tools have performed very well, all right. Why in complex pipelines? They required more careful configuration and monitoring. All right, like now you're trying to automate some particular applications or software applications.

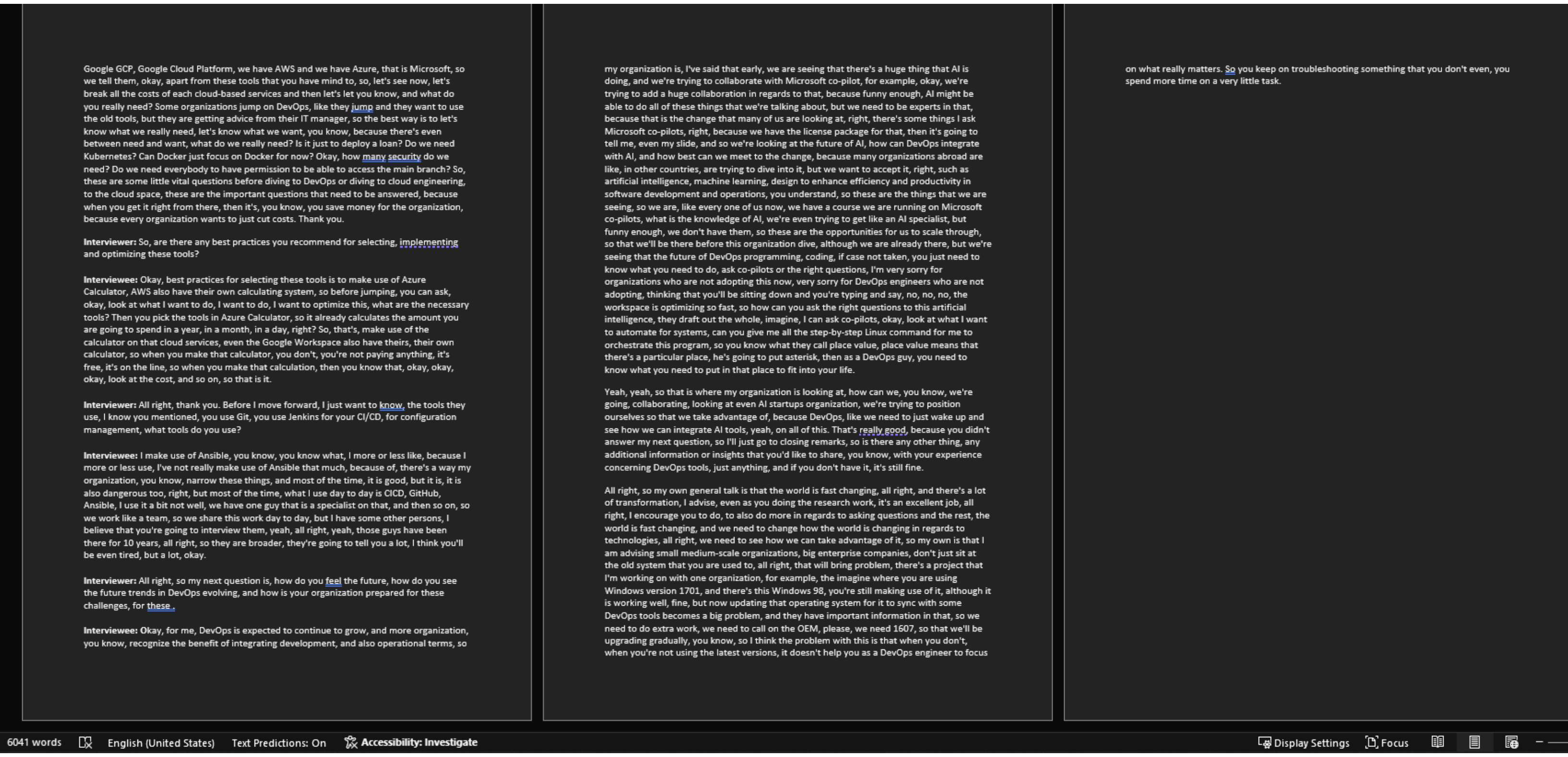
If it's a very small, maybe organization, just me and you or a startup, it's very effective, all right. But when that tools, like maybe you have several containers, like three containers, for example, Docker containers, all right, and you're trying to replicate some applications on those containers. So they are few, they are just like three, then it's very easy.

But when you are trying to orchestrate like 1,000, 2,000, you understand, applications and the rest, then you will see that sometimes the tools will just misbehave. You understand? The workload is too much. Yeah, the workloads, thank you.

That is the right word, the workload is too much and then just become slow. Sometimes you have to wait, you have to be patient for three minutes. You know, some of us just like something fast though.

Three minutes is not big. I was like, well, guys, guys, guys, be patient. It's going to work.

Let's just wait for it, you know. So that is it. So the larger it is, it affects, yeah.



Google GCP, Google Cloud Platform, we have AWS and we have Azure, that is Microsoft, so we tell them, okay, apart from these tools that you have mind to, so, let's see now, let's break all the costs of each cloud-based services and then let's let you know, and what do you really need? Some organizations jump on DevOps, like they jump and they want to use the old tools, but they are getting advice from their IT manager, so the best way is to let's know what we really need, let's know what we want, you know, because there's even between need and want, what do we really need? Is it just to deploy a loan? Do we need Kubernetes? Can Docker just focus on Docker for now? Okay, how many security do we need? Do we need everybody to have permission to be able to access the main branch? So, these are some little vital questions before diving to DevOps or diving to cloud engineering, to the cloud space, these are the important questions that need to be answered, because when you get it right from there, then it's, you know, you save money for the organization, because every organization wants to just cut costs. Thank you.

Interviewer: So, are there any best practices you recommend for selecting, implementing and optimizing these tools?

Interviewee: Okay, best practices for selecting these tools is to make use of Azure Calculator, AWS also have their own calculating system, so before jumping, you can ask, okay, look at what I want to do, I want to do, I want to optimize this, what are the necessary tools? Then you pick the tools in Azure Calculator, so it already calculates the amount you are going to spend in a year, in a month, in a day, right? So, that's, make use of the calculator on that cloud services, even the Google Workspace also have theirs, their own calculator, so when you make that calculator, you don't, you're not paying anything, it's free, it's on the line, so when you make that calculation, then you know that, okay, okay, okay, look at the cost, and so on, so that is it.

Interviewer: All right, thank you. Before I move forward, I just want to know, the tools they use, I know you mentioned, you use Git, you use Jenkins for your CI/CD, for configuration management, what tools do you use?

Interviewee: I make use of Ansible, you know, you know what, I more or less like, because I more or less use, I've not really make use of Ansible that much, because of, there's a way my organization, you know, narrow these things, and most of the time, it is good, but it is, it is also dangerous too, right, but most of the time, what I use day to day is CICD, GitHub, Ansible, I use it a bit not well, we have one guy that is a specialist on that, and then so on, so we work like a team, so we share this work day to day, but I have some other persons, I believe that you're going to interview them, yeah, all right, yeah, those guys have been there for 10 years, all right, so they are broader, they're going to tell you a lot, I think you'll be even tired, but a lot, okay.

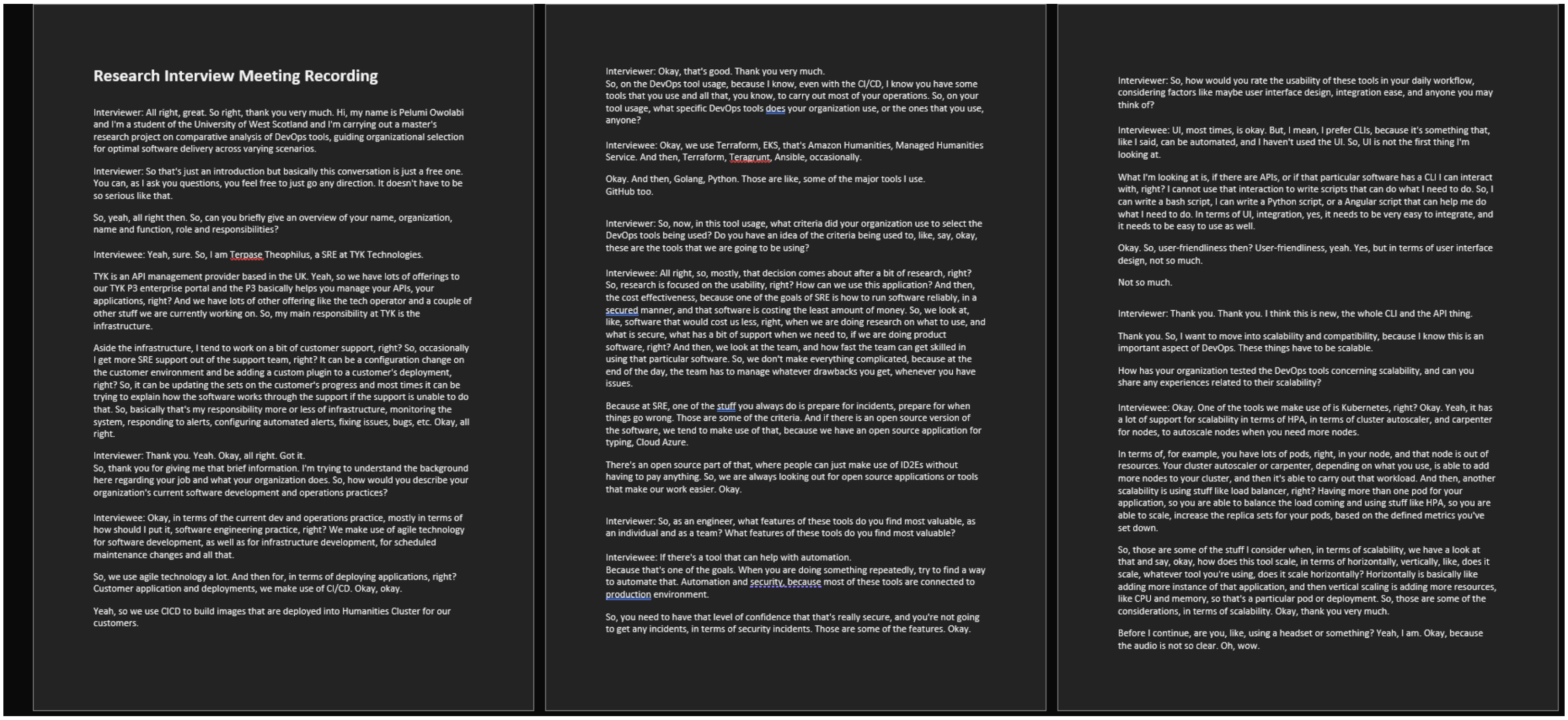
Interviewer: All right, so my next question is, how do you feel the future, how do you see the future trends in DevOps evolving, and how is your organization prepared for these challenges, for these .

Interviewee: Okay, for me, DevOps is expected to continue to grow, and more organization, you know, recognize the benefit of integrating development, and also operational terms, so my organization is, I've said that early, we are seeing that there's a huge thing that AI is doing, and we're trying to collaborate with Microsoft co-pilot, for example, okay, we're trying to add a huge collaboration in regards to that, because funny enough, AI might be able to do all of these things that we're talking about, but we need to be experts in that, because that is the change that many of us are looking at, right, there's some things I ask Microsoft co-pilots, right, because we have the license package for that, then it's going to tell me, even my slide, and so we're looking at the future of AI, how can DevOps integrate with AI, and how best can we meet to the change, because many organizations abroad are like, in other countries, are trying to dive into it, but we want to accept it, right, such as artificial intelligence, machine learning, design to enhance efficiency and productivity in software development and operations, you understand, so these are the things that we are seeing, so we are, like every one of us now, we have a course we are running on Microsoft co-pilots, what is the knowledge of AI, we're even trying to get like an AI specialist, but funny enough, we don't have them, so these are the opportunities for us to scale through, so that we'll be there before this organization dive, although we are already there, but we're seeing that the future of DevOps programming, coding, if case not taken, you just need to know what you need to do, ask co-pilots or the right questions, I'm very sorry for organizations who are not adopting this now, very sorry for DevOps engineers who are not adopting, thinking that you'll be sitting down and you're typing and say, no, no, no, the workspace is optimizing so fast, so how can you ask the right questions to this artificial intelligence, they draft out the whole, imagine, I can ask co-pilots, okay, look at what I want to automate for systems, can you give me all the step-by-step Linux command for me to orchestrate this program, so you know what they call place value, place value means that there's a particular place, he's going to put asterisk, then as a DevOps guy, you need to know what you need to put in that place to fit into your life.

Yeah, yeah, so that is where my organization is looking at, how can we, you know, we're going, collaborating, looking at even AI startups organization, we're trying to position ourselves so that we take advantage of, because DevOps, like we need to just wake up and see how we can integrate AI tools, yeah, on all of this. That's really good, because you didn't answer my next question, so I'll just go to closing remarks, so is there any other thing, any additional information or insights that you'd like to share, you know, with your experience concerning DevOps tools, just anything, and if you don't have it, it's still fine.

All right, so my own general talk is that the world is fast changing, all right, and there's a lot of transformation, I advise, even as you doing the research work, it's an excellent job, all right, I encourage you to do, to also do more in regards to asking questions and the rest, the world is fast changing, and we need to change how the world is changing in regards to technologies, all right, we need to see how we can take advantage of it, so my own is that I am advising small medium-scale organizations, big enterprise companies, don't just sit at the old system that you are used to, all right, that will bring problem, there's a project that I'm working on with one organization, for example, the imagine where you are using Windows version 1701, and there's this Windows 98, you're still making use of it, although it is working well, fine, but now updating that operating system for it to sync with some DevOps tools becomes a big problem, and they have important information in that, so we need to do extra work, we need to call on the OEM, please, we need 1607, so that we'll be upgrading gradually, you know, so I think the problem with this is that when you don't, when you're not using the latest versions, it doesn't help you as a DevOps engineer to focus on what really matters. So you keep on troubleshooting something that you don't even, you spend more time on a very little task.

RESPONDENT 3



Research Interview Meeting Recording

Interviewer: All right, great. So right, thank you very much. Hi, my name is Pelumi Owolabi and I'm a student of the University of West Scotland and I'm carrying out a master's research project on comparative analysis of DevOps tools, guiding organizational selection for optimal software delivery across varying scenarios.

Interviewer: So that's just an introduction but basically this conversation is just a free one. You can, as I ask you questions, you feel free to just go any direction. It doesn't have to be so serious like that.

So, yeah, all right then. So, can you briefly give an overview of your name, organization, name and function, role and responsibilities?

Interviewee: Yeah, sure. So, I am Terpase Theophilus, a SRE at TYK Technologies.

TYK is an API management provider based in the UK. Yeah, so we have lots of offerings to our TYK P3 enterprise portal and the P3 basically helps you manage your APIs, your applications, right? And we have lots of other offering like the tech operator and a couple of other stuff we are currently working on. So, my main responsibility at TYK is the infrastructure.

Aside the infrastructure, I tend to work on a bit of customer support, right? So, occasionally I get more SRE support out of the support team, right? It can be a configuration change on the customer environment and be adding a custom plugin to a customer's deployment, right? So, it can be updating the sets on the customer's progress and most times it can be trying to explain how the software works through the support if the support is unable to do that. So, basically that's my responsibility more or less of infrastructure, monitoring the system, responding to alerts, configuring automated alerts, fixing issues, bugs, etc. Okay, all right.

Interviewer: Thank you. Yeah. Okay, all right. Got it.
So, thank you for giving me that brief information. I'm trying to understand the background here regarding your job and what your organization does. So, how would you describe your organization's current software development and operations practices?

Interviewee: Okay, in terms of the current dev and operations practice, mostly in terms of how should I put it, software engineering practice, right? We make use of agile technology for software development, as well as for infrastructure development, for scheduled maintenance changes and all that.

So, we use agile technology a lot. And then for, in terms of deploying applications, right? Customer application and deployments, we make use of CI/CD. Okay, okay.

Yeah, so we use CICD to build images that are deployed into Humanities Cluster for our customers.

Interviewer: Okay, that's good. Thank you very much.
So, on the DevOps tool usage, because I know, even with the CI/CD, I know you have some tools that you use and all that, you know, to carry out most of your operations. So, on your tool usage, what specific DevOps tools does your organization use, or the ones that you use, anyone?

Interviewee: Okay, we use Terraform, EKS, that's Amazon Humanities, Managed Humanities Service. And then, Terraform, Teragrunt, Ansible, occasionally.

Okay. And then, Golang, Python. Those are like, some of the major tools I use.
GitHub too.

Interviewer: So, now, in this tool usage, what criteria did your organization use to select the DevOps tools being used? Do you have an idea of the criteria being used to, like, say, okay, these are the tools that we are going to be using?

Interviewee: All right, so, mostly, that decision comes about after a bit of research, right? So, research is focused on the usability, right? How can we use this application? And then, the cost effectiveness, because one of the goals of SRE is how to run software reliably, in a secured manner, and that software is costing the least amount of money. So, we look at, like, software that would cost us less, right, when we are doing research on what to use, and what is secure, what has a bit of support when we need to, if we are doing product software, right? And then, we look at the team, and how fast the team can get skilled in using that particular software. So, we don't make everything complicated, because at the end of the day, the team has to manage whatever drawbacks you get, whenever you have issues.

Because at SRE, one of the stuff you always do is prepare for incidents, prepare for when things go wrong. Those are some of the criteria. And if there is an open source version of the software, we tend to make use of that, because we have an open source application for typing, Cloud Azure.

There's an open source part of that, where people can just make use of ID2Es without having to pay anything. So, we are always looking out for open source applications or tools that make our work easier. Okay.

Interviewer: So, as an engineer, what features of these tools do you find most valuable, as an individual and as a team? What features of these tools do you find most valuable?

Interviewee: If there's a tool that can help with automation.
Because that's one of the goals. When you are doing something repeatedly, try to find a way to automate that. Automation and security, because most of these tools are connected to production environment.

So, you need to have that level of confidence that that's really secure, and you're not going to get any incidents, in terms of security incidents. Those are some of the features. Okay.

Interviewer: So, how would you rate the usability of these tools in your daily workflow, considering factors like maybe user interface design, integration ease, and anyone you may think of?

Interviewee: UI, most times, is okay. But, I mean, I prefer CLIs, because it's something that, like I said, can be automated, and I haven't used the UI. So, UI is not the first thing I'm looking at.

What I'm looking at is, if there are APIs, or if that particular software has a CLI I can interact with, right? I cannot use that interaction to write scripts that can do what I need to do. So, I can write a bash script, I can write a Python script, or a Angular script that can help me do what I need to do. In terms of UI, integration, yes, it needs to be very easy to integrate, and it needs to be easy to use as well.

Okay. So, user-friendliness then? User-friendliness, yeah. Yes, but in terms of user interface design, not so much.

Not so much.

Interviewer: Thank you. Thank you. I think this is new, the whole CLI and the API thing.

Thank you. So, I want to move into scalability and compatibility, because I know this is an important aspect of DevOps. These things have to be scalable.

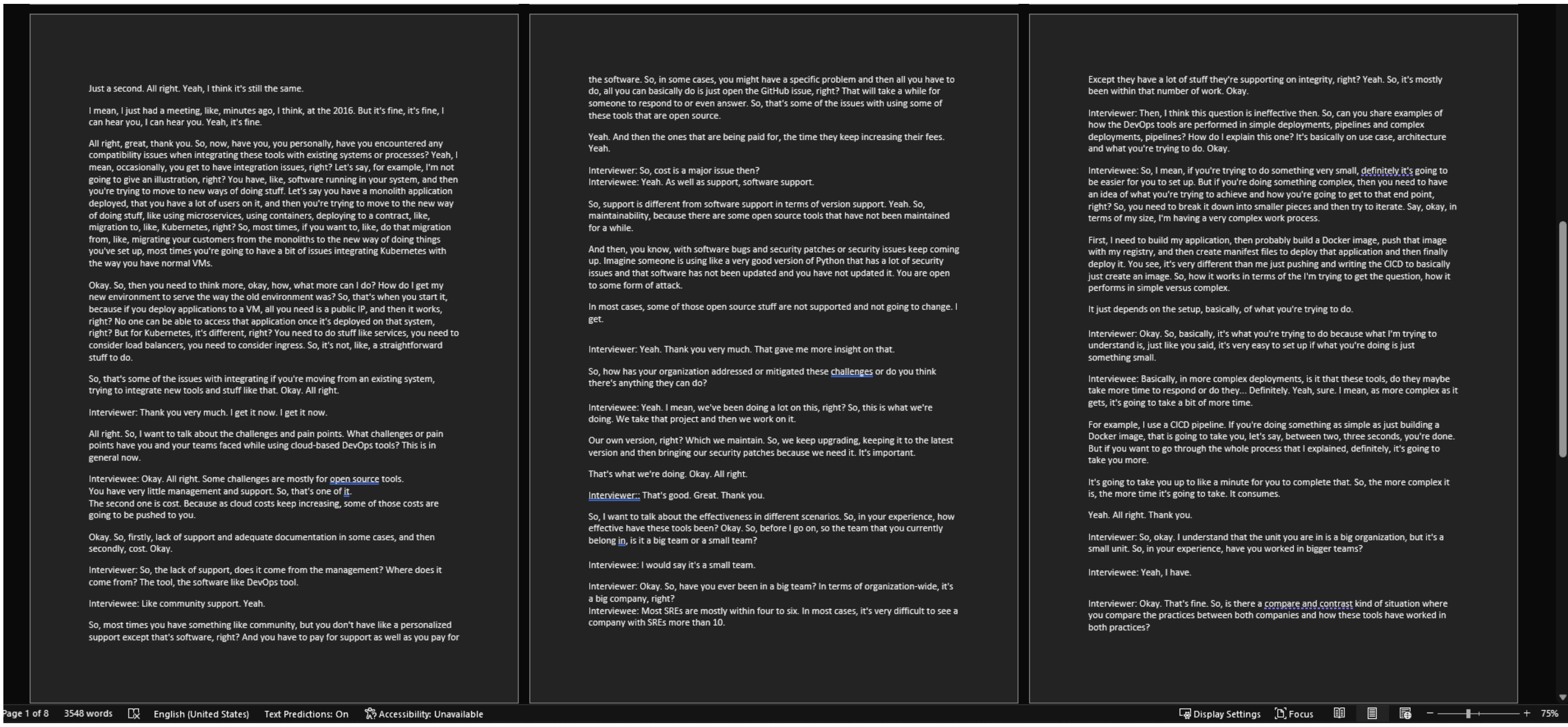
How has your organization tested the DevOps tools concerning scalability, and can you share any experiences related to their scalability?

Interviewee: Okay. One of the tools we make use of is Kubernetes, right? Okay. Yeah, it has a lot of support for scalability in terms of HPA, in terms of cluster autoscaler, and carpenter for nodes, to autoscale nodes when you need more nodes.

In terms of, for example, you have lots of pods, right, in your node, and that node is out of resources. Your cluster autoscaler or carpenter, depending on what you use, is able to add more nodes to your cluster, and then it's able to carry out that workload. And then, another scalability is using stuff like load balancer, right? Having more than one pod for your application, so you are able to balance the load coming and using stuff like HPA, so you are able to scale, increase the replica sets for your pods, based on the defined metrics you've set down.

So, those are some of the stuff I consider when, in terms of scalability, we have a look at that and say, okay, how does this tool scale, in terms of horizontally, vertically, like, does it scale, whatever tool you're using, does it scale horizontally? Horizontally is basically like adding more instance of that application, and then vertical scaling is adding more resources, like CPU and memory, so that's a particular pod or deployment. So, those are some of the considerations, in terms of scalability. Okay, thank you very much.

Before I continue, are you, like, using a headset or something? Yeah, I am. Okay, because the audio is not so clear. Oh, wow.



Just a second. All right. Yeah, I think it's still the same.

I mean, I just had a meeting, like, minutes ago, I think, at the 2016. But it's fine, it's fine, I can hear you, I can hear you. Yeah, it's fine.

All right, great, thank you. So, now, have you, you personally, have you encountered any compatibility issues when integrating these tools with existing systems or processes? Yeah, I mean, occasionally, you get to have integration issues, right? Let's say, for example, I'm not going to give an illustration, right? You have, like, software running in your system, and then you're trying to move to new ways of doing stuff. Let's say you have a monolith application deployed, that you have a lot of users on it, and then you're trying to move to the new way of doing stuff, like using microservices, using containers, deploying to a contract, like, migration to, like, Kubernetes, right? So, most times, if you want to, like, do that migration from, like, migrating your customers from the monoliths to the new way of doing things you've set up, most times you're going to have a bit of issues integrating Kubernetes with the way you have normal VMs.

Okay. So, then you need to think more, okay, how, what more can I do? How do I get my new environment to serve the way the old environment was? So, that's when you start it, because if you deploy applications to a VM, all you need is a public IP, and then it works, right? No one can be able to access that application once it's deployed on that system, right? But for Kubernetes, it's different, right? You need to do stuff like services, you need to consider load balancers, you need to consider ingress. So, it's not, like, a straightforward stuff to do.

So, that's some of the issues with integrating if you're moving from an existing system, trying to integrate new tools and stuff like that. Okay. All right.

Interviewer: Thank you very much. I get it now. I get it now.

All right. So, I want to talk about the challenges and pain points. What challenges or pain points have you and your teams faced while using cloud-based DevOps tools? This is in general now.

Interviewee: Okay. All right. Some challenges are mostly for open source tools.
You have very little management and support. So, that's one of it.
The second one is cost. Because as cloud costs keep increasing, some of those costs are going to be pushed to you.

Okay. So, firstly, lack of support and adequate documentation in some cases, and then secondly, cost. Okay.

Interviewer: So, the lack of support, does it come from the management? Where does it come from? The tool, the software like DevOps tool.

Interviewee: Like community support. Yeah.

So, most times you have something like community, but you don't have like a personalized support except that's software, right? And you have to pay for support as well as you pay for the software. So, in some cases, you might have a specific problem and then all you have to do, all you can basically do is just open the GitHub issue, right? That will take a while for someone to respond to or even answer. So, that's some of the issues with using some of these tools that are open source.

Yeah. And then the ones that are being paid for, the time they keep increasing their fees. Yeah.

Interviewer: So, cost is a major issue then?
Interviewee: Yeah. As well as support, software support.

So, support is different from software support in terms of version support. Yeah. So, maintainability, because there are some open source tools that have not been maintained for a while.

And then, you know, with software bugs and security patches or security issues keep coming up. Imagine someone is using like a very good version of Python that has a lot of security issues and that software has not been updated and you have not updated it. You are open to some form of attack.

In most cases, some of those open source stuff are not supported and not going to change. I get.

Interviewer: Yeah. Thank you very much. That gave me more insight on that.

So, how has your organization addressed or mitigated these challenges or do you think there's anything they can do?

Interviewee: Yeah. I mean, we've been doing a lot on this, right? So, this is what we're doing. We take that project and then we work on it.

Our own version, right? Which we maintain. So, we keep upgrading, keeping it to the latest version and then bringing our security patches because we need it. It's important.

That's what we're doing. Okay. All right.

Interviewer:: That's good. Great. Thank you.

So, I want to talk about the effectiveness in different scenarios. So, in your experience, how effective have these tools been? Okay. So, before I go on, so the team that you currently belong in, is it a big team or a small team?

Interviewee: I would say it's a small team.

Interviewer: Okay. So, have you ever been in a big team? In terms of organization-wide, it's a big company, right?
Interviewee: Most SREs are mostly within four to six. In most cases, it's very difficult to see a company with SREs more than 10.

Except they have a lot of stuff they're supporting on integrity, right? Yeah. So, it's mostly been within that number of work. Okay.

Interviewer: Then, I think this question is ineffective then. So, can you share examples of how the DevOps tools are performed in simple deployments, pipelines and complex deployments, pipelines? How do I explain this one? It's basically on use case, architecture and what you're trying to do. Okay.

Interviewee: So, I mean, if you're trying to do something very small, definitely it's going to be easier for you to set up. But if you're doing something complex, then you need to have an idea of what you're trying to achieve and how you're going to get to that end point, right? So, you need to break it down into smaller pieces and then try to iterate. Say, okay, in terms of my size, I'm having a very complex work process.

First, I need to build my application, then probably build a Docker image, push that image with my registry, and then create manifest files to deploy that application and then finally deploy it. You see, it's very different than me just pushing and writing the CICD to basically just create an image. So, how it works in terms of the I'm trying to get the question, how it performs in simple versus complex.

It just depends on the setup, basically, of what you're trying to do.

Interviewer: Okay. So, basically, it's what you're trying to do because what I'm trying to understand is, just like you said, it's very easy to set up if what you're doing is just something small.

Interviewee: Basically, in more complex deployments, is it that these tools, do they maybe take more time to respond or do they... Definitely. Yeah, sure. I mean, as more complex as it gets, it's going to take a bit of more time.

For example, I use a CICD pipeline. If you're doing something as simple as just building a Docker image, that is going to take you, let's say, between two, three seconds, you're done. But if you want to go through the whole process that I explained, definitely, it's going to take you more.

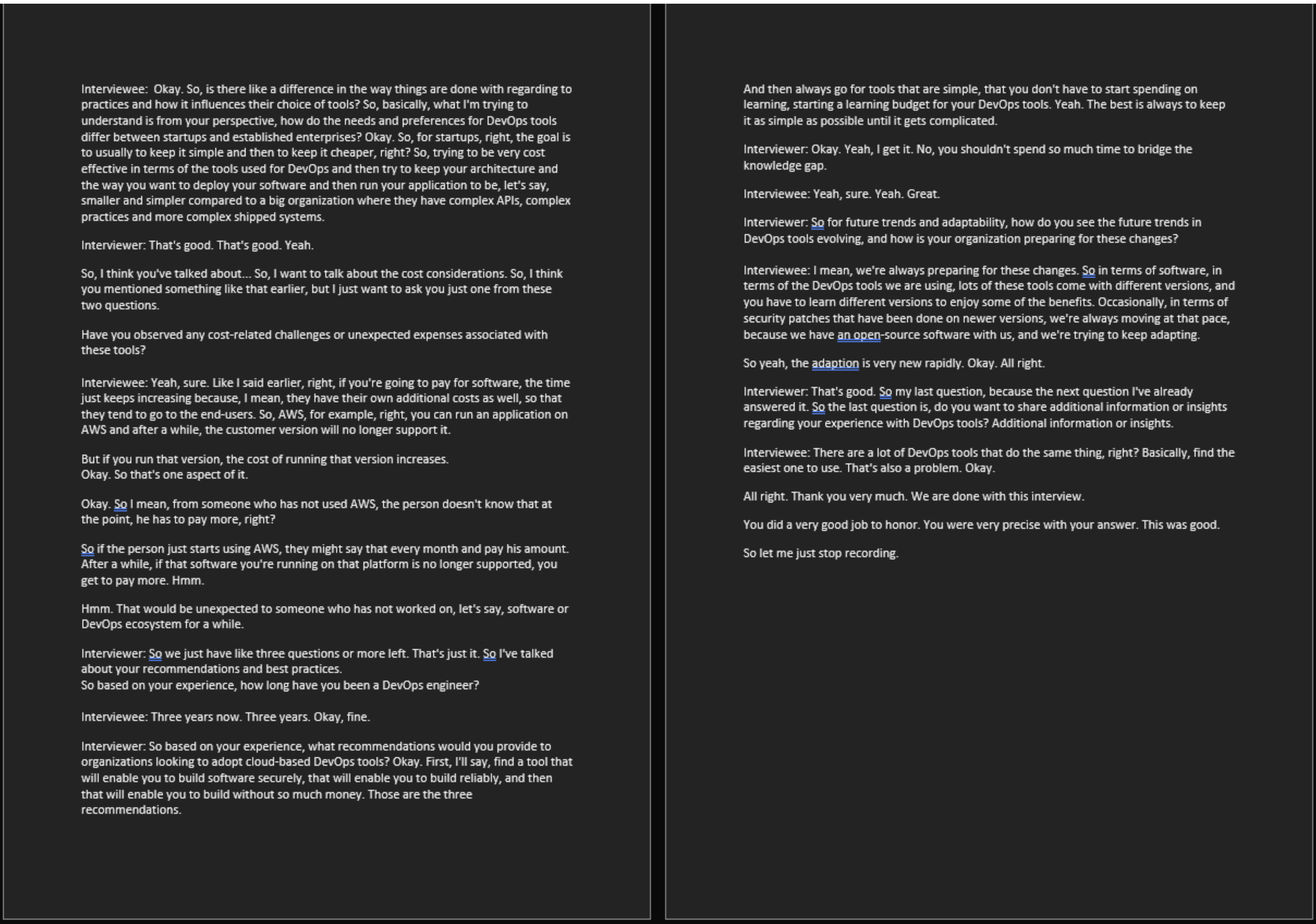
It's going to take you up to like a minute for you to complete that. So, the more complex it is, the more time it's going to take. It consumes.

Yeah. All right. Thank you.

Interviewer: So, okay. I understand that the unit you are in is a big organization, but it's a small unit. So, in your experience, have you worked in bigger teams?

Interviewee: Yeah, I have.

Interviewer: Okay. That's fine. So, is there a compare and contrast kind of situation where you compare the practices between both companies and how these tools have worked in both practices?



Interviewee: Okay. So, is there like a difference in the way things are done with regarding to practices and how it influences their choice of tools? So, basically, what I'm trying to understand is from your perspective, how do the needs and preferences for DevOps tools differ between startups and established enterprises? Okay. So, for startups, right, the goal is to usually to keep it simple and then to keep it cheaper, right? So, trying to be very cost effective in terms of the tools used for DevOps and then try to keep your architecture and the way you want to deploy your software and then run your application to be, let's say, smaller and simpler compared to a big organization where they have complex APIs, complex practices and more complex shipped systems.

Interviewer: That's good. That's good. Yeah.

So, I think you've talked about... So, I want to talk about the cost considerations. So, I think you mentioned something like that earlier, but I just want to ask you just one from these two questions.

Have you observed any cost-related challenges or unexpected expenses associated with these tools?

Interviewee: Yeah, sure. Like I said earlier, right, if you're going to pay for software, the time just keeps increasing because, I mean, they have their own additional costs as well, so that they tend to go to the end-users. So, AWS, for example, right, you can run an application on AWS and after a while, the customer version will no longer support it.

But if you run that version, the cost of running that version increases.
Okay. So that's one aspect of it.

Okay. So I mean, from someone who has not used AWS, the person doesn't know that at the point, he has to pay more, right?

So if the person just starts using AWS, they might say that every month and pay his amount. After a while, if that software you're running on that platform is no longer supported, you get to pay more. Hmm.

Hmm. That would be unexpected to someone who has not worked on, let's say, software or DevOps ecosystem for a while.

Interviewer: So we just have like three questions or more left. That's just it. So I've talked about your recommendations and best practices.
So based on your experience, how long have you been a DevOps engineer?

Interviewee: Three years now. Three years. Okay, fine.

Interviewer: So based on your experience, what recommendations would you provide to organizations looking to adopt cloud-based DevOps tools? Okay. First, I'll say, find a tool that will enable you to build software securely, that will enable you to build reliably, and then that will enable you to build without so much money. Those are the three recommendations.

And then always go for tools that are simple, that you don't have to start spending on learning, starting a learning budget for your DevOps tools. Yeah. The best is always to keep it as simple as possible until it gets complicated.

Interviewer: Okay. Yeah, I get it. No, you shouldn't spend so much time to bridge the knowledge gap.

Interviewee: Yeah, sure. Yeah. Great.

Interviewer: So for future trends and adaptability, how do you see the future trends in DevOps tools evolving, and how is your organization preparing for these changes?

Interviewee: I mean, we're always preparing for these changes. So in terms of software, in terms of the DevOps tools we are using, lots of these tools come with different versions, and you have to learn different versions to enjoy some of the benefits. Occasionally, in terms of security patches that have been done on newer versions, we're always moving at that pace, because we have an open-source software with us, and we're trying to keep adapting.

So yeah, the adaption is very new rapidly. Okay. All right.

Interviewer: That's good. So my last question, because the next question I've already answered it. So the last question is, do you want to share additional information or insights regarding your experience with DevOps tools? Additional information or insights.

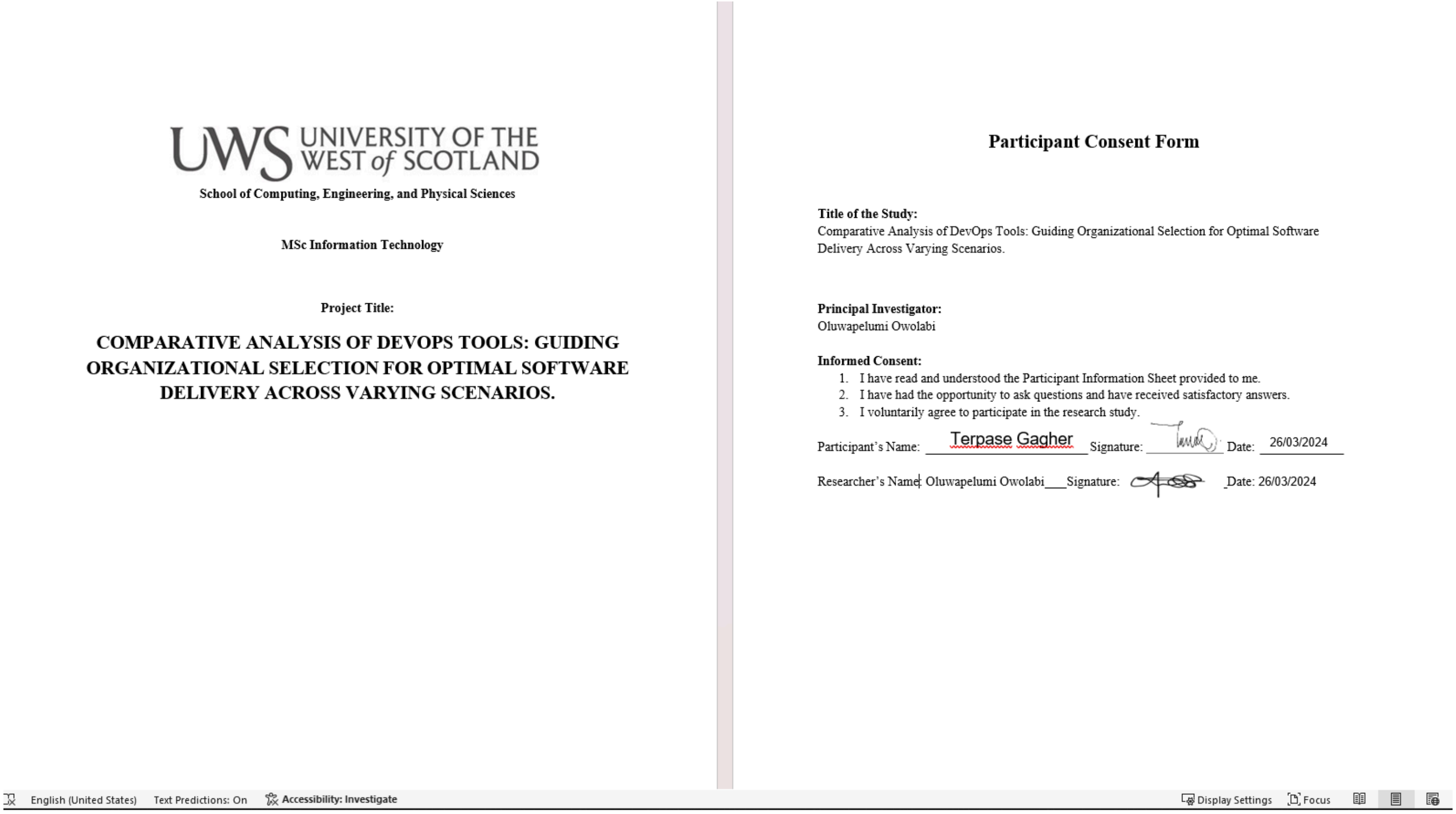
Interviewee: There are a lot of DevOps tools that do the same thing, right? Basically, find the easiest one to use. That's also a problem. Okay.

All right. Thank you very much. We are done with this interview.

You did a very good job to honor. You were very precise with your answer. This was good.

So let me just stop recording.

CONSENT FORMS



UWS UNIVERSITY OF THE WEST *of* SCOTLAND

School of Computing, Engineering, and Physical Sciences

MSc Information Technology

Project Title:

COMPARATIVE ANALYSIS OF DEVOPS TOOLS: GUIDING ORGANIZATIONAL SELECTION FOR OPTIMAL SOFTWARE DELIVERY ACROSS VARYING SCENARIOS.

Participant Consent Form

Title of the Study:
Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios.

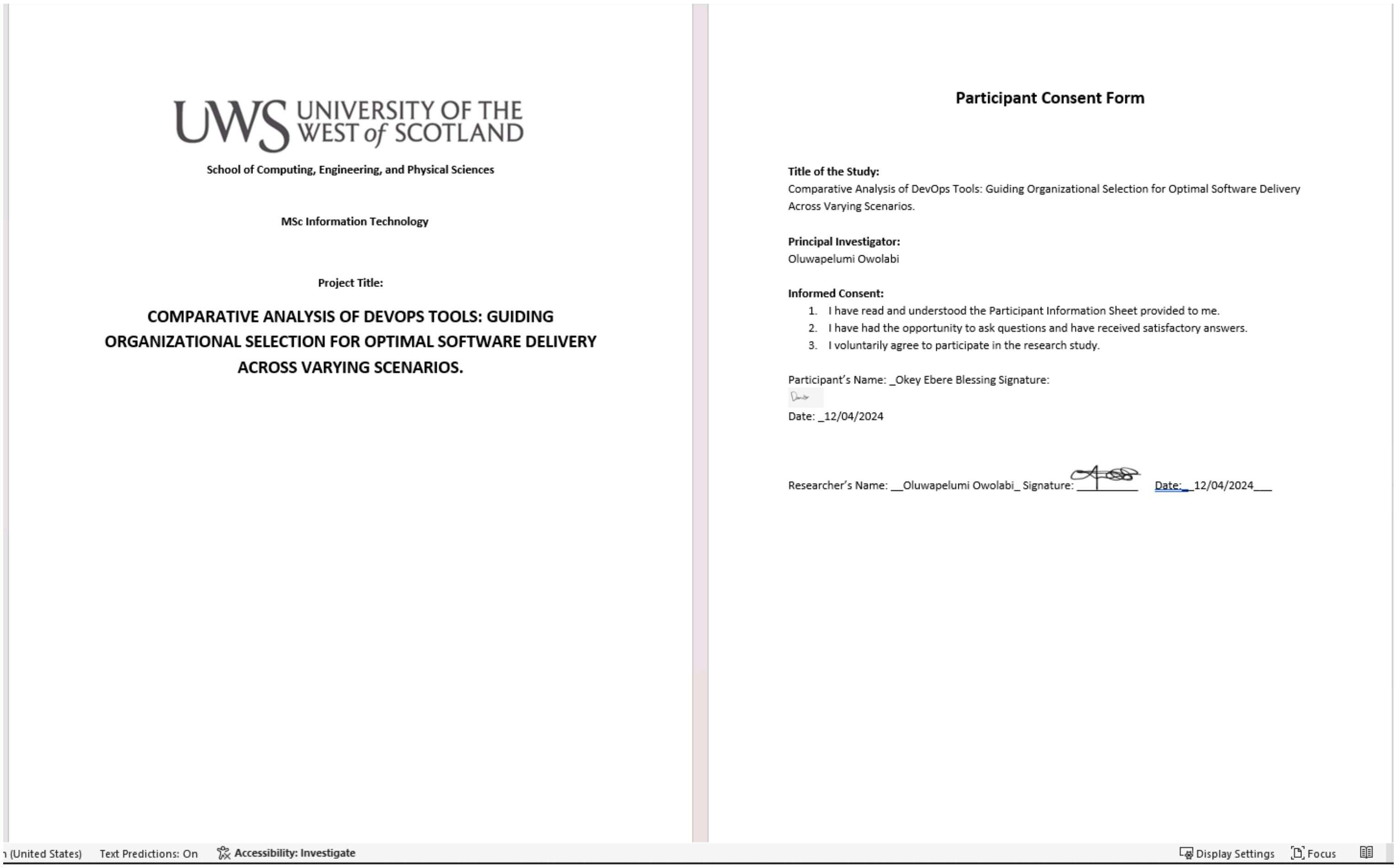
Principal Investigator:
Oluwapelumi Owolabi

Informed Consent:

1. I have read and understood the Participant Information Sheet provided to me.
2. I have had the opportunity to ask questions and have received satisfactory answers.
3. I voluntarily agree to participate in the research study.

Participant's Name: Terpase Gagher Signature: ________ Date: 26/03/2024

Researcher's Name Oluwapelumi Owolabi___Signature: ________ Date: 26/03/2024



UWS UNIVERSITY OF THE WEST *of* SCOTLAND

School of Computing, Engineering, and Physical Sciences

MSc Information Technology

Project Title:

COMPARATIVE ANALYSIS OF DEVOPS TOOLS: GUIDING ORGANIZATIONAL SELECTION FOR OPTIMAL SOFTWARE DELIVERY ACROSS VARYING SCENARIOS.

Participant Consent Form

Title of the Study:
Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios.

Principal Investigator:
Oluwapelumi Owolabi

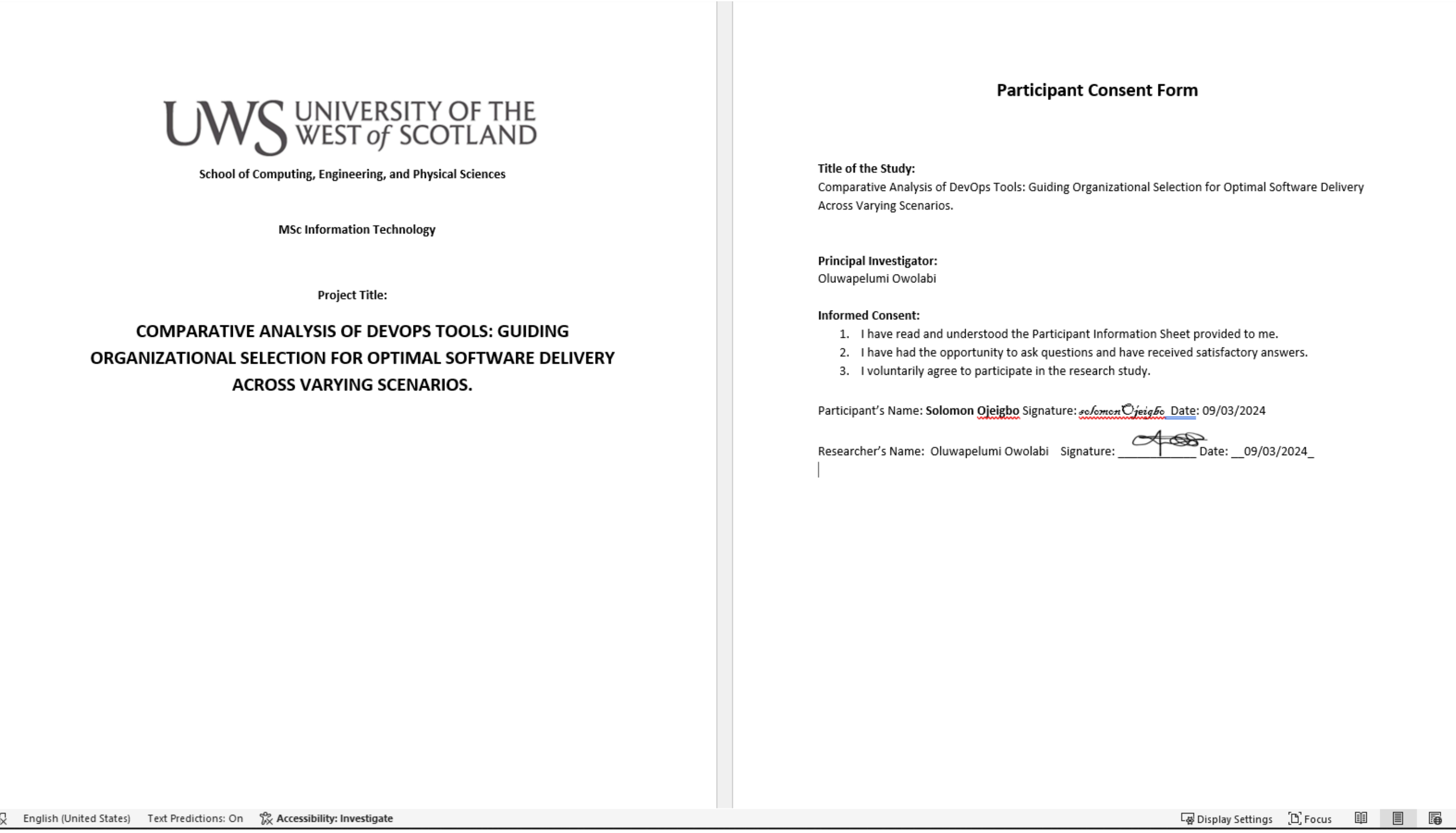
Informed Consent:

1. I have read and understood the Participant Information Sheet provided to me.
2. I have had the opportunity to ask questions and have received satisfactory answers.
3. I voluntarily agree to participate in the research study.

Participant's Name: _Okey Ebere Blessing Signature:

Date: _12/04/2024

Researcher's Name: __Oluwapelumi Owolabi_ Signature: ________ Date:_ _12/04/2024___



UWS UNIVERSITY OF THE WEST *of* SCOTLAND

School of Computing, Engineering, and Physical Sciences

MSc Information Technology

Project Title:

COMPARATIVE ANALYSIS OF DEVOPS TOOLS: GUIDING ORGANIZATIONAL SELECTION FOR OPTIMAL SOFTWARE DELIVERY ACROSS VARYING SCENARIOS.

Participant Consent Form

Title of the Study:
Comparative Analysis of DevOps Tools: Guiding Organizational Selection for Optimal Software Delivery Across Varying Scenarios.

Principal Investigator:
Oluwapelumi Owolabi

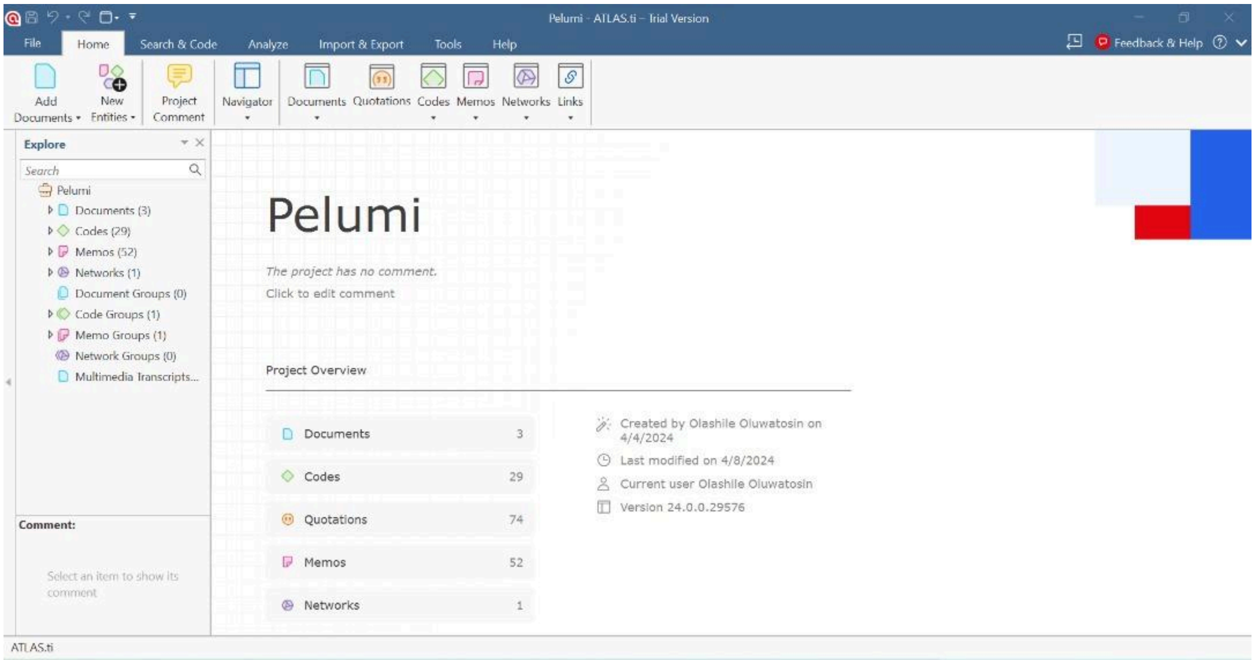
Informed Consent:

1. I have read and understood the Participant Information Sheet provided to me.
2. I have had the opportunity to ask questions and have received satisfactory answers.
3. I voluntarily agree to participate in the research study.

Participant's Name: **Solomon Ojeigbo** Signature: *solomon Ojeigbo* Date: 09/03/2024

Researcher's Name: Oluwapelumi Owolabi Signature: _____________ Date: __09/03/2024_

APPENDIX C - CODING REPORT FOR THE DATA ANALYSIS



- Pelumi
  - Documents (3)
  - Codes (29)
  - Memos (52)
  - Networks (1)
  - Document Groups (0)
  - Code Groups (1)
  - Memo Groups (1)
  - Network Groups (0)
  - Multimedia Transcripts...

Pelumi

The project has no comment.

Click to edit comment

Project Overview

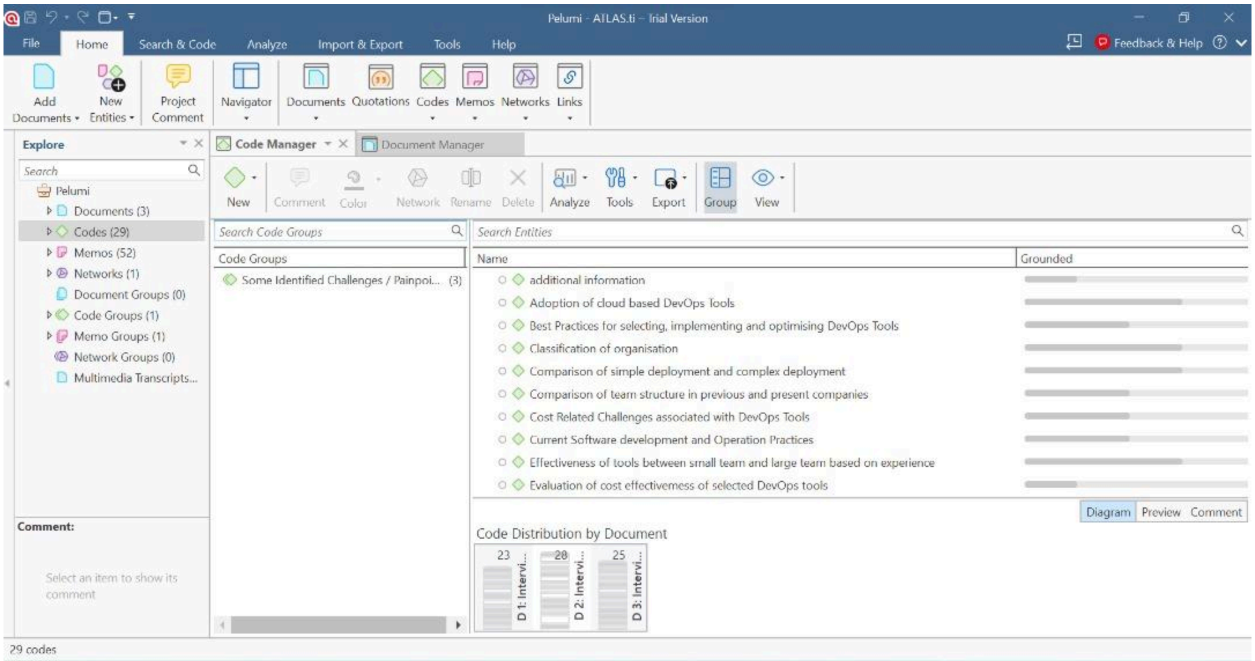
Documents	3
Codes	29
Quotations	74
Memos	52
Networks	1

Created by Olashile Oluwatosin on 4/4/2024

Last modified on 4/8/2024

Current user Olashile Oluwatosin

Version 24.0.0.29576



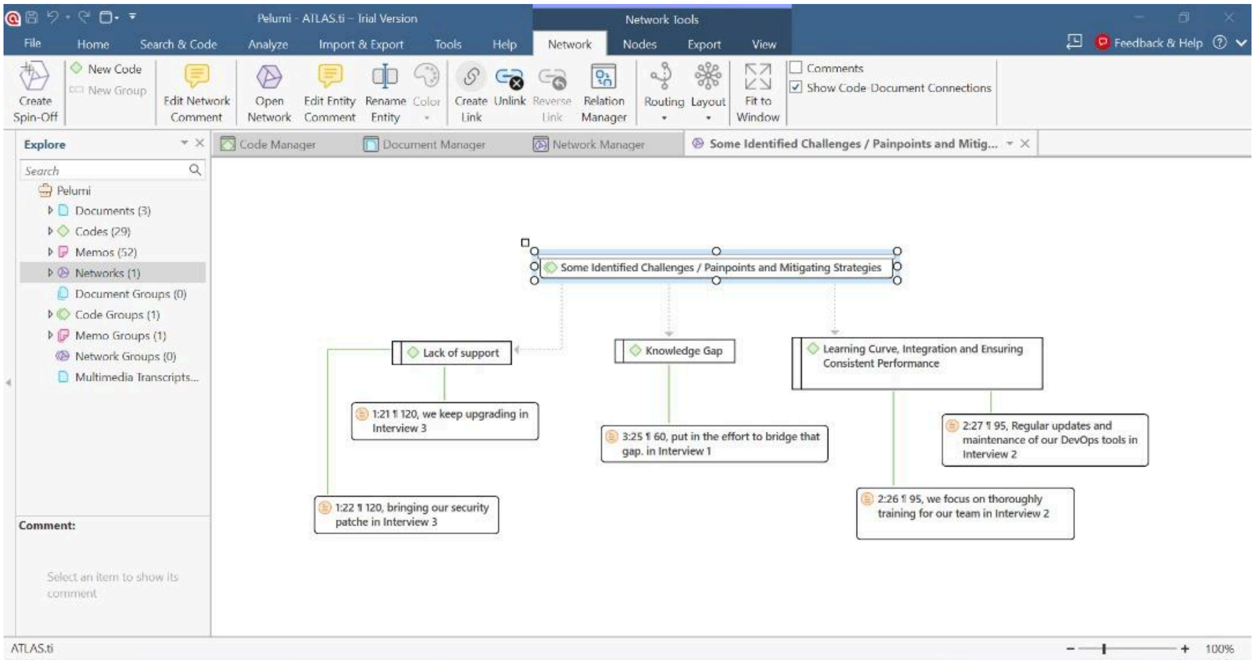
- Pelumi
  - Documents (3)
  - Codes (29)
  - Memos (52)
  - Networks (1)
  - Document Groups (0)
  - Code Groups (1)
  - Memo Groups (1)
  - Network Groups (0)
  - Multimedia Transcripts...

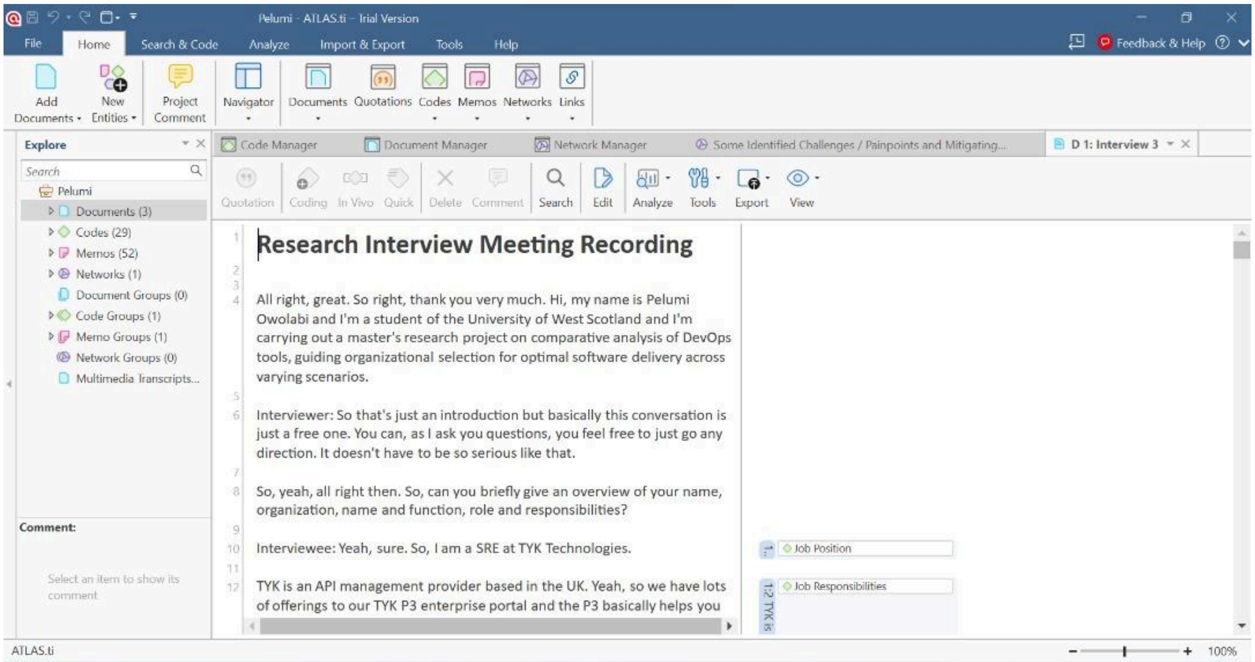
Code Groups	
Some Identified Challenges / Painpoi...	(3)

Name	Grounded
additional information	
Adoption of cloud based DevOps Tools	
Best Practices for selecting, implementing and optimising DevOps Tools	
Classification of organisation	
Comparison of simple deployment and complex deployment	
Comparison of team structure in previous and present companies	
Cost Related Challenges associated with DevOps Tools	
Current Software development and Operation Practices	
Effectiveness of tools between small team and large team based on experience	
Evaluation of cost effectivemess of selected DevOps tools	

Code Distribution by Document

D 1: Intervi...	D 2: Intervi...	D 3: Intervi...
23	28	25


Pelumi - ATLAS.ti - Trial Version
Network Tools
File
Home
Search & Code
Analyze
Import & Export
Tools
Help
Network
Nodes
Export
View
Feedback & Help
Create Spin-Off
New Code
New Group
Edit Network Comment
Open Network
Edit Entity Comment
Rename Entity
Color
Create Link
Unlink
Reverse Link
Relation Manager
Routing
Layout
Fit to Window
□ Comments
☑ Show Code Document Connections
Explore
Search
Pelumi
Documents (3)
Codes (29)
Memos (52)
Networks (1)
Document Groups (0)
Code Groups (1)
Memo Groups (1)
Network Groups (0)
Multimedia Transcripts...
Comment:
Select an item to show its comment
Code Manager
Document Manager
Network Manager
Some Identified Challenges / Painpoints and Mitig...
Some Identified Challenges / Painpoints and Mitigating Strategies
Lack of support
Knowledge Gap
Learning Curve, Integration and Ensuring Consistent Performance
1:21 ¶ 120, we keep upgrading in Interview 3
1:22 ¶ 120, bringing our security patche in Interview 3
3:25 ¶ 60, put in the effort to bridge that gap. in Interview 1
2:27 ¶ 95, Regular updates and maintenance of our DevOps tools in Interview 2
2:26 ¶ 95, we focus on thoroughly training for our team in Interview 2
ATLAS.ti
100%



Research Interview Meeting Recording

All right, great. So right, thank you very much. Hi, my name is Pelumi Owolabi and I'm a student of the University of West Scotland and I'm carrying out a master's research project on comparative analysis of DevOps tools, guiding organizational selection for optimal software delivery across varying scenarios.

Interviewer: So that's just an introduction but basically this conversation is just a free one. You can, as I ask you questions, you feel free to just go any direction. It doesn't have to be so serious like that.

So, yeah, all right then. So, can you briefly give an overview of your name, organization, name and function, role and responsibilities?

Interviewee: Yeah, sure. So, I am a SRE at TYK Technologies.

TYK is an API management provider based in the UK. Yeah, so we have lots of offerings to our TYK P3 enterprise portal and the P3 basically helps you

1 Interview 3

20 Codes:

○ additional information

1 Quotations:

1:20   199 in Interview 3

There are a lot of DevOps tools that do the same thing, right? Basically, find the easiest one to use. That's also a problem. Okay.

○ Adoption of cloud based DevOps Tools

3 Quotations:

1:19   193 in Interview 3

: I mean, we're always preparing for these changes. So in terms of software, in terms of the DevOps tools we are using, lots of these tools come with different versions, and you have to learn different versions to enjoy some of the benefits. Occasionally, in terms of security patches that have been done on newer versions, we're always moving at that pace, because we have an open-source software with us, and we're trying to keep adapting

2:25   217 in Interview 2

so my organization is, I've said that early, we are seeing that there's a huge thing that AI is doing, and we're trying to collaborate with Microsoft co-pilot, for example, okay, we're trying to add a huge collaboration in regards to that, because funny enough, AI might be able to do all of these things that we're talking about, but we need to be experts in that, because that is the change that many of us are looking at, right, there's some things I ask Microsoft co-pilots, right, because we have the license package for that, then it's going to tell me, even my slide, and so we're looking at the future of AI, how can DevOps integrate with AI, and how best can we meet to the change, because many organizations abroad are like, in other countries, are trying to dive into it, but we want to accept it, right, such as artificial intelligence, machine learning, design to enhance efficiency and productivity in software development and operations, you understand, so these are the things that we are seeing, so we are, like every one of us now, we have a course we are running on Microsoft co-pilots, what is the knowledge of AI, we're even trying to get like an AI specialist, but funny enough, we don't have them, so these are the opportunities for us to scale through, so that we'll be there before this organization dive, although we are already there, but we're seeing that the future of DevOps programming, coding, if case not taken, you just need to know what you

3:23 85 in Interview 1

: So- Yeah, we have plans in place to move, yeah.

○ **Comparison of simple deployment and complex deployment**

3 Quotations:

1:14 136 – 150 in Interview 3

: So, I mean, if you're trying to do something very small, definitely it's going to be easier for you to set up. But if you're doing something complex, then you need to have an idea of what you're trying to achieve and how you're going to get to that end point, right? So, you need to break it down into smaller pieces and then try to iterate. Say, okay, in terms of my size, I'm having a very complex work process.

First, I need to build my application, then probably build a Docker image, push that image with my registry, and then create manifest files to deploy that application and then finally deploy it. You see, it's very different than me just pushing and writing the CICD to basically just create an image. So, how it works in terms of the I'm trying to get the question, how it performs in simple versus complex.

It just depends on the setup, basically, of what you're trying to do.

Interviewer: Okay. So, basically, it's what you're trying to do because what I'm trying to understand is, just like you said, it's very easy to set up if what you're doing is just something small.

Interviewee: Basically, in more complex deployments, is it that these tools, do they maybe take more time to respond or do they... Definitely. Yeah, sure. I mean, as more complex as it gets, it's going to take a bit of more time.

For example, I use a CICD pipeline. If you're doing something as simple as just building a Docker image, that is going to take you, let's say, between two, three seconds, you're done. But if you want to go through the whole process that I explained, definitely, it's going to take you more.

It's going to take you up to like a minute for you to complete that. So, the more complex it is, the more time it's going to take. It consumes.

Yeah. All right. Thank you.

2:16 130 – 140 in Interview 2

ust like Jenkins and some other DevOps tools, like Git and the rest, in simple deployment pipelines, all right, these tools have performed very well, all right. Why in complex pipelines? They required more careful configuration and monitoring. All right, like now you're trying to automate some particular applications or software applications.

If it's a very small, maybe organization, just me and you or a startup, it's very effective, all right. But when that tools, like maybe you have several containers, like three containers, for example, Docker containers, all right, and you're trying to replicate some applications on those containers. So they are few, they are just like three, then it's very easy.

But when you are trying to orchestrate like 1,000, 2,000, you understand, applications and the rest, then you will see that sometimes the tools will just misbehave. You understand? The workload is too much. Yeah, the workloads, thank you.

That is the right word, the workload is too much and then just become slow. Sometimes you have to wait, you have to be patient for three minutes. You know, some of us just like something fast though.

Three minutes is not big. I was like, well, guys, guys, guys, be patient. It's going to work.

Let's just wait for it, you know. So that is it. So the larger it is, it affects, yeah.

3:19 74 – 75 in Interview 1

What I'm referring to is that complex deployments and simple deployments sound like straightforward deployments. The complexity of the build process is not regarding, and it is not based on the build process. The complexity is putting those resources or writing those scripts or writing those functionality that is needed to build that particular application. So that's something that is so complex. The dependencies that are needed to build a Java file, the ones that are used to build a Java application, and the ones needed to build a Python application are not the same.

So there's a streamline when it comes to complexity in both aspects. I don't really say the complexity is on the tooling aspects, but I feel it's on the source code aspects. Okay, all right.

○ **Comparison of team structure in previous and present companies**

2 Quotations:

1:13 128 in Interview 3

I would say it's a small team.

3:17 70 in Interview 1

No, this is bigger than the other.

○ **Cost Related Challenges associated with DevOps Tools**

2 Quotations:

1:16 167 – 176 in Interview 3

Yeah, sure. Like I said earlier, right, if you're going to pay for software, the time just keeps increasing because, I mean, they have their own additional costs as well, so that they tend to go to the end-users. So, AWS, for example, right, you can run an application on AWS and after a while, the customer version will no longer support it.

But if you run that version, the cost of running that version increases.

Okay. So that's one aspect of it.

Okay. So I mean, from someone who has not used AWS, the person doesn't know that at the point, he has to pay more, right?

So if the person just starts using AWS, they might say that every month and pay his amount. After a while, if that software you're running on that platform is no longer supported, you get to pay more. Hmm.

Hmm. That would be unexpected to someone who has not worked on, let's say, software or DevOps ecosystem for a while.

2:20 197 – 201 in Interview 2

: All right, okay, so now, this does not come in from us, all right, like most of the time when you support an organization, all right, they have done some projects for example, okay, and some of them fail to turn off their virtual machines. I'm just going to the basics, and these virtual machines are running server on Linux, right, and they have to do a test in the environment, and then the thing is working well, so they go to the re-environment, the re-virtual machine, for example, which the company is using, and they replicate what they have done in the other VM. So, the issue we find out that, you will be surprised that most of these, sorry, security, ICS or something, they are security experts in regards to all of these.

After they have done these deployments, they leave the other virtual machine to be running that they use for tests, and then the other one, the other one that is going to live, that is already working, we also, so they have it simultaneous operation, kubernetes is running on the test environment, kubernetes is running on the re-environment, to be honest, bro, they have two times the amount they need to spend. So, Microsoft or AWS, we drop the payment for their costs of the day, right? And then, before you know, they are trying to look at having meetings back to back to reduce the cost. That is the challenge. I found out, another one was pressing his phone, assessing his devices on his phone, fine, Microsoft had that, AWS had that, and the guy mistakenly handed his phone to someone else, that is laziness from his side, he is a security expert, I don't know why security experts have this mistake, it seems like he is not confident.

So, this guy left his phone, another smart guy used that phone, opened like 10 virtual machines and was running, what they call crypto mining, was mining seriously on those devices, but they never turned on Azure alerts, all right? So, those things were mining and they spent over 40,000 US dollars, all right. So that is just it it is just security, everything you are working on, security, DevOps guys, like DevSec, that is what we call DevSecOps. They do DevOps and also security, we don't have a lot of DevSec, also, I am going into that already, so security around your system, very important, like Cloud Watch for AWS, it is very important, it tells you when a particular item is idle, it tells you, “bro, these things are idle, turn it off”, so optimize costs and the rest, so, yeah.

○ **Effectiveness of tools between small team and large team based on experience**

3 Quotations:

1:15 159 in Interview 3

Okay. So, for startups, right, the goal is to usually to keep it simple and then to keep it cheaper, right? So, trying to be very cost effective in terms of the tools used for DevOps and then try to keep your architecture and the way you want to deploy your software and then run your application to be, let's say, smaller and simpler compared to a big organization where they have complex APIs, complex practices and more complex shipped systems.

2:18 152 – 162 in Interview 2

Okay, so like startups now, for example, well, the needs and preference for DevOps to differ from startups and established enterprises, like startups often prioritize agility and cost effectiveness, you understand? While established enterprise focus more on scalability and integration with existing systems, you understand? So that is just the major difference. That's the difference.

Yeah, startups prioritize agility and cost effectiveness. Like some startups, like what startups do? Baba, sorry, sorry to use that. But that is our local language, so you have to be used to it, right? It's okay, it's okay.

So you say, please, we can't buy Kubernetes, we can't use all these things. Cut the costs. Cut the costs. Which is fine. So we just have to go back and say, okay, guys, okay, let us tone down Kubernetes first. Let's just use these other tools.

Let's see what we can do. Let's help them strive with their business and the rest. And then sometimes we go into close them, like beginning with the owners of the products, like AWS or Azure.

We say these are startups, they are just starting, right? Is there a way you can optimize the costs? All right, yeah, so the pricing system. So they can say, okay, how many are they? What's their number? Okay, okay, these are the, just this one application. So that's why we have, that's why you have to partner with us because we have good partnership with them, yeah, the organization and then we can cut costs.

But for big enterprises, those they focus on the scalability and integration of their existing system. Can it work with Windows 17.0? Say, wow, really? So you guys are still operating here, right? Okay, let us go and ask the OEM and let's see what happens.

3:18 72 in Interview 1

The one thing about being a DevOps Engineer and moving from one position to another is that when you go to different environments, they use different tools. So, the tools we used then are different now, so I can't really compare.

○ **Experience of Compatibility Issues with tools**

4 Quotations:

1:9 80 – 84 in Interview 3

Yeah, I mean, occasionally, you get to have integration issues, right? Let's say, for example, I'm not going to give an illustration, right? You have, like, software running in your system, and then you're trying to move to new ways of doing stuff. Let's say you have a monolith application deployed, that you have a lot of users on it, and then you're trying to move to the new way of doing stuff, like using microservices, using containers, deploying to a contract, like, migration to, like, Kubernetes, right? So, most times, if you want to, like, do that migration from, like, migrating your customers from the monoliths to the new way of doing things you've set up, most times you're going to have a bit of issues integrating Kubernetes with the way you have normal VMs.

Okay. So, then you need to think more, okay, how, what more can I do? How do I get my new environment to serve the way the old environment was? So, that's when you start it, because if you deploy applications to a VM, all you need is a public IP, and then it works, right? No one can be able to access that application once it's deployed on that system, right? But for Kubernetes, it's different, right? You need to do stuff like services, you need to consider load balancers, you need to consider ingress. So, it's not, like, a straightforward stuff to do.

So, that's some of the issues with integrating if you're moving from an existing system, trying to integrate new tools and stuff like that. Okay. All right.

2:10 69 – 76 in Interview 2

Yeah, there's a time that I was working with, there's a particular project that I work on. I don't want to call the name. So we were having this issue.

I tried to make use of my Linux tools to install. I installed, I tried to Node 16, right? And I didn't, I was trying, I installed Node 16, it was successful. I'm trying to deploy it into the environment, but we're still having issue.

And I know that, okay, I'm using the old version. Where is the new version? But the documentation carried the old version. So I was following based on documentation.

So I have to tell my organization, come, we need a new version. Let's give it a try. Yeah, so we upgrade it and then it was working fine.

So version two is very important for you to be up to date.

2:11 78 – 89 in Interview 2

Yeah, like, you know, like example, if you're a programmer, there's what we call Node.js.

So this Node.js has several packages and several versions, all right? So one thing is that you need to be sure if this Node or JavaScript or .NET has the current version. Versions. In the cloud, things are being updated.

Okay. You know, cloud make use of, you don't have to bother about that. They update their systems.

Yeah. So for you to make things sync well, you need to ensure that you're updated. Even the packages in your Linux, before you start, you do packets to update the packets so that it's able to be compatible.

So all these packages like Node.js, JavaScript, .NET has to be updated for you to run those integrations.

Interviewer: Okay, so you're saying compatibility issues is largely caused by where you are using, where you are not using the updated versions, yeah?

Interviewee: Yeah, yeah, because some organizations, they are very good at old things. I'm sorry, I'm very sorry.

3:10 54 in Interview 1

No, no, no. No. I've not, I've not had any compatibility issues.

○ **Future Trends in cloud-based DevOps**

3 Quotations:

1:19 193 in Interview 3

: I mean, we're always preparing for these changes. So in terms of software, in terms of the DevOps tools we are using, lots of these tools come with different versions, and you have to learn different versions to enjoy some of the benefits. Occasionally, in terms of security patches that have been done on newer versions, we're always moving at that pace, because we have an open-source software with us, and we're trying to keep adapting

2:24 217 – 221 in Interview 2

Interviewee: Okay, for me, DevOps is expected to continue to grow, and more organization, you know, recognize the benefit of integrating development, and also operational terms, so my organization is, I've said that early, we are seeing that there's a huge thing that AI is doing, and we're trying to collaborate with Microsoft co-pilot, for example, okay, we're trying to add a huge collaboration in regards to that, because funny enough, AI might be able to do all of these things that we're talking about, but we need to be experts in that, because that is the change that many of us are looking at, right, there's some things I ask Microsoft co-pilots, right, because we have the license package for that, then it's going to tell me, even my slide, and so we're looking at the future of AI, how can DevOps integrate with AI, and how best can we meet to the change, because many organizations abroad are like, in other countries, are trying to dive into it, but we want to accept it, right, such as artificial intelligence, machine learning, design to enhance efficiency and productivity in software development and operations, you understand, so these are the things that we are seeing, so we are, like every one of us now, we have a course we are running on Microsoft co-pilots, what is the knowledge of AI, we're even trying to get like an AI specialist, but funny enough, we don't have them, so these are the opportunities for us to scale through, so that we'll be there before this organization dive, although we are already there, but we're seeing that the future of DevOps programming, coding, if case not taken, you just need to know what you need to do, ask co-pilots or the right questions, I'm very sorry for organizations who are not adopting this now, very sorry for DevOps engineers who are not adopting, thinking that you'll be sitting down and you're typing and say, no, no, no, the workspace is optimizing so fast, so how can you ask the right questions to this artificial intelligence, they draft out the whole, imagine, I can ask co-pilots, okay, look at what I want to automate for systems, can you give me all the step-by-step Linux command for me to orchestrate this program, so you know what they call place value, place value means that there's a particular place, he's going to put asterisk, then as a DevOps guy, you need to know what you need to put in that place to fit into your life.

Yeah, yeah, so that is where my organization is looking at, how can we, you know, we're going, collaborating, looking at even AI startups organization, we're trying to position ourselves so that we take advantage of, because DevOps, like we need to just wake up and see how we can integrate AI tools, yeah, on all of this. That's really good, because you didn't answer my next question, so I'll just go to closing remarks, so is there any other thing, any additional information or insights that you'd like to share, you know, with your experience concerning DevOps tools, just anything, and if you don't have it, it's still fine.

All right, so my own general talk is that the world is fast changing, all right, and there's a lot of transformation, I advise, even as you doing the research work, it's an excellent job, all right, I encourage you to do, to also do more in regards to asking questions and the rest, the world is fast changing, and we need to change how the world is changing in regards to technologies, all right, we need to see how we can take advantage of it, so my own is that I am advising small medium-scale organizations, big enterprise companies, don't just sit at the old system that you are used to, all right, that will bring problem, there's a project that I'm working on with one organization, for example, the imagine where you are using Windows version 1701, and there's this Windows 98, you're still making use of it, although it is working well, fine, but now updating that operating system for it to sync with some DevOps tools becomes a big problem, and they have important information in that, so we need to do extra work, we need to call on the OEM, please, we need 1607, so that we'll be upgrading gradually, you know, so I think the problem with this is that when you don't, when you're not using the latest versions, it doesn't help you as a DevOps engineer to focus on what really matters. So you keep on troubleshooting something that you don't even, you spend more time on a very little task.

3:24 86 – 97 in Interview 1

Because most of these old legacy tools have reached the end of their life. So I feel that for any organization that is planning to start up, I would advise going on cloud. Just move all your services to the cloud.

The way they're going, yeah, the cloud is the future. Because most organizations now are adopting that cloud system.

Yeah, when it comes to cost optimization, time-saving, and also having time to do the relevant things. Like, how will I really put it? There's a term for it: Focusing on more important things. I'm working while fully on-prem.

So, most of the time, we focus on less important things, on things that are not our core responsibility. But if you are on the cloud, you won't have time for that. You will have time to focus on what really matters.

That's the advice I would give to them.

Though they said the cloud is expensive, there is a long-term plan. Okay. You have the opportunity to speak with an AWS sales representative.

I know that they have reserved plans that can help you with the price. Just commit to their services. For example, I'll be using the service for 15 years, and they will cut the price for you. So, okay. Yeah.

It could be more cost-effective, but it's quite alright.

But it gives you the revenue or to generate more funds. More income, yes. Because your application is 99.999999 uptime.

You don't have any issues. You are generating funds. Yes, it will cost you money, but you generate twice what you spent.

It means you're able to afford what you are currently spending. Because I've had a series of issues, that all your hosting database is down and before you know it, almost all your services are not, are done also, you keep on getting escalations for customers because people are not able to use your products and your solutions.

It's a big issue because when you give, when you own that gap, you give people, you give your final consumers that thought of looking for other solutions that can solve that problem, and before you know it, the stability of that your product is minimal because why do people need Moniepoint now in Nigeria? You keep hearing "Moniepoint, Moniepoint." Yes, that doesn't make them the only switching company. There are other ones, but people are looking for things that will solve their problems quickly, saving them time