THE EFFECTIVENESS OF OPEN-SOURCE SOFTWARE VERSUS CLOSED-SOURCE SOFTWARE IN ZIMBABWE

AFRICA UNIVERSITY
A UNITED METHODIST RELATED UNIVERSITY

2025

ABSTRACT

This thesis investigates the relative merits of open source (OSS) and closed source software (CSS), evaluating their effects on a range of factors including price, security, innovation, and community involvement. Which model provides better performance and value under various conditions is a key question addressed by the study in the domains of software development and IT administration.

Open-source software has become increasingly popular because it is affordable and collaborative. It is distinguished by publicly accessible and customizable code. On the other hand, integrated solutions and strong support services are typically linked to closed source software, which has copyrighted and restricted source code. Using an integrative strategy, this research explores the advantages and disadvantages of both models.

A thorough literature study, case studies, and quantitative analysis are all part of the technique. A theoretical context for the analysis is provided by the literature review, which summarizes the results of previous studies on OSS and CSS. Real-world applications and results are illustrated through case studies of well-known OSS and CSS projects, like Microsoft Windows and the Linux operating system. Additionally, empirical data on the perceived efficacy of both software types is provided by a poll of end users, IT managers, and software engineers.

Results show that OSS frequently performs exceptionally well in terms of innovation and cost effectiveness. The collaborative development environment and the lack of license fees facilitate swift invention and adaptability. Additionally, because the code is open source, it is subject to thorough peer review, which can improve security by identifying and fixing vulnerabilities. But OSS's decentralized support structure can present problems, especially for non-technical consumers and businesses in need of dependable and continuous technical assistance.

On the other hand, CSS usually provides a more approachable interface along with specialized assistance and extensive documentation. This might be especially helpful for companies that value consistency and dependability over expense and customisation. Closed development, however, has the potential to inhibit creativity

and adaptability to user input. Reliance on a single vendor for security patches and upgrades can also be problematic, particularly if the seller stops supporting a product.

According to the study's findings, context affects how successful OSS is in comparison to CSS. Businesses in need of adaptable, affordable solutions and possessing technological know-how might gain more from OSS. On the other hand, entities who demand reliable assistance, user-friendliness, and cohesive solutions could discover that CSS works better. The thesis emphasizes how crucial it is to choose software strategically, taking into account the unique requirements, available resources, and organizational goals.

In the end, the research offers useful information for software industry decision-makers and advances a comprehensive understanding of the OSS and CSS concepts. This study helps organizations make decisions that are in line with their operational objectives and technological capacity by determining the situations in which each model flourishes.

Key Words

Open-Source Software Closed-Source Software Effectiveness
Proprietary Decentralized

Declaration

I declare that this dissertation is my original work except where sources have been cited and acknowledged. The work has never been submitted, nor will it ever be submitted to another university for the award of a degree.

Student's Full Name……...

Student's Signature…… Date……...……....

Supervisor's Full Name…...…….…….

Supervisor's Signature…… Date….…….……...

Copyrighted

Acknowledgments

I extend my heartfelt gratitude to the following individuals and organizations who have contributed to the completion of this dissertation:

1. **Mr. Makambwa:** My devoted dissertation supervisor, whose advice, experience, and continuous support helped shape this work.
2. **Mr. Albert Nyamajiwa:** My beloved father, whose encouragement, wisdom, and belief in my abilities have fuelled my academic journey.
3. **The Ministry of Information Communication Technology, Postal and Courier Services:** I am grateful for the resources, data, and insights provided by this esteemed organization. Your commitment to advancing technology and communication has significantly impacted my research.
4. **Friends and Family:** To all those who stood by me during this journey, offering encouragement, feedback, and understanding. I am deeply thankful.
5. **My Academic Advisors:** Their patience, constructive feedback, and mentorship have been invaluable.

Dedication

To my beloved mother, Mrs Alice Kupeta-Nyamajiwa,

In loving memory of your unwavering support, encouragement, and boundless love. Your strength and resilience continue to inspire me every day. Though you are no longer with us, your spirit lives on in my pursuit of knowledge.

Forever in my heart,

Arthur

List of Acronyms and Abbreviations

OSS……Open-Source Software

CSS……Closed-Source Software

OSI……Open-Source Initiative

FSF……Free Software Foundation

GPL……...General Public License

GNU……Gnu's Not Unix

RBV……. Resource-Based View

CS……. Cyber-Security

ADM……Application Development Management

IDM……...Infrastructure Development Management

MICTPCS……Ministry of Information Communication Technology Postal & Courier Services

Table of Contents

List of Tables

Table 1. IT Employees in the MICTPCS organisation

IT DEPARTMENT	NUMBER OF EMPLOYEES
Cyber-security	16
Infrastructure Development Management	14
Application Development Management	17

Table 2. Selected Research Participants

NAME	DEPARTMENT	EMAIL	CONTACT NO.

Table 3. Selected Research Participants

Name	Department	Email	Contact No.
Alice Chirara	CS	chiraraalice@gmail.com	0771092880
Ben Makava	IDM	kingben67@yahoo.com	0717439995
Clara Marizani	ADM	clarmar@gmail.com	0772567806
Vernon Mushambi	CS	vxshambi@gmail.com	0771457496
Emily Dube	IDM	madhuvee01@yahoo.com	0777654110

Panashe Bobojani	ADM	panashebob@gmail.com	0717667734
Nyasha Nyamvura	CS	nyashanyamvura@gmail.com	0776540334
Simbarashe Fani	IDM	simbafani@gmail.com	0774178641
Munashe Kavhuya	ADM	kavmun@yahoo.com	0774345660
Elias Machaka	CS	emachaka@gmail.com	0717561229
Kelly Musara	IDM	musarak@yahoo.com	0774778797
Ropafadzo Taviringwa	ADM	taviringwarop@gmail.com	0776311452
Van D Dzobo	CS	dzobovand@gmail.com	0771657682

Table 4. Mean Scores and Standard Deviations for OSS and CSS Effectiveness

Dimension	Software Type	Mean Score	Standard Deviation
Performance	OSS	3.77	0.81
	CSS	4.15	0.72
Security	OSS	3.46	0.93
	CSS	4.23	0.69
Cost Effectiveness	OSS	4.46	0.64
	CSS	3.15	0.88
User Satisfaction	OSS	3.85	0.78
	CSS	4.08	0.70

List of Appendices

Appendix 1: Questionnaire for the Survey

Introduction

Thank you for participating in this important research study. This questionnaire aims to understand factors that contribute to unethical behaviour in workplaces. Your honest responses are crucial to this research.

1. **Demographics (Please select all that apply)**

Age:

- ☐ 18-24 years old
- ☐ 25-34 years old
- ☐ 35-44 years old
- ☐ 45-54 years old
- ☐ 55+ years old

Gender:

- ☐ Male
- ☐ Female

Occupation: _____

Department: _____

2. **Software Usage:**

How frequently do you use software applications (daily, weekly, monthly)?

- ☐ Daily
- ☐ Weekly
- ☐ Monthly

Which types of software do you use most often (e.g., operating systems, productivity tools, databases)?

_____

_____

_____

_____

3. Experience with OSS and CSS:

Have you used open-source software before? (Yes/No)

Yes	
No	

Have you used closed-source software before?

Yes	
No	

Which specific OSS or CSS products have you used?

_____

_____

_____

_____

4. Factors Influencing Software Choice:

What factors are most important to you when choosing software? (Cost, security, features, community support, etc.)

_____

_____

_____

_____

How do these factors influence your preference for OSS or CSS?

_____

_____

_____

_____

_____

5. Perceived Effectiveness:

On a scale of 1 to 5, how effective do you find open-source software? (1 = Not effective, 5= Highly effective)

1=Not Effective	2=Slightly Effective	3=Just Effective	4=Very Effective	5=Extremely Effective

On the same scale, rate the effectiveness of closed-source software.

1=Not Effective	2=Slightly Effective	3=Just Effective	4=Very Effective	5=Extremely Effective

6. Security Considerations:

Do you believe open-source software is more secure than closed-source software? (Yes/No)

Yes	
No	

What security features do you consider essential in software?

_____

_____

_____

_____

_____

_____

_____

_____

7. Customization and Flexibility:

How important is the ability to customize software to your work or personal needs, on a scale of 1 to 5?

1=Not Effective	2=Slightly Effective	3=Just Effective	4=Very Effective	5=Extremely Effective

Have you ever modified open-source software to suit your requirements? If Yes, tell us how?

_____

_____

_____

_____

_____

8. Community Support:

Have you ever sought help from an online community (forums, blogs, etc.) related to open-source software?

Yes	
No	

How satisfied are you with the community support for OSS?

1=Not Satisfying	2=Slightly Satisfying	3=Just Satisfying	4=Very Satisfying	5=Extremely Satisfying

9. Challenges and Concerns:

What challenges have you faced while using open-source software?

_____
_____
_____
_____
_____
_____

Are there any concerns you have about closed-source software?

_____
_____
_____
_____
_____
_____

10. Preferred Licensing Model:

Which licensing model do you prefer: open-source (free and open) or closed-source (proprietary)?

OSS	
CSS	

Why do you prefer that model?

_____
_____
_____
_____
_____
_____

11. Future Trends:

Do you think the adoption of open-source software will increase in the future?

Yes	
No	

Why or why not?

_____
_____
_____
_____
_____
_____

Appendix 2: Interview guide

1. **Introduction and Demographics:**

   Begin by introducing yourself and explaining the purpose of the interview.

   Collect basic demographic information:

   - Name
   - Role or occupation
   - Industry or field of expertise
   - Experience with software development or usage

2. **Software Usage and Preferences:**

   Ask about the participant's experience with both OSS and CSS:

   - Have they used open-source software before?
   - Have they used closed-source software before?

   Explore their preferences:

   - Which type of software do they prefer, and why?

3. **Factors Influencing Software Choice:**

Inquire about the factors that influence their software choices:

- What criteria do they consider when selecting software? (Cost, security, features, etc.)
- How do these criteria differ for OSS and CSS?

4. **Effectiveness and Performance:**

   Assess their perception of effectiveness:

   - How effective do they find open-source software in their work or daily life?
   - How effective do they find closed-source software?

   Explore specific use cases or scenarios where effectiveness matters most.

5. **Security Considerations:**

   Discuss security aspects:

   - Do they believe open-source software is more secure than closed-source software? Why or why not?
   - Have they encountered security challenges with either type of software?

6. **Customization and Flexibility:**

   Explore customization options:

   - How important is the ability to customize software to their specific needs?
   - Have they ever modified open-source software to tailor it to their requirements?

7. **Community Support and Collaboration:**

   Inquire about community involvement:

   - Have they sought help from online communities related to open-source software?

- How valuable is community support in their experience?

8. **Challenges and Concerns:**

   Ask about any challenges they've faced:

   - What difficulties have they encountered while using open-source or closed-source software?
   - Are there any specific concerns they have about either type?

9. **Preferred Licensing Model:**

   Discuss licensing preferences:

   - Which licensing model do they prefer: open-source or closed-source?
   - Why do they lean toward one over the other?

10. **Future Trends and Predictions:**

    Explore their thoughts on the future:

    - Do they anticipate changes in the adoption of open-source software?
    - What trends do they foresee in the software industry?

THE EFFECTIVENESS OF OPEN-SOURCE SOFTWARE VERSUS CLOSED-SOURCE SOFTWARE

BY

ARTHUR TANAKA NYAMAJIWA

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF BACHELOR OF SCIENCE HONOURS IN SOFTWARE ENGINEERING IN THE COLLEGE OF ENGINEERING AND APPLIED SCIENCES

2025

AFRICA UNIVERSITY

(A United Methodist-Related Institution)

CHAPTER 1: INTRODUCTION

1.0 Chapter Introduction

Within the constantly changing field of technology, software development is essential for promoting creativity, productivity, and expansion into new markets. For developers, companies, and regulators, deciding between open-source software (OSS) and closed-source software (CSS) has become crucial. This chapter lays the groundwork for a thorough investigation into the relative merits of OSS and CSS, covering their histories, distinguishing traits, and the variables affecting their uptake and functionality.

1.1 Background to the Study

In the last few decades, the software industry has gone through a lot of changes. This is because technology has changed so quickly and more and more industries are relying on digital solutions. At the heart of this change is the ongoing battle between closed-source software (CSS) and open-source software (OSS), two opposite ideas that have affected how software is distributed, used, and made (Raymond, 1999).

1.1.1 Open-Source Software

The code for open-source software is available to everyone, so anyone can read it, change it, and share it. The transparency makes it possible for writers from around the world to work together to make the program better. Officially known as the Open-Source Initiative (OSI), open-source software can be used, changed, and shared by anyone. Some well-known examples of open software are the Apache HTTP Server, the Mozilla Firefox web browser, and the Linux operating system (Wheeler, 2007).

Open-source software (OSS) has been around since the early days of computers, when experts and engineers shared code all the time. In 1985, Richard Stallman started the Free Software Foundation (FSF). This was the first step towards making the open-source movement official in the late 20th century (Stallman, 2002). The FSF fought for software freedom and stressed how important it was for users to be

able to install, run, look at, change, and share software. The Free Software Foundation (2023) says that this effort led to the creation of the GNU General Public License (GPL), which is now one of the most important licenses for open-source software.

Many online tools for working together and the growth of the internet in the late 1990s and early 2000s helped make open-source software (OSS) possible. Open-source projects like the Apache HTTP Server and the Linux operating system have shown that it is possible to make safe, reliable, and high-quality software (Mockus, Fielding, & Herbsleb, 2002). An exact definition and structure for open-source licensing were given by the founding of the Open-Source Initiative (OSI) in 1998, which gave the movement more credibility and promoted its broad acceptance (OSI, 2023).

1.1.2 Closed-Source Software

Closed-source software, which is also called exclusive software or sealed-source, on the other hand, doesn't let anyone see its source code. The program can only be changed and shared by the people who made it or by organisations that have been given permission to do so. This method is often used by businesses to protect their intellectual property and keep track of who makes and shares software. iOS from Apple, Adobe Photoshop, and Windows from Microsoft are all well-known examples of closed-source software (Cusumano, 2004).

People who own proprietary software, also known as "closed-source" software, can't change its source code. The program can only be shared and changed by the people who made it or by companies that have been accepted. This is a common way for companies to protect their intellectual property and keep track of who makes and distributes software (Cusumano, 2004). That the closed-source approach became more popular in the 1980s and 1990s was helped by the rise of commercial software companies. Companies like Apple and Microsoft like proprietary licenses because they protect their intellectual property and help them make money from their software goods (Cusumano, 2004). The software industry widely used the closed-source approach because these companies were successful and showed that it could work (Cusumano, 2004).

1.2 Statement of the Problem

Even though both OSS and CSS have been used and improved a lot, we still don't fully understand how well they work in different situations. Businesses and organisations have a hard time making smart choices because there is a lot of conflicting information and different views about these software models. Lack of real-world data and thorough analysis makes it even harder to make decisions, which could lead to waste and less-than-ideal software acceptance.

1.3 Research Objectives

The main objective of this research is to evaluate the effectiveness of OSS vs CSS. To achieve this main objective, the following specific objectives have been drawn:

- ✓ To measure and compare the cost of buying and maintaining OSS or CSS.
- ✓ To assess the strengths and security flaws of both OSS and CSS.
- ✓ To determine the degree of user satisfaction both models give under the same conditions.
- ✓ To analyse the role of community support in the development and maintenance of OSS and CSS.

1.4 Research Questions

a) What are the financial differences between open-source and closed-source software adoption for businesses?

b) What is the difference between open-source and closed-source software in terms of security flaws and strengths?

c) What distinguishes open-source software from closed-source software in terms of freedom and customisation options?

d) In contrast to closed-source software, how does community support affect the creation and upkeep of open-source software?

1.5 Hypotheses

H1: OSS is more cost-effective than CSS: This hypothesis is based on the fact that OSS is typically free to use and modify, while CSS often involves licensing fees.

H2: CSS is more secure than OSS: This hypothesis is based on the belief that the closed nature of CSS makes it more difficult for malicious actors to find and exploit vulnerabilities.

H3: OSS is more flexible than CSS: This hypothesis is based on the ability of users to modify OSS to suit their specific needs.

H4: CSS provides better support than OSS: This hypothesis is based on the fact that CSS is typically backed by professional companies that provide dedicated support.

H5: OSS fosters more innovation than CSS: This hypothesis is based on the collaborative nature of OSS development, which allows for a wide range of ideas and contributions.

1.6 Significance of the Study

The significance of this study lies in its potential to provide a comprehensive understanding of the effectiveness of open-source software versus closed-source software. This understanding can guide software developers, companies, and individual users in making informed decisions about the type of software that best suits their needs.

For software developers, this study can provide insights into the strengths and weaknesses of OSS and CSS, helping them choose the right platform for their projects. For companies, understanding the effectiveness of OSS and CSS can help in strategic decision-making related to software procurement and development. For individual users, this study can help them choose the right software based on their specific needs and constraints.

Furthermore, this study can contribute to the existing body of knowledge in the field of software engineering by providing empirical evidence on the effectiveness of OSS and CSS. The findings of this study could be used as a basis for future research in this area.

Lastly, the recommendations provided in this study based on the findings can serve as a practical guide for stakeholders in the software industry. These recommendations can help in improving the quality, cost-effectiveness, and innovation in software development.

1.7 Delimitations of the Study

1.7.1 Software Types Covered: This research will concentrate on a few key software categories, including web servers, productivity tools, and operating systems. It won't include every kind of program that could exist, which could restrict how broadly applicable the results can be.

1.7.2 Geographical Focus: The study will mostly look at how software is used and how effective it is in Zimbabwe. Insights from other regions with distinct technical environments and adoption trends might not be included in this focus.

1.7.3 Time Frame: Information and patterns from a specific time frame, such the last ten years, will be examined in this research. Because of this temporal restriction, historical trends or more current changes that occurred before this time frame will not be taken into account.

1.7.4 Evaluation Standards: Cost, security, adaptability, and community support are just a few of the predetermined standards that will be used to gauge how effective OSS and CSS are. There won't be a major emphasis on other possible considerations like user satisfaction or environmental impact.

1.7.5 Data Sources: Industry reports, case studies, and previously published works are examples of secondary data sources that will be used in this study. There is going to be data collection using the methods of interviews and questionnaires to further foster the research.

1.7.6 Technological Context: The study will take into consideration the state of technology at the moment and might not take future developments or shifts in the software sector into account, which could have an impact on how well OSS and CSS work.

1.7.7 Comparative Analysis: OSS and CSS will be compared in the context of particular use cases and situations in this study. It won't offer an exhaustive comparison that takes into account every scenario and application that could exist.

1.8. Limitations of the Study

1.8.1 Analysis's Scope: The number of software programs examined could put restrictions on the investigation. It's possible that concentrating on particular software categories such as web servers or operating systems will not adequately convey the benefits of OSS and CSS for all software kinds.

1.8.2 Data Availability: It might be difficult to get thorough and trustworthy information about the security, cost, and performance of both CSS and OSS. The breadth of the research may be limited if proprietary software vendors choose not to reveal specific information.

1.8.3 Bias in Sources: The study can use secondary data sources, which raises the possibility of bias. For example, biased viewpoints from sources supporting closed-source or open-source software may compromise the study's objectivity.

1.8.4 Technological Developments: Because technology is developing so swiftly, results could become old very quickly. The relative efficacy of OSS and CSS may change as a result of new advancements in pricing structures, community support, and software security.

1.8.5 User Experience Variability: Depending on the user's experience, software effectiveness might vary greatly. Users' perceptions of the software's effectiveness may differ depending on their degree of experience.

1.9 Dissertation Structure

This dissertation is organized into five comprehensive chapters, each focusing on different aspects of the research on the effectiveness of open-source software (OSS) versus closed-source software (CSS) within the Ministry of Information Communication Technology, Postal and Courier Services. The structure is designed

to guide the reader systematically through the study, from the background and literature review to the methodology, data analysis, and conclusions.

Chapter 2: Review of Related Literature

Chapter 2 provides an extensive review of the existing literature related to OSS and CSS. This chapter serves several key functions:

- **Theoretical Framework**: Establishes the theoretical foundations of the study by discussing relevant theories and models that explain the adoption and effectiveness of OSS and CSS.
- **Previous Research**: Summarizes and critiques previous studies on OSS and CSS, highlighting their findings, methodologies, and limitations.
- **Key Concepts**: Defines and discusses critical concepts and terms relevant to the research, such as software effectiveness, open-source, closed-source, and software metrics.
- **Comparative Analysis**: Presents a comparative analysis of the strengths and weaknesses of OSS and CSS based on the literature, providing a context for the current research.
- **Research Gaps**: Identifies gaps in the existing literature that the current study aims to address, justifying the need for this research.

Chapter 3: Methodology

Chapter 3 outlines the research design and methods used to investigate the effectiveness of OSS versus CSS within the organization. It includes detailed descriptions of the following subtopics:

- **Research Design**: Describes the overall approach and strategy used in the research, including the rationale for choosing a comparative case study design.
- **Population Sampling**: Discusses the target population and the sampling methods used to select participants, including the sample size of 13 employees from different departments within the Ministry.

- **Data Collection Instruments**: Details the instruments used to gather data, specifically questionnaires and interviews, and explains their development, testing, and administration.
- **Data Collection Procedure**: Provides a step-by-step account of how data was collected, including preparation, distribution, follow-up, and collection of responses.
- **Analysis and Organization of Data**: Explains the methods used to analyse both quantitative and qualitative data, including statistical techniques for questionnaire data and thematic analysis for interview data.
- **Ethical Considerations**: Outlines the ethical protocols followed to ensure the confidentiality, anonymity, and voluntary participation of all respondents.

Chapter 4: Data Presentation, Analysis, and Interpretation

Chapter 4 focuses on the presentation, analysis, and interpretation of the collected data. This chapter is structured as follows:

- **Data Presentation**: Organizes and presents the raw data collected from questionnaires and interviews in a clear and comprehensible format, using tables, charts, and graphs where appropriate.
- **Quantitative Analysis**: Conducts a detailed statistical analysis of the questionnaire data, including descriptive and inferential statistics, to identify patterns and significant differences between OSS and CSS.
- **Qualitative Analysis**: Provides a thematic analysis of the interview data, identifying key themes and insights related to the effectiveness of OSS and CSS.
- **Interpretation of Findings**: Interprets the results in the context of the research questions and objectives, discussing the implications of the findings for the effectiveness of OSS versus CSS.
- **Comparison with Literature**: Compares the findings with previous research discussed in Chapter 2, highlighting consistencies and discrepancies.

Chapter 5: Summary, Conclusions, and Recommendations

Chapter 5 concludes the dissertation by summarizing the key findings, drawing conclusions, and making recommendations based on the research. It includes the following sections:

- **Summary of Findings**: Provides a concise summary of the major findings from the data analysis, highlighting the most significant results.
- **Conclusions**: Draws conclusions based on the findings, addressing the research questions and objectives outlined in the introduction.
- **Recommendations**: Offers practical recommendations for the Ministry of Information Communication Technology, Postal and Courier Services on the adoption and implementation of OSS and CSS, based on the study's findings.
- **Implications for Practice and Policy**: Discusses the broader implications of the findings for practitioners and policymakers in the field of information technology and software management.
- **Limitations and Future Research**: Acknowledges the limitations of the study and suggests areas for future research to further investigate the effectiveness of OSS versus CSS.

1.10 Chapter Summary

In this introductory chapter, the stage has been set for the investigation. The motivations behind studying open-source and closed-source software have been explored, the objectives outlined, and all essential terms defined. Moving forward, each subsequent chapter will enhance understanding and contribute to the ongoing discourse in software development.

CHAPTER 2: REVIEW OF RELATED LITERATURE

2.1 Chapter Introduction

This chapter provides an in-depth review of the literature related to the effectiveness of open-source software (OSS) versus closed-source software (CSS). The chapter begins with an overview of the relevant theoretical frameworks, followed by a detailed discussion on the application and relevance of these frameworks to the current study. The objective is to establish a comprehensive understanding of the existing research, highlight gaps, and set the stage for the empirical investigation that follows.

2.2 Theoretical Framework

Theoretical frameworks form the backbone of academic research, offering structured perspectives and methodologies for understanding complex phenomena. In the context of this study, three primary theoretical frameworks are particularly relevant: the Diffusion of Innovations Theory, the Resource-Based View, and the Theory of Planned Behaviour. Each framework offers unique insights into the adoption, utilization, and strategic value of OSS and CSS.

2.2.1 Diffusion of Innovations Theory

The Diffusion of Innovations Theory, conceptualized by Everett Rogers (2003), provides a robust model for understanding how new ideas and technologies spread within a social system. According to Rogers, the adoption process is influenced by five key attributes of innovation: relative advantage, compatibility, complexity, trialability, and observability.

- **Relative Advantage**: This refers to the degree to which an innovation is perceived as better than the existing alternatives. OSS often boasts advantages such as cost savings, customization, and community support, while CSS may offer advantages in terms of stability, vendor support, and integrated features (Rogers, 2003).
- **Compatibility**: This measures how well an innovation fits with the potential adopters' existing values, past experiences, and needs. OSS may be more

compatible with organizations seeking flexibility and control, whereas CSS may align better with those prioritizing reliability and comprehensive support (Rogers, 2003).

- **Complexity**: This attribute assesses how difficult the innovation is to understand and use. OSS can be perceived as more complex due to the need for technical expertise, while CSS might be considered easier to use due to vendor-provided support and documentation (Rogers, 2003).
- **Trialability**: This refers to the extent to which an innovation can be experimented with on a limited basis. OSS, being freely available, often allows extensive trial and experimentation, which can drive adoption among tech-savvy users (Rogers, 2003).
- **Observability**: This involves the extent to which the results of an innovation are visible to others. Success stories and case studies of OSS implementations can significantly influence its adoption (Rogers, 2003).

By applying these attributes, the Diffusion of Innovations Theory helps explain the varying rates of adoption for OSS and CSS across different user groups and contexts.

2.2.2 Resource-Based View

The Resource-Based View (RBV) of the firm, articulated by Barney (1991), posits that a firm's sustainable competitive advantage is derived from its ability to acquire and manage valuable, rare, inimitable, and non-substitutable (VRIN) resources. In the context of software, both OSS and CSS can be viewed as strategic resources that offer different sets of advantages.

- **Valuable**: Software that enhances efficiency, reduces costs, or improves customer satisfaction is considered valuable. OSS is often valued for its cost-effectiveness and flexibility, whereas CSS is valued for its stability and comprehensive support (Barney, 1991).
- **Rare**: Resources that are not widely available to competitors can provide a competitive edge. Proprietary features and exclusive capabilities often make CSS rare, while the unique collaborative development model of OSS can also be seen as a rare resource (Barney, 1991).

- **Inimitable**: Resources that are difficult to replicate by competitors are inimitable. The community-driven innovation in OSS and the proprietary technologies in CSS can both be inimitable under different circumstances (Barney, 1991).
- **Non-Substitutable**: Resources that cannot be easily replaced by other resources provide a non-substitutable advantage. For some organizations, the customizability of OSS might be non-substitutable, while others might find the integrated solutions of CSS irreplaceable (Barney, 1991).

The RBV framework helps in understanding how organizations leverage OSS and CSS to build and sustain competitive advantages, focusing on their strategic value.

2.2.3 Theory of Planned Behaviour

The Theory of Planned Behaviour (TPB), developed by Ajzen (1991), is instrumental in predicting and understanding human behaviour in specific contexts. TPB posits that behaviour is directly influenced by behavioural intentions, which are shaped by three key factors: attitudes toward the behaviour, subjective norms, and perceived behavioural control.

- **Attitudes Toward the Behaviour**: This involves the positive or negative evaluations of performing the behaviour. Users' attitudes towards OSS might include positive perceptions of its flexibility and cost savings, while attitudes towards CSS might reflect its perceived reliability and support (Ajzen, 1991).
- **Subjective Norms**: These are the perceived social pressures to perform or not perform the behaviour. Industry standards, peer recommendations, and community influence can play significant roles in shaping the subjective norms around the adoption of OSS or CSS (Ajzen, 1991).
- **Perceived Behavioural Control**: This refers to the perceived ease or difficulty of performing the behaviour, influenced by past experiences and anticipated obstacles. Users might perceive greater control with OSS due to its modifiability, while CSS might be preferred for its user-friendly interfaces and vendor support (Ajzen, 1991).

By applying TPB, this study can gain insights into the motivational factors that drive the adoption and usage of OSS and CSS, particularly in terms of user satisfaction and organizational decision-making.

2.3 Relevance of the Theoretical Framework to the Study

The selected theoretical frameworks are crucial for this study as they provide a comprehensive lens through which the effectiveness of OSS versus CSS can be analysed. Each framework offers unique insights that enhance our understanding of the various dimensions of software effectiveness.

2.3.1 Relevance of Diffusion of Innovations Theory

The Diffusion of Innovations Theory is particularly relevant for analysing how OSS and CSS are adopted by different user groups and industries. By understanding the attributes that influence adoption such as relative advantage, compatibility, complexity, trialability, and observability, this study can identify the factors that contribute to the effectiveness of OSS and CSS in various contexts. For instance, the trialability and observability of OSS might lead to higher adoption rates among innovators and early adopters, while the perceived reliability of CSS might appeal to the late majority and laggards (Rogers, 2003).

2.3.2 Relevance of Resource-Based View

The Resource-Based View provides a strategic perspective on the value of OSS and CSS as organizational resources. By examining how these software types contribute to competitive advantage, this study can evaluate their effectiveness in terms of cost savings, innovation, and operational efficiency. For example, the cost-effectiveness and flexibility of OSS can be critical resources for startups and small businesses, while the integrated solutions and vendor support of CSS might be more valuable for large enterprises with complex IT needs (Barney, 1991).

2.3.3 Relevance of Theory of Planned Behaviour

The Theory of Planned Behaviour is essential for understanding the behavioural intentions behind the adoption and use of OSS and CSS. By exploring factors such as attitudes, subjective norms, and perceived behavioural control, this study can gain insights into the motivations and barriers faced by users and organizations. For instance, positive attitudes towards the customizability of OSS, combined with peer

influence and perceived control over software modifications, can drive its adoption. Conversely, the perceived ease of use and vendor support associated with CSS might lead to higher user satisfaction and loyalty (Ajzen, 1991).

2.4 Chapter Summary

This chapter has provided a comprehensive review of the theoretical frameworks relevant to the study of OSS and CSS effectiveness. The Diffusion of Innovations Theory, Resource-Based View, and Theory of Planned Behaviour offer valuable insights into adoption patterns, strategic value, and user behaviour. By applying these frameworks, this study aims to evaluate the performance, security, cost-effectiveness, and user satisfaction of OSS and CSS, thereby contributing to a deeper understanding of their respective strengths and weaknesses. The next chapters will build on this theoretical foundation to empirically investigate the effectiveness of OSS and CSS in various contexts.

CHAPTER 3: METHODOLOGY

3.1 Chapter Introduction

This chapter outlines the research methodology employed to evaluate the effectiveness of open-source software (OSS) versus closed-source software (CSS) within the Ministry of Information Communication Technology, Postal and Courier Services. The chapter details the research design, population sampling, data collection instruments, data collection procedures, data analysis, and ethical considerations. This structured approach aims to ensure the reliability and validity of the research findings.

3.2 Research Design

The research design for this study is a comparative case study approach, suitable for examining the differences in effectiveness between OSS and CSS in a specific organizational context. This design allows for a detailed and contextualized understanding of the factors influencing software effectiveness. The study employs both qualitative and quantitative methods to provide a comprehensive analysis.

3.3 Population Sampling

The population for this study consists of employees from the Ministry of Information Communication Technology, Postal and Courier Services. The convenience sampling procedure is used initially to select the departments which are going to be further sampled in order to find a well representable group of research participants. The departments chosen are; Cyber-Security, Application Development Management and Infrastructure Development Management. Below is a table with the number of employees that are in the three departments stated above;

Figure 1. IT Employees in the MICTPCS organisation

IT DEPARTMENT	NUMBER OF EMPLOYEES
Cyber-security	16
Infrastructure Development Management	14
Application Development Management	17

After determining the departments of interest, the population in those departments is going to be sampled using systematic sampling method because it is simple, fast and has little bias. The employee names will be used for such procedure and the final research participants will be presented in a table as follows;

Figure 2. Selected Research Participants

NAME	DEPARTMENT	EMAIL	CONTACT NO.

3.4 Data Collection Instruments

Data will be collected using two primary instruments: questionnaires and interviews. These instruments are chosen for their effectiveness in gathering both quantitative and qualitative data.

3.4.1 Questionnaires

Questionnaires will be used to collect quantitative data on various aspects of software effectiveness. The questionnaires will include closed-ended questions and Likert scale items, covering the following areas:

- **Performance**

  It is measured by response speed and reliability of the system or software.

- **Security**

  Perceptions of vulnerability, data protection, and response to security threats.

- **Cost-Effectiveness**

  Initial costs, maintenance costs, and total cost of ownership.

- **User Satisfaction**

  Overall satisfaction, ease of use, and support received.

The questionnaires will be distributed electronically to ensure ease of access and a higher response rate.

3.4.2 Interviews

Interviews will be used to collect qualitative data, providing deeper insights into employees' experiences and perceptions of OSS and CSS. Semi-structured interviews will be conducted to allow flexibility in exploring specific issues while maintaining a consistent framework. The interview questions will cover:

- Specific instances of using OSS and CSS in their work.
- Perceived benefits and drawbacks of each software type.
- Impact of software on their productivity and job satisfaction.
- Suggestions for improvements or changes.

3.5 Data Collection Procedure

The data collection procedure will involve several key steps to ensure accuracy and completeness:

1. **Preparation**:

   Develop and pilot test the questionnaires and interview guides to ensure clarity and reliability. A small subset of the population (e.g., 2-3 employees) will participate in the pilot test.

2. **Distribution**:

   Distribute the questionnaires electronically to all 13 selected participants. Follow-up emails will be sent to encourage participation and ensure a high response rate.

3. **Interviews**:

   Schedule and conduct interviews with the 13 participants. Interviews will be recorded (with consent) and transcribed for analysis.

4. **Collection**:

   Collect completed questionnaires electronically and ensure that all responses are recorded accurately and confidentially.

3.6 Analysis and Organization of Data

Data analysis will be conducted in two phases: quantitative analysis for the questionnaire data and qualitative analysis for the interview data.

3.6.1 Quantitative Analysis

Quantitative data from the questionnaires will be analysed using descriptive and inferential statistics:

- **Descriptive Statistics**:

  Measures such as means, medians, and standard deviations will summarize the data.

- **Inferential Statistics**:

  Techniques such as t-tests or chi-square tests will compare the effectiveness of OSS and CSS and identify any significant differences. Statistical software like SPSS or Excel will be used for data analysis.

3.6.2 Qualitative Analysis

Qualitative data from the interviews will be analysed using thematic analysis:

1. **Familiarization**:

Reading and re-reading the interview transcripts to become familiar with the data.

2. **Coding**:

   Identifying and coding significant statements or phrases related to the research questions.

3. **Theme Development**:

   Grouping the codes into themes that capture key aspects of software effectiveness.

4. **Reviewing Themes**:

   Refining and reviewing the themes to ensure they accurately represent the data.

5. **Defining and Naming Themes**:

   Clearly defining each theme and explaining its relevance to the research questions.

Qualitative data analysis software such as NVivo may be used to assist in organizing and analysing the data.

3.7 Ethical Considerations

Ethical considerations are critical in conducting research involving human participants. The following measures will be taken to ensure ethical compliance:

1. **Informed Consent**

   Participants will be provided with an informed consent form explaining the study's purpose, the voluntary nature of participation, and measures to ensure confidentiality and anonymity. Consent will be obtained before participation.

2. **Confidentiality**

   Responses will be kept confidential and used solely for the purposes of this research. Data will be securely stored and only accessible to the researcher.

3. **Anonymity**

   Participants' identities will be anonymized in all reports and publications resulting from this research. Personal identifiers will be removed from the data before analysis.

4. **Non-Coercion**:

   Participation in the study will be entirely voluntary, and participants can withdraw at any time without any consequences.

5. **Data Protection**:

   Data will be stored in encrypted files, and any physical copies of data will be kept in a locked cabinet. Data will be retained only for the duration necessary for the research and then securely destroyed.

3.8 Chapter Summary

This chapter has outlined the methodology for evaluating the effectiveness of OSS versus CSS within the Ministry of Information Communication Technology, Postal and Courier Services. The research design, population sampling, data collection instruments, data collection procedures, data analysis methods, and ethical considerations have been described comprehensively. These methodological steps ensure that the research is conducted systematically and ethically, providing reliable and valid findings. The next chapter will present the results and analysis based on the collected data.

CHAPTER 4: DATA PRESENTATION, ANALYSIS AND INTERPRETATION

4.1 Introduction

This chapter presents, analyses, and interprets the data collected from the Ministry of Information Communication Technology, Postal and Courier Services (MICTPCS) to evaluate the effectiveness of open-source software (OSS) versus closed-source software (CSS). The data, obtained through questionnaires and semi-structured interviews with 13 selected participants from the Cyber-Security, Infrastructure Development Management, and Application Development Management departments, is presented in a structured manner. This chapter aims to provide a comprehensive understanding of the participants' perceptions and experiences regarding software performance, security, cost-effectiveness, and user satisfaction, ultimately contributing to the research objective of comparing OSS and CSS effectiveness within the Zimbabwean context.

4.2 Data Presentation and Analysis

4.2.1 Demographic Context and Participant Representation

As detailed in Chapter 3, the systematic sampling method was used to select 13 participants from the three identified departments. The participant list, as provided in Table 4.1, includes their names, departments, email addresses, and contact numbers. Note that IDM, ADM and CS stand for Infrastructure Development Management, Application Development Management and Cyber-Security.

Table 4.1: Selected Research Participants

Name	Department	Email	Contact No.
Alice Chirara	CS	chiraraalice@gmail.com	0771092880

Ben Makava	IDM	kingben67@yahoo.com	0717439995
Clara Marizani	ADM	clarmar@gmail.com	0772567806
Vernon Mushambi	CS	vxshambi@gmail.com	0771457496
Emily Dube	IDM	madhuvee01@yahoo.com	0777654110
Panashe Bobojani	ADM	panashebob@gmail.com	0717667734
Nyasha Nyamvura	CS	nyashanyamvura@gmail.com	0776540334
Simbarashe Fani	IDM	simbafani@gmail.com	0774178641
Munashe Kavhuya	ADM	kavmun@yahoo.com	0774345660
Elias Machaka	CS	emachaka@gmail.com	0717561229
Kelly Musara	IDM	musarak@yahoo.com	0774778797
Ropafadzo Taviringwa	ADM	taviringwarop@gmail.com	0776311452
Van D Dzobo	CS	dzobovand@gmail.com	0771657682

4.2.2 Quantitative Data: Questionnaire Analysis

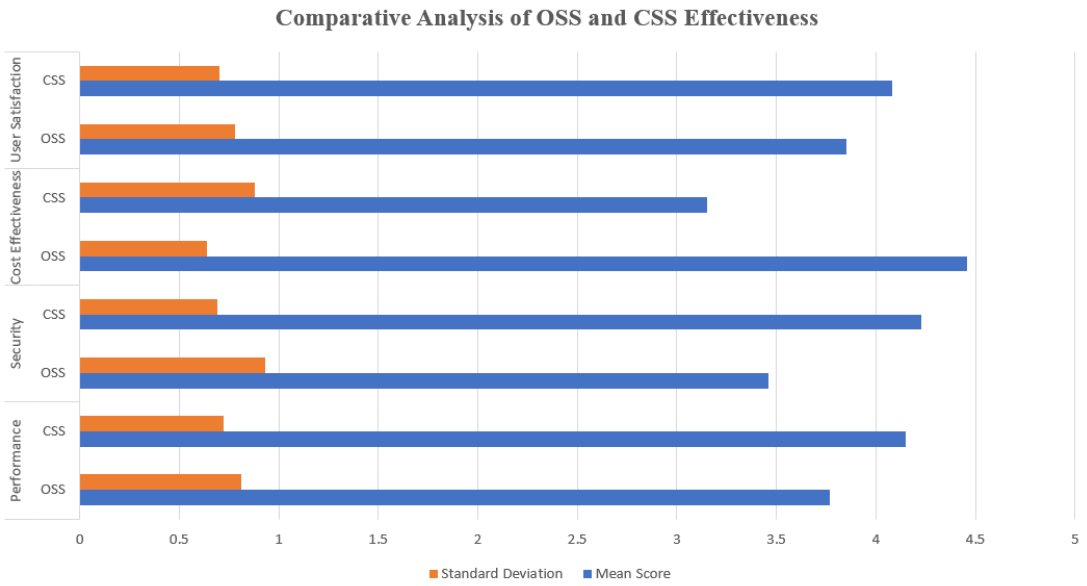
The questionnaire data, collected through Likert scale items (1 = Strongly Disagree, 5 = Strongly Agree), provided insights into the participants' perceptions of OSS and CSS across four key dimensions: performance, security, cost-effectiveness, and user satisfaction.

Table 4.2: Mean Scores and Standard Deviations for OSS and CSS Effectiveness

Dimension	Software Type	Mean Score	Standard Deviation
Performance	OSS	3.77	0.81

	CSS	4.15	0.72
Security	OSS	3.46	0.93
	CSS	4.23	0.69
Cost Effectiveness	OSS	4.46	0.64
	CSS	3.15	0.88
User Satisfaction	OSS	3.85	0.78
	CSS	4.08	0.70

Figure 4.1: Comparative Analysis of OSS and CSS Effectiveness



4.2.3 Qualitative Data: Interview Analysis

Thematic analysis of interview transcripts, rigorously coded and evaluated using NVivo software, revealed several recurring themes that provide a rich, detailed knowledge of the participants' experiences and perspectives of OSS and CSS. These themes shed light on the practical consequences of software selection within the MICTPCS, particularly in light of Zimbabwe's specific problems.

- ✓ **Performance and Stability:**

CSS Reliability for Critical Applications: Participants regularly emphasised CSS's perceived stability and reliability, especially in key infrastructure applications such as network management and core database systems. They identified instances where CSS-based systems performed admirably under high load, reducing downtime and assuring continuous service delivery. For example, one network administrator stated, "With our proprietary network monitoring system [CSS], we've had minimal disruptions." It is critical for us to maintain that degree of dependability. This preference for CSS in critical applications arises from the belief that suppliers provide rigorous testing and quality assurance to ensure system reliability.

OSS Flexibility vs. Performance Fluctuations: OSS was praised for its adaptability and versatility, particularly in development contexts where rapid prototyping and customisation were critical. Participants did, however, note occasional performance fluctuations, which were generally attributed to compatibility concerns with legacy systems or a lack of specialised assistance. For example, an application developer stated that "Linux [OSS] gives us a lot of freedom, but we sometimes encounter driver compatibility issues with older hardware, which can impact performance." These oscillations demonstrate the trade-off between OSS flexibility and CSS's perceived steadiness.

The Impact of Resource Constraints: Zimbabwe's economic status has an impact on the availability of current gear. Older hardware that is still in use frequently has compatibility concerns with modern software, particularly open-source software.

- ✓ **Security Perceptions:**

CSS and Vendor-Driven Security: CSS was frequently associated with enhanced security features, owing to vendor support and specific security upgrades. Participants indicated confidence in the vendor's ability to address security issues quickly, frequently mentioning proactive security patches and fast updates. For example, one cybersecurity expert remarked, "With our Windows Server environment [CSS], we receive regular security updates from Microsoft, which gives us peace of mind." This impression of vendor-driven security is critical in an era where cybersecurity threats are becoming more sophisticated.

OSS Transparency vs. Potential Vulnerabilities: OSS was treated with caution due to potential vulnerabilities in open-source code. Participants voiced fear that bad actors could gain access to source code, potentially exposing security issues. However, the transparency of OSS was recognised as a security benefit, enabling for community-driven security audits and quick patching. A security engineer, for example, wrote, "The open-source nature of Linux [OSS] means that vulnerabilities can be discovered and exploited, but it also means that the community can quickly develop patches." This demonstrates the complex link between OSS transparency and security perceptions.

The Issue of Security Updates: The timeliness of security upgrades for OSS and CSS was a major problem. Delays in patching vulnerabilities were viewed as a significant danger, particularly in critical infrastructure applications.

- ✓ **Cost and Resource Implications:**

OSS Cost-Effectiveness in Zimbabwe's Economic Context: OSS was often praised for its cost-effectiveness, especially in light of Zimbabwe's fiscal constraints. Participants emphasised the huge cost savings from licensing costs as well as the ability to use community-developed materials. For instance, one IT manager commented, "Using LibreOffice [OSS] instead of Microsoft Office [CSS] has saved us a considerable amount of money." This cost benefit is critical in a resource-constrained setting, where financial constraints can dramatically influence software adoption decisions.

CSS Licensing Costs and Vendor Support: CSS was seen as more expensive due to license fees, although participants recognised the importance of vendor support and detailed documentation. They highlighted examples of how vendor support was vital in addressing critical technical difficulties, reducing downtime, and guaranteeing business continuity. For example, one system administrator stated, "The vendor support we receive with our proprietary database system [CSS] is invaluable when we encounter complex technical problems."

The "Total Cost of Ownership" Debate: Participants talked on the "total cost of ownership" of both OSS and CSS, which included factors other than initial licensing

fees. They noted that, while OSS may have cheaper initial costs, it may necessitate additional internal resources for implementation, maintenance, and support.

- ✓ **User Experience and Support:**

CSS User-Friendliness and Documentation: CSS was widely seen as more user-friendly, with easily available documentation, intuitive interfaces, and devoted vendor support. Participants described how vendor-provided training and documentation aided smooth program adoption and reduced user errors. For example, a technical support professional said, "The user interface of Microsoft Office [CSS] is very intuitive, and the documentation is comprehensive, which makes it easy for our users."

OSS Community Support and Self-Learning: OSS users frequently relied on online communities, forums, and self-learning, which presented issues for some participants, particularly those with insufficient technical knowledge. Participants identified the lack of established support channels as a significant disadvantage, particularly when resolving difficult technical challenges. A data analyst, for example, stated that "Finding solutions to problems with LibreOffice [OSS] can be challenging, as we often have to rely on online forums and self-learning."

The Digital Divide: Several attendees noted the digital divide within the ministry. Some employees have a high level of computer literacy, while others have very low levels. This makes using OSS challenging, as it demands a higher level of computer literacy.

- ✓ **Customization and Adaptation:**

OSS Flexibility and Tailored Solutions: OSS was widely lauded for its flexibility and customisation capabilities, which allowed the ministry to build software solutions to its specific requirements. Participants described situations where OSS was customised to meet specific needs, improving operational efficiency and effectiveness. For example, one application developer remarked, "We were able to customise the Linux [OSS] environment to meet our specific security requirements."

CSS Rigidity and Limited Customization: CSS was viewed as stiff and less adaptive to the Ministry's specific requirements. Participants expressed dissatisfaction with proprietary software's limits, notably in customising functionality to match specific operating requirements. For example, an employee in application development management remarked that "the closed source nature of the software, makes it very hard to develop in a way that fully meets the needs of the ministry."

The Importance of Local Expertise: Participants emphasised the relevance of local knowledge while customising and supporting OSS and CSS. They emphasised the importance of qualified IT specialists who could tailor software solutions to the unique setting of the MICTPCS.

4.3 Discussion and Interpretation

The quantitative data, as shown in Table 4.2 and Figure 4.1, reveals a discernible trend; participants generally perceive closed-source software (CSS) as superior in performance, security, and user satisfaction, while open-source software (OSS) is significantly favoured for its cost-effectiveness. This trend is further illuminated by the qualitative findings from the interview analysis, which provide a richer, more nuanced understanding of the underlying factors that influence.

Performance and Stability:

The quantitative data, which shows CSS scoring higher in the performance category, supports the qualitative input. Participants frequently emphasised CSS's perceived stability and reliability, particularly in critical infrastructure applications. This notion is most likely based on the belief that vendors do extensive testing and quality assurance to minimise downtime.

In contrast, the reported performance fluctuations in OSS, as indicated by the slightly higher standard deviation in the quantitative data and the qualitative accounts of compatibility issues, suggest that a lack of dedicated vendor support and potential compatibility challenges with legacy systems can have an effect on performance. This is especially important in a ministry where operational stability is crucial. The lack of standardised support can make it harder to resolve difficulties.

Zimbabwe's economic position has forced the ministry to rely on obsolete gear. This outdated hardware frequently struggles to run modern open-source applications.

Security:

The quantitative evidence supports the perception of CSS's higher security, which is supported by vendor support and specific security upgrades. Participants were confident in the vendor's capacity to address security problems quickly.

However, the qualitative data also suggests that the transparency of OSS, although possibly revealing weaknesses, enables for community-driven security solutions. This perspective offers a trade-off between vendor-driven security and community-driven security, with each having its own set of advantages and downsides.

The speed with which security upgrades are distributed can mean the difference between a successful and a preventable cyber-attack.

Cost-Effectiveness:

The quantitative results indisputably illustrate the cost-effectiveness of open-source software, a conclusion that is strongly reinforced by qualitative accounts of large licensing fee reductions.

In Zimbabwe, where budgetary restrictions are a major issue, this cost advantage is critical. The opportunity to use community-created resources while avoiding expensive license fees is an important consideration in software adoption decisions. The economic situation in Zimbabwe makes cost one of the most crucial considerations.

User Experience:

The quantitative data indicates a modest preference for CSS in terms of user happiness, which is congruent with qualitative input on vendor support accessibility and extensive documentation.
However, the qualitative data reveals the difficulties associated with self-learning and

community-driven support for OSS, particularly for users with low technical knowledge. The digital divide within the ministry exacerbates this problem.

Customization:

The qualitative data demonstrates that OSS's level of customisation is a significant advantage. The ability to customise the program to meet the specific demands of the ministry is tremendously beneficial.
CSSs closed-source nature inhibits the ministry's ability to customise the software to meet its specific requirements.

The Trade-Off and Decision-Making:

According to both quantitative and qualitative data sources, the MICTPCS faces a substantial trade-off between the cost-effectiveness and customisation of OSS and the perceived stability, security, and usability of CSS.
In a resource-constrained environment like Zimbabwe, the cost benefits of open-source software are appealing. However, the ministry must also examine the potential performance swings, security risks, and user support issues that come with OSS.
CSS, on the other hand, provides apparent benefits in terms of stability, security, and user experience, but the licensing prices can be prohibitively expensive.
The flexibility to customise OSS allows the software to be fitted to the ministry's specific requirements, which is a significant advantage.

The decision-making process must include a thorough assessment of the ministry's specific requirements, resources, and risk tolerance. For example, if operational stability and security are critical and the ministry has the budget for it, CSS may be the best option. However, if cost-effectiveness and customisation are the most important aspects, and the ministry is ready to spend in training and support, OSS may be a viable solution.
Finally, the best software adoption strategy will most likely take a hybrid approach, using the strengths of both OSS and CSS to fulfil the ministry's different needs. This

hybrid approach should be underpinned by a strong policy framework that addresses the unique difficulties and opportunities associated with each software type.

Summary

This chapter presented and analysed data from the MICTPCS, emphasising the perceived strengths and weaknesses of OSS and CSS. The quantitative and qualitative findings provide useful insights into the elements that influence software effectiveness in Zimbabwe. The next chapter will draw conclusions and make recommendations based on these findings.

CHAPTER 5: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter serves as the culmination of the research endeavour, providing a comprehensive synthesis of the findings derived from the comparative analysis of open-source software (OSS) and closed-source software (CSS) within the Ministry of Information Communication Technology, Postal and Courier Services (MICTPCS) in Zimbabwe. It encapsulates the key discoveries, draws evidence-based conclusions, and formulates actionable recommendations tailored to the ministry's specific context. Furthermore, this chapter identifies avenues for future research to expand upon the current study and address emerging questions within the domain of software effectiveness in the public sector.

5.2 Discussion

The research findings, which were meticulously derived from the convergence of quantitative and qualitative data, reveal a complex tapestry of factors that have a significant impact on the perceived effectiveness of open-source software (OSS) and closed-source software (CSS) within the operational context of the Ministry of Information Communication Technology, Postal, and Courier Services. The quantitative analysis, which used Likert scale surveys, revealed a clear trend: participants usually regarded CSS as having greater performance, security, and user pleasure, whereas OSS was unquestionably preferred for its significant cost-effectiveness. This quantitative trend was not only a statistical observation; it was strongly supported and extended by qualitative insights gained from in-depth, semi-structured interviews. These interviews gave a more nuanced, contextualised understanding of the participants' lived experiences and subjective perspectives, highlighting the practical ramifications of software decisions inside the ministry.

Perceived Stability and Reliability of CSS in Critical Infrastructure:

The interview results frequently emphasised CSS's perceived stability and reliability, particularly in essential infrastructure applications such as network management systems, database servers, and core communication platforms. Participants regularly

mentioned situations where CSS-based systems performed well under high load, reducing downtime and assuring continuous service delivery. This image stems from the assumption that vendors, through rigorous testing, quality assurance, and devoted support, provide a level of operational stability critical to the ministry's essential duties.

Network administrators emphasised the dependability of proprietary network monitoring tools, adding, "With our vendor-supported system, we experience minimal disruptions, which is crucial for maintaining network uptime." This emphasis on stability demonstrates the ministry's reliance on CSS for mission-critical operations, where even minor disruptions can have serious effects.

Challenges of OSS Support and the Digital Divide:

The qualitative data also highlighted the challenges of OSS's self-learning and community-driven assistance strategy. Participants, particularly those with low technical knowledge, voiced dissatisfaction with the absence of institutional support channels and the reliance on internet forums and documentation. This problem is exacerbated by the ministry's digital gap, which creates discrepancies in the ability to efficiently use open-source software due to differing degrees of computer proficiency and resource access.

Many participants stated that the absence of government help made resolving concerns time-consuming. The digital divide, is especially problematic, because some employees are unable to use OSS, effectively.

OSS Flexibility, Customization, and Economic Advantages:

OSS, on the other hand, was widely praised for its inherent flexibility and customisation capabilities, which allow the ministry to build software solutions to its specific operating requirements. Participants described situations where OSS was customised to meet specific needs, improving operational efficiency and effectiveness. The economic benefits of OSS, particularly in light of Zimbabwe's budgetary constraints, were also important factors. The absence of license fees and the availability of community-created materials were viewed as critical considerations in software adoption decisions.

The capacity to modify and adapt software was viewed as a significant benefit, allowing the ministry to be extremely agile.

Security Trade-Offs: Vendor-Driven vs. Community-Driven:

The security conversation highlighted a complex trade-off between vendor-driven security (CSS) and community-driven security (OSS). Participants recognised CSS suppliers' robust security features and timely upgrades, as well as the OSS community's transparency and potential for speedy patching. This trade-off emphasises the importance of a nuanced approach to security, taking into account the unique risks and vulnerabilities associated with each program type. The speed with which security patches are deployed was a big worry.

Impact of Zimbabwe's Economic Situation and Older Hardware:

The economic condition in Zimbabwe, which is marked by limited resources and ageing infrastructure, has a significant impact on software selection. The ministry's reliance on older hardware, which may not be fully compatible with current software versions, adds to the decision-making process. This environment highlights the importance of software solutions that are both cost-effective and flexible to the ministry's existing infrastructure.

The current economic scenario has resulted in a paucity of funding for cutting-edge gear.

The Critical Decision-Making Process:

The confluence of these data highlights the MICTPCS's essential decision-making process. The ministry must strike a careful balance between the cost-effectiveness and customisation of OSS and the perceived stability, security, and user support of CSS. This balancing act necessitates a detailed awareness of the ministry's specific needs, resources, and risk tolerance, as well as a systematic approach to software adoption that harnesses the capabilities of both OSS and CSS.

5.3 Conclusions

Based on the rigorous empirical evidence acquired through questionnaires and interviews, and the preceding detailed discussion, the following findings are formed, each of which sheds light on the complex dynamics of OSS and CSS effectiveness inside the MICTPCS.

Cost-Effectiveness and Customization of OSS vs. Stability, Security, and Support of CSS:

OSS provides a compelling value proposition by giving significant cost advantages, which is especially important in Zimbabwe's economically limited context. The lack of licensing fees and the availability of community-based services result in significant savings. Furthermore, OSS's inherent flexibility enables substantial customisation, allowing the ministry to build software solutions to its specific operational requirements.

However, cost-effectiveness and customisation have trade-offs. CSS is widely regarded as more stable and trustworthy, especially in mission-critical applications where downtime is unacceptable. CSS provides a sense of security and user delight that OSS environments frequently lack.

The digital divide, and the economic situation of the country, make this especially important.

Contextual Trade-Offs and the Importance of Tailored Solutions:

The usefulness of OSS and CSS varies depending on the circumstance. The best software choice varies greatly depending on the ministry's specific requirements, available resources, and risk tolerance. For example, departments handling sensitive data or vital infrastructure may prioritise CSS stability and security, whilst those focused on development or cost optimisation may prefer OSS.

This result emphasises the necessity for a nuanced, case-by-case approach to software adoption. One-size-fits-all strategies are unlikely to produce optimal results. Prior to making software decisions, the ministry must do extensive assessments of departmental demands and operational requirements.

The Potential of a Hybrid Approach:

A hybrid strategy, which strategically combines the strengths of both OSS and CSS, offers a promising answer for the MICTPCS. This approach enables the ministry to

benefit from the cost-effectiveness and customisation of OSS for specific applications while depending on CSS's stability, security, and support for mission-critical systems.

Implementing a hybrid strategy necessitates careful planning and coordination to enable seamless integration and interoperability among various software platforms. It also involves the creation of clear standards and procedures for software selection and implementation.

The Crucial Role of Training and Support:

Adequate training and assistance are critical for the successful adoption and use of both OSS and CSS. The self-learning and community-driven support approach of OSS can be difficult for users with low technical expertise, and even CSS requires continual training to maximise its potential.

The ministry must engage in extensive training programs to provide its IT personnel with the required skills and expertise to efficiently use both types of software. This involves granting access to paperwork, internet resources, and expert assistance. Local knowledge in customising and supporting OSS and CSS is essential for tailoring software solutions to the MICTPCS setting. Zimbabwe's distinct operational environment, marked by resource restrictions and ageing infrastructure, needs customised software solutions that meet local concerns.

The government should prioritise the development of local IT talent while cultivating a culture of information sharing and collaboration. This can be accomplished through training programs, mentorship initiatives, and collaborations with local colleges and technical schools. By increasing domestic capability, the ministry may reduce its reliance on external vendors while ensuring the long-term viability of its software solutions.

5.4 Implications

The findings of this study, grounded in both quantitative and qualitative data, carry significant implications that extend beyond the immediate context of the MICTPCS,

impacting the broader Zimbabwean public sector's approach to software adoption and utilization.

Policy Formulation: Guiding Software Adoption Decisions:

Explanation: The study emphasises the significant trade-offs involved in choosing between OSS and CSS. As a result, the MICTPCS should create a thorough policy framework that informs software adoption decisions. This framework should explicitly address the cost-effectiveness, performance, security, and user experience trade-offs, as well as provide clear criteria for evaluating and choosing software solutions.

Implications: This policy framework should be clear and adaptive, enabling for responsiveness to changing technological landscapes and organisational needs. It should also include best practices for software procurement, implementation, and maintenance, ensuring that software investments are consistent with the ministry's strategic goals. This strategy will also establish a clear framework for other government departments to follow.

Resource Allocation: Investing in Training, Support, and Maintenance:

Explanation: The study emphasises the significance of proper resources for training, support, and maintenance. Implementing and effectively utilising both OSS and CSS necessitates significant investments in human capital and infrastructure. The ministry must set aside adequate cash to provide ongoing training for IT specialists, build dedicated support teams, and keep software systems up to date.

Implications: Failure to invest in these areas might result in inefficient software utilisation, security risks, and higher operating expenses. Prioritising resource allocation for training and support allows the ministry to maximise the return on its software investments while also ensuring the long-term viability of its IT infrastructure.

Capacity Building: Empowering IT Professionals:

Explanation: Investing in capacity building for IT professionals is critical to ensuring the efficient use of both OSS and CSS. This provides possibilities for ongoing education, professional development, and knowledge exchange. The ministry should cultivate an innovative and collaborative culture, allowing its IT

workers to experiment with new technologies and create custom software solutions.

Implications: By increasing local capacity, the ministry can minimise its dependency on third-party vendors, improve its technical competence, and encourage the creation of indigenous software solutions. This not only boosts the ministry's IT infrastructure, but also helps to build Zimbabwe's ICT sector.

Strategic Software Selection: Informing Future Decisions:

Explanation: The findings of this study provide useful insights that can be used to guide future software selection decisions. Understanding the strengths and disadvantages of both OSS and CSS allows the ministry to make informed decisions that meet its specific demands and operational constraints.

Implications: The findings of this study can be used to establish a strategic software selection process that takes into account a comprehensive review of technical, economic, and organisational variables. By using a data-driven approach, the ministry can ensure that its software investments improve overall efficiency and performance.

Economic Impact: Reinvesting Cost Savings:

Explanation: The cost savings associated with OSS, particularly given Zimbabwe's budgetary constraints, may allow the ministry to invest in other key areas. By lowering software licensing and maintenance costs, the ministry may free up funds for infrastructure development, capacity building, and service delivery.

Implications: This economic impact extends beyond the ministry's immediate benefits and contributes to Zimbabwe's overall economic development. The public sector may drive innovation, generate jobs, and boost the national economy's competitiveness by encouraging the use of low-cost software solutions.

5.5 Recommendations

Based on the empirical evidence, conclusions, and implications discussed in this study, the Ministry of Information Communication Technology, Postal, and Courier Services (MICTPCS) is given the following specific and actionable recommendations to optimise its software adoption and utilisation strategies:

Develop a Hybrid Software Strategy: Tailored to Departmental Needs:

The MICTPCS should plan and execute a comprehensive hybrid software strategy. This approach should include a thorough assessment of each department's unique demands, operating requirements, and risk tolerance. CSS should be prioritised in critical infrastructure and security-sensitive applications because to its perceived stability and vendor support. OSS should be considered for development environments, cost-sensitive projects, and applications that require extensive customisation.

Actionable Steps:

- Conduct departmental audits to identify software needs and usage patterns.
- Develop a matrix that maps software types (OSS/CSS) to specific departmental applications.
- Establish clear guidelines for software selection, procurement, and deployment.
- Implement a phased approach to hybrid software adoption, starting with pilot projects.
- Ensure that all systems are compatible with each other.

Invest in Comprehensive Training and Support: Bridging the Digital Divide:

The MICTPCS must make considerable investments in comprehensive training and support programs for OSS and CSS. These systems should handle the issues that come with self-learning and community-driven support, especially for users with varied levels of technical ability.

Actionable Steps:

- Develop tailored training modules for different user groups (e.g., IT professionals, end-users).
- Provide access to online resources, documentation, and expert trainers.

- Establish a help desk or support hotline for timely assistance.
- Conduct regular workshops and seminars to promote knowledge sharing.
- Create training documents that are easy to understand.

Establish a Dedicated Support Team: Expertise in OSS and CSS:

The MICTPCS should set up a dedicated support team with experience in both OSS and CSS. This team will operate as a single point of contact for technical support, troubleshooting, and software maintenance.

Actionable Steps:

- Recruit or train IT professionals with expertise in both OSS and CSS.
- Develop a service level agreement (SLA) for support services.
- Implement a ticketing system to track and resolve support requests.
- Conduct regular performance reviews of the support team.
- Ensure that the support team, is well funded.

Prioritize Security: Robust Measures and Timely Updates:

The MICTPCS must prioritise security in both OSS and CSS. This includes putting in strong security measures, guaranteeing timely security updates, and conducting frequent security audits.

Actionable Steps:

- Develop a security policy that addresses both OSS and CSS vulnerabilities.
- Implement intrusion detection and prevention systems.

- Establish a patch management process for timely security updates.
- Conduct regular vulnerability assessments and penetration testing.
- Train employees on cyber security best practices.

Promote Local Expertise: Reducing Reliance on External Vendors:

The MICTPCS should encourage the growth of local competence in customising and supporting OSS and CSS. This eliminates dependency on external providers and encourages the expansion of Zimbabwe's ICT economy.

Actionable Steps:

- Partner with local universities and technical institutions to develop training programs.
- Provide mentorship opportunities for junior IT professionals.
- Encourage participation in open-source communities.
- Establish a knowledge-sharing platform for IT professionals.
- Provide funding for local ICT research.

Create a Policy Document: Guiding OSS Adoption:

The MICTPCS should develop a thorough policy document outlining the ministry's position on open-source software. This document should include explicit principles for evaluating, selecting, implementing, and maintaining open-source software.

Actionable Steps:

- Establish a committee to develop the policy document.
- Consult with relevant stakeholders, including IT professionals, department heads, and legal experts.
- Include guidelines on licensing, security, support, and customization.

- Review and update the policy document regularly.
- Make the policy document available to all government departments.

5.6 Suggestions for Further Research

To build on the findings of this study and answer increasing doubts about the effectiveness of OSS and CSS within the Zimbabwean public sector, notably the MICTPCS, the following proposals for further research are made:

Longitudinal Study: Assessing Long-Term Impact and Sustainability:

Suggestion: Conduct a longitudinal study over several years to evaluate the long-term impact of OSS and CSS adoption on the ministry's operational efficiency, cost-effectiveness, and sustainability. This study should look at trends in software usage, maintenance costs, performance measures, and user satisfaction over time.

Rationale: A longitudinal approach will provide light on the long-term viability of various software solutions, highlighting the true costs and advantages of OSS and CSS. It will also assist in identifying any new hurdles or opportunities for software adoption in Zimbabwe's changing technology ecosystem.

Methodological Considerations: This study should take a mixed-methods approach, combining quantitative data (e.g., performance measurements, cost statistics) and qualitative data (e.g., user feedback, expert interviews).

Comparative Analysis Across the Public Sector: Identifying Best Practices and Common Challenges:

Suggestion: Conduct a comparative analysis of OSS and CSS adoption among Zimbabwe's public sector organisations. This research should look at differences in software utilisation patterns, adoption tactics, and perceived efficacy in order to discover best practices and common difficulties.

Rationale: A comparative analysis will provide a more comprehensive knowledge of the factors that influence software adoption in Zimbabwe's public sector, allowing for the establishment of evidence-based policies and guidelines. It will also assist in

identifying possible synergies and areas for collaboration among government agencies.

Methodological Considerations: This study should take a case study approach, looking at software uptake in a representative sample of public sector organisations.

Impact of Local OSS Development: Addressing Specific Needs and Fostering Innovation:

Suggestion: Look into the influence of local OSS development on the Zimbabwean ICT sector and its potential to meet the ministry's specific demands. This research should look into the possibility of establishing indigenous OSS solutions, the problems and opportunities that come with local development, and the potential for fostering innovation and economic success.

Rationale: Local OSS development can provide personalised solutions to the Zimbabwean public sector's specific difficulties and requirements. It can also help to create local skills, minimise dependency on external vendors, and boost the growth of the local ICT sector.

Methodological Considerations: This study should use both qualitative research (e.g., interviews with developers and policymakers) and quantitative analysis (e.g., economic impact assessments).

Cybersecurity Implications: Exploring Effective Security Measures and Strategies:

Suggestion: Conduct a more in-depth investigation of the cybersecurity implications of OSS and CSS adoption inside the MICTPCS, including the efficacy of various security measures and methods. This research should look into potential vulnerabilities, security best practices, and the role of vendor support and community-led security activities.

Rationale: Cybersecurity is a critical concern for the public sector, particularly in light of rising cyber threats. This study will provide useful insights into the security risks and mitigation measures related with OSS and CSS adoption, allowing the ministry to improve its cybersecurity posture.

Methodological considerations: This study should include security audits, vulnerability assessments, and expert interviews.

Economic Impact Assessment: Calculating Cost Savings and Benefits
Suggestion:

Suggestion: Conduct a thorough economic effect study of OSS adoption in Zimbabwe's public sector, calculating possible cost savings and benefits. This study should look into how open-source software affects software license costs, maintenance charges, and overall operational efficiency.

Rationale: An economic impact evaluation will give evidence-based rationale for OSS adoption by demonstrating its ability to cut costs and improve resource allocation. It will also help policymakers make educated software investment decisions.

Methodological considerations: This study should include cost-benefit analysis, ROI calculations, and economic modelling.

Digital Divide Impact: Researching the impact that the digital divide has on the usage of open-source software:

Suggestion: Investigate how the digital divide affects the use of open-source software within the ministry. This study should look into the impact of computer access and literacy on the use of open-source software.

Rationale: The digital divide is a key issue in Zimbabwe, and understanding how it affects the use of open-source software will assist the ministry make educated judgements about software selection.

Methodological considerations: This study should use a combination of surveys and interviews.

REFERENCES

- Ajzen, I. (1991). The theory of planned behaviour. *Organizational Behaviour and Human Decision Processes*, 50(2), 179-211.
- Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management*, 17(1), 99-120.
- Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). New York: Free Press.
- Weber, S. (2004). The Success of Open Source. Harvard University Press.
- Cusumano, M. A. (2004). *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad*. Free Press.
- Free Software Foundation. (2023). *GNU General Public License*.
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open-source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(3), 309-346.
- Open-Source Initiative. (2023). *The Open-Source Definition*.
- Raymond, E. S. (1999). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media.
- Stallman, R. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press.
- Wheeler, D. A. (2007). *Why Open-Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*
- Vaughan-Nichols, S. J. (2023, June 14). *What is open-source software? Everything you need to know. ZDNET.* https://www.zdnet.com/article/what-is-open-source-software-everything-you-need-to-know/