SECURITY TESTING CHALLENGES OF WEB DEVELOPERS IN THE LAGOS, NIGERIA IT INDUSTRY

By

MOYINOLUWA IBUKUNOLUWA AJAYI

Student No. 216076371

Submitted in Fulfilment of the Academic Requirements for the Degree of

MASTER OF COMMERCE

in the Subject of

Information Systems and Technology

In the

School of Management, IT and Governance

College of Law and Management Studies

Supervisor: Mr. Deepak Kumar

Co-Supervisor: Professor Irene Govender

March 2020

DECLARATION

I, Moyinoluwa Ibukunoluwa Ajayi, declare that,

The research reported in this dissertation, except where otherwise indicated, is my original research.

This dissertation has not been submitted for any degree or examination at any other university.

This dissertation does not contain other persons' text, data, pictures, graphics or other information, unless specifically acknowledged as being sourced from relevant sources.

This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other sources. Where other written sources have been quoted, then:

> Their words have been re-written but the general information attributed to authors has been sourced.
> Where their exact words have been used, their writing has been placed inside quotation marks and clearly referenced.

Where I have reproduced a publication of which I am author, co-author or editor, I have indicated in detail which part of the publication was written by myself alone and have fully referenced such publications.

This dissertation does not contain data, text, graphics or tables copied and pasted from webpages, unless acknowledged, and the source being detailed in the dissertation and in the relevant reference section.

MOYINOLUWA IBUKUNOLUWA AJAYI

Student No. 216076371

March 2020

ACKNOWLEDGEMENTS

I am eternally grateful to my heavenly Father, God Almighty for giving me the wisdom, zeal and ability for this research. God is my rock.

My sincere thanks and appreciation go to the following people who encouraged me throughout this study and supported me.

- My beloved parents, Dr. and Prof (Mrs.) D. D. Ajayi for their love and support. Thank you for believing in my dreams. God bless you.

- My supervisors: Mr. Deepak Kumar and Prof. Irene Govender for their timely responses and guidance throughout my study.

- My siblings, Paul, Goodness and Mercy Ajayi. My dear grandparents, Chief and Mrs. J. A. Fadare, and to all relatives for their love and care. Special thanks to Pastor and Mrs. Abiodun Isola.

- My pastors, Daddy & Mummy Adejimi, for their mentorship and encouragements always.

- My UKZN colleagues Kenny, Eniola, Badru Abdulbaqi, and Sis. Toyin Ofusori, who were helpful at many times during this research. Also, my dear friend, Bukola Aborode.

ABSTRACT AND KEY TERMS

Web applications are instrumental for businesses. Due to the susceptible nature of the internet, which is their main operating environment, many vulnerabilities that compromise web applications are constantly reported. Despite these vulnerabilities, there is a huge pressure on web development teams to build applications to meet business demands. This leads to compromise in the quality and security testing process integrated into the development life cycle. Related studies have revealed that although there are many frameworks and tools to support Security testing, many of these developed frameworks and tools are often poorly adopted and are thus found ineffective. Studies have also revealed that in Nigeria, a huge amount of money is lost annually to software importation from foreign countries due to the low quality of indigenously-developed applications in the Information Technology industry.

This study investigates the practice of security testing among web development teams in the Information Technology industry in Lagos in Nigeria, and the factors that affect its actual usage. Three companies were randomly selected for the study, and both quantitative and qualitative research methods were used. A conceptual framework adapted from the technology acceptance model was used to guide the data collection instruments. The quantitative research method involved statistical analysis of eighty-two responses to the closed-ended Likert-type questionnaire. The qualitative research method involved using the data obtained from the interviews conducted with eight industry experts.

Findings from the study revealed three basic approaches to security testing used by web development teams in Lagos, Nigeria. Perceived usefulness, perceived ease of use and behavioural intention were significant constructs of the conceptual framework that predict the use of security testing among web developers in Lagos, Nigeria. Factors found to affect the effective usage of security testing techniques were human resources, project constraints, and ethical and compliance factors. To improve the usage of security testing, more awareness, training and technical support are required for development teams. Furthermore, ethical and compliance policies need to be provided by regulatory bodies in the industry to guide the security testing process for teams.

Project timelines should also be made flexible to give room for adequate security testing implementation in the Software development life cycle.

Key Terms: Information systems security; Information technology; Lagos; Nigeria; Perceived ease of use; Perceived usefulness; Secure software; Security testing; Software development life cycle; Software development teams; Technology acceptance model; Web applications; Web security.

TABLE OF CONTENTS

CHAPTER THREE: RESEARCH METHODOLOGY

CHAPTER FOUR: QUANTITATIVE ANALYSIS

CHAPTER SIX: DISCUSSION OF FINDINGS

CHAPTER SEVEN: CONCLUSIONS AND RECOMMENDATIONS

REFERENCES

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANOVA	Analysis of variance
API	Application program interface
ATT	Attitude
BIU	Behavioural intention to use
CF	Conceptual framework
CIA	Confidentiality, Integrity, Availability
CIAA	Confidentiality, Integrity, Availability, Authentication
CSRF	Cross site request forgery
ICT	Information and communications technology
IOTs	Internet of Things
IS	Information Systems
IT	Information technology
NIST	National institute of security
NITDA	Nigerian Information technology development agency
OSSTMM	Open source security testing methodology manual
OWASP	Open web application security project
PEOU	Perceived ease of use

PTES	Penetration testing standards
PU	Perceived usefulness
RAD	Rapid application development
RQ	Research question
STD	Standard deviation
SDLC	Software development life cycle
SDT	Software development teams
SQLi	Structured query language injection
ST	Security testing
TAM	Technology acceptance model
TPB	Theory of planned behaviour
TRA	Theory of reasoned action
UML	Unified modelling language
UTUAT	Unified theory of use and acceptance of technology
WAs	Web applications
WWW	World Wide Web
XSS	Cross site scripting
SME	Small to medium scale enterprise
STD	Standard deviation

CHAPTER ONE

INTRODUCING THE STUDY

1.1. Introduction

Web Applications (WAs) have become useful and instrumental for businesses in recent years. The rapid growth of the internet and its increasing availability across the world has created a constant demand for information by users (Jaiswal, Raj, & Singh, 2014). The world is more interconnected than ever, and there are huge speculations that WAs will drive businesses and provide a more integrated ecosystem in the years ahead. For companies and organisations, WAs are important interfaces through which they engage their customers regarding the products and services they offer. Organisations can achieve their goals faster because of the wide publicity and positive image that WAs create for their products. The accessibility and interoperability of WAs make it easy for companies to establish relationships and collaborations with other businesses across different domains, thus eliminating the problem of a fixed location. Realtime support and engagement is possible and hence companies can grow over time (Li, Das, & Dowe, 2014; Patil & Phil, 2014).

Web developers are a subgroup of a software development team (SDT) who build web applications. They possess the skills and technical competencies to build WAs across different domains such as ecommerce, education, healthcare, banking and payments, etc. The technologies and strategies needed for building WAs vary (Hope & Walther, 2008). The increasing vulnerability reports of the World Wide Web (WWW), which is the operating environment of WAs, makes their development critical. There is heightened pressure on businesses to adopt the use of WAs especially because of the ease and accessibility they provide to improve daily operations. The business demands from companies, coupled with the challenges of deploying safe and secure applications are major concerns for development teams (Doğan, Betin-Can, & Garousi, 2014; Gottipalla & Yalla, 2014).

The increased demand for WAs has impacted on the quality and security of development processes. In a bid to keep up with the demands and current trends in the

industry, management teams in different organisations set timelines and targets for software teams to meet the financial goals in constrained timeframes. As a result, software development teams (SDTs) switch between traditionally known software development life cycles (SDLCs) and more contemporary and progressive ones to meet the business demands in designated time (Chóliz, Vilas, & Moreira, 2015). SDTs are constantly faced with challenges that often lead to compromises being taken between known standards and industry best practices. Consequently, strategically implementing and applying security in the development life cycle is difficult (Cruzes, Felderer, Oyetoyan, Gander, & Pekaric, 2016; Sowunmi & Misra, 2015).

The nature of the internet and its interconnectedness has created a huge threat environment for WAs and this poses a risk for all users. Cybercrime and cyberattacks are widely reported by several notable organisations (Nieles, Dempsey, & Pillitteri, 2017). While the interactions between application programming interfaces (APIs), databases and the introduction of web services have been advantageous to businesses, it poses an escalated risk if not securely integrated in the SDLC (Mahendra & Khan, 2016).

Security should be a priority for organisations and their SDTs as WAs are critical software that are prone to malicious attacks. These attacks are often introduced through flaws in the SDLC and can be an impediment to organisational growth (Stuttard & Pinto, 2011). Companies have had major data losses due to breaches in their systems. These attacks are dangerous because the cybercriminals and hackers can launch attacks and exploit WAs without creating any apparent indication of a break-in (Khari, Sangwan, & Vaishali, 2016). The hackers can gain access to confidential information and use an already compromised system to launch further attacks on users as in the case of phishing (APWG, 2006).

Security testing (ST) is a type of software testing used to determine if an information system (IS) protects data and maintains functionality as intended. It helps to prevent vulnerabilities and make WAs less susceptible to compromise in the event of cyber-attacks. It is indispensable and critical in the development process and should be applied in all phases of the SDLC (Jaiswal et al. 2014). Appropriate ST approaches identify and prevent flaws, so that they can be fixed before deployment to the end users. Bugs and

flaws may potentially lead to security violations, which invalidate the integrity and confidentiality of applications. The extent of appropriate ST done through the SDLC is a major determinant of the security and quality of an application (Krishnaveni, Prabakaran, & Sivamohan, 2015).

Performing in-depth security assessments of all systems in an enterprise is a long and cost-intensive process (Scarfone, Souppaya, Cody, & Orebaugh, 2008). During this lengthy process, it is possible that development teams may not apply appropriate ST strategies that test and detect bugs in a manner that is commensurate with the degree of risk to the application. Additionally, some teams may not be able to manage the security requirements of a project appropriately due to various limiting factors.

1.2. Background to the study

Over the years, the practice of ST in software development has been carried out with different methods, frameworks and approaches (Finifter & Wagner, 2011). Many teams rely on existing security technologies to be integrated in the application coding process (Kaushik & Mohan, 2013), while other teams may not have a systematic approach because there is a lack of understanding about the science of the ST process. Still others, rely on last minute security scans and penetration tests (Türpe, 2008). In other instances, companies have segregated security teams which limit security responsibilities to certain people as opposed to implementing a collective responsibility of the entire SDT. (Chóliz *et al.*, 2015).

WAs have been known to present great risks and concerns and these are widely reported. In 2011, leading security research companies such as the Open Web Application Security Program (OWASP) and White-Hat Security Sentinel reported that about 64% of WAs contain some form of vulnerability that can be exploited by known hackers (Vala & Jasek, 2011). Symantec also reported in 2016 that about 78% of websites interfacing for WAs had some form of vulnerability, of which 15% were critical vulnerabilities to the applications (Symantec, 2016). A major pattern noted by the research is that most application vulnerabilities are introduced right at the beginning of the development life cycle. This may be attributed to a widely held view that most

web developers may not have a fully-fledged understanding of security vulnerabilities and how to prevent them (Stuttard & Pinto, 2011).

In Nigeria, recent efforts and directions have been taken towards driving development and growth through technological innovation and software development in the information technology (IT) industry. For web-based businesses and organisations, several compliance policies have also been set by governing bodies especially with respect to cybersecurity in the IT industry. A study by Sowunmi and Misra (2015) revealed that an insufficient supply of software products to the Nigerian IT industry was due to the inability and inadequacy of software companies to produce quality and secure software that meets acceptable and international standards by users and investors alike. As a result, business organisations and government agencies resort to outsourcing software needs to foreign vendors. The study by Sowumi and Misra (2015) has revealed that Nigeria loses an average of one billion USD ($1 000 000 000) per annum due to the importation of software applications from foreign vendors. Issues relating to a lack of awareness, quality training, and management bureaucracy are some of the many challenges that affect SDTs across the Nigerian IT industry (Sowunmi & Misra, 2015). While future projections are high, especially with innovations in mobility and the Internet of Things (IOTs), it is important that this study be conducted so that more awareness is created and solutions proffered to resolve the present challenges.

1.3. Research problem

The rapid development of WAs to meet business demands has generated several limitations. Increased constraints such as low budgets, delivery deadlines and lack of resources have affected the quality of the processes in the SDLC. This further leads to a compromise in the security implementation (Avancini, 2012).

In Nigeria, the software industry has been affected by the earlier stated constraints in recent times and this is causing a negative impact on the growth of the nation's economy (Oladejo & Ogunbiyi 2014). In Lagos, the economic capital of Nigeria, a growing number of international investors have visited several companies in the IT industry and more recognition is being given to the local IT industry globally (Ogunsola, 2016). A recent study however reported that only about 10% of indigenously developed

applications are used locally and this has been attributed to the quality and security of the application development processes across web and software development teams in the country (Sowunmi & Misra, 2015).

This present research work intends to identify the ST challenges among web developers and SDTs in the Lagos IT Industry. The study will also propose ways of mitigating these ST challenges and improve the quality and security of applications developed within the Lagos IT industry and which impact positively on the Nigerian economy.

1.4. Research questions

This study seeks to find out the current state of ST practices in Nigeria and to identify factors that affect its adoption among web development teams. Accordingly, the main research question (RQ) that drives this study is as follows:

> *How can ST practices among web development teams in the IT Industry be improved in Lagos, Nigeria?*

The main RQ is answered by considering the following sub-questions:

> **RQ1:** How is ST conducted by web development teams in Lagos, Nigeria software industry?
> **RQ 2:** How does perceived usefulness (PU), perceived ease of use (PEOU), behavioural intention (BIU) and attitude (ATT) predict the actual usage of ST approaches among web development teams in Lagos?
> **RQ 3:** What factors affect the effective use of appropriate ST approaches among web developers in Lagos, Nigeria?
> **RQ 4:** How can the use and adoption of ST among web developers in the Lagos IT industry be improved?

1.5. Research objectives

The following research objectives will be achieved by the study:

i. To discover the ST approaches currently used by web development teams in Lagos, Nigeria.

ii. To understand how the PU, PEOU, BIU and ATT influence the use of ST approaches among web development teams in Lagos.

iii. To identify factors that affect the use of ST in development by web developers in Lagos, Nigeria.

iv. To propose better ways of improving the use and adoption of ST among web development teams in Lagos, Nigeria

1.6. The technology acceptance model

The framework used to achieve the stated research objectives is adapted from the second model of the technology acceptance model (TAM2). The TAM was developed in 1986 by Fred Davis (Davis, 1989) as an IS model that has been widely used to explain technology acceptance and adoption over the years. The TAM is based on individual views that affect the actual use of an IT/IS system. Several iterations of the TAM have evolved over the years and the TAM2 is the second iteration adapted in this study.

The TAM2 explains IT user behaviour by defining a basis for depicting how some external variables could influence the attitude and intention to use (Davis, 1989). The TAM stipulates that the actual use of a technology system/process by a user is directly or indirectly affected by its perceived usefulness, the user's attitude and behavioural intentions, with PEOU directly dependent on PU.

The TAM has been used in several studies to understand and predict the usage of IT systems and processes among professionals within an organisational setting (Erasmus, Rothmann, & Van Eeden, 2015; Johnson, 2005; Paquet, 2013). The relationships among the constructs of the framework are measured to predict usage and adoption of ST in the sample. A more detailed discussion of the TAM and the conceptual framework is given in chapter two of this study.

1.7. Overview of the research methodology

This study seeks to investigate the current state of ST practice among IT companies in Lagos, Nigeria and utilises an exploratory design. The target population consisted of members of web development teams and included software engineers, software testers/quality engineers and project managers. A mixed methods approach was chosen to critically explore the research problem. Accordingly, a combination of quantitative and qualitative research approaches was used.

i. The primary data for the quantitative aspect of the study was collected through survey research using a Likert-scale questionnaire. Purposive sampling was used to choose the three (3) companies to participate from the population. The questionnaires were then shared to members of development teams in each company. The quantitative data was analysed using descriptive and inferential statistics with the use of the Statistical Package for Social Sciences (SPSS).

ii. The qualitative data was obtained by conducting interviews with managing heads and leaders of each development team. To identify the interview participants, an expert sampling technique was used as a required level of expertise from the participants was needed to achieve the desired feedback. Subsequently, the recorded interviews were transcribed and thematic analysis was used to identify themes and patterns.

iii. The findings of both sets of analyses were then synthesised to provide an overall understanding and proffer future recommendations.

1.8. Significance of the study

Although similar work has been done in software testing (ST) in Nigeria, much less is known about ST and its adoption in WA development. Most related studies in Nigeria (Sowunmi & Misra, 2015; Osho, Misra, & Osho) focus on generalised software quality assurance in practice without any specific focus on security of WAs. They also do not specifically reference adoption or acceptance. Outside the Nigerian context, most existing research focuses on discovering new vulnerabilities and developing new

techniques of mitigating them (Chóliz et al. 2015; Li et al. 2014). The challenge, however is that although new ST models are constantly proposed through research, in practice these models are not readily adopted. The implementation and application of appropriate ST strategies by practicing teams in the IT industry need to be improved to enhance secure software development. It is hypothesised that the conceptual model proposed in this study will be more adaptable and efficient in explaining the challenges around adoption.

1.9. Structure of the study

The structure of the study will be as follows:

Chapter one: This chapter provides an overview and background to the study, the problem statement, study objectives and a summary of the methodology used.

Chapter two: This chapter details the literature reviewed for the study. Existing ST approaches, techniques and tools and their limitations are extensively discussed in line with the stated research objectives. The conceptual framework (CF) for the study is also discussed in the latter part of the chapter.

Chapter three: This chapter describes the research approach and methodology used in the study. It also explains how the conceptual model for the research was adapted from the TAM. Finally, an overview of the methods, sampling and data collection techniques used in the research is presented.

Chapter four: This chapter presents the statistical analysis of the quantitative data obtained.

Chapter five: This chapter discusses the thematic analysis of the qualitative data gathered for the study.

Chapter six: This chapter provides a discussion of the findings from both the qualitative and quantitative analyses presented. A synthesis of the overall findings is also provided.

Chapter seven: This chapter provides a conclusion to the study, as well as a recapitulation of the RQs and the methodology utilised. A summary of the findings and some recommendations are made for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

This chapter consists of a review of extant literature in the field of ST with respect to the development process of WAs. It explains what the current practices are, as well as what factors have been known to be significant in the adoption of appropriate ST from the research. The key parameters in the study are presented here.

Some of the key terms used in this chapter are defined as follows:

i. *Vulnerability*: This is defined as a weakness in an application. It could be a development bug or a design flaw that could be an entry point for an attack in a WA (McGraw, 2006).

ii. *Bug*/*Flaw*: A bug is an error in an application code. A flaw in an application can cause it to function inappropriately and result in a failure (Stuttard & Pinto, 2011). It can also be called a defect.

iii. *Threats*: Regarding WAs, threats are exploited vulnerabilities that can cause loss or damage in an application (Salini & Kanmani, 2012)

iv. *Attacks*: These are deliberate and focused attempts to compromise the quality, reliability and functionality of an application causing specific failures to occur (Graham, van Veenendaal, & Evans, 2008)

v. *Attacker*: An individual whose goal it is to forcefully break into an application, launch attacks and steal confidential data (Stuttard & Pinto, 2011).

vi. *Software development life cycle (SDLC)*: This involves a sequence of phases used in the development of a software application. It helps to capture, verify, and implement all requirements needed to make an application useful to an organisation (Erdogan, 2009).

2.2. Security testing definition for web applications

Websites are interfaces that serve as interaction media between users and WAs in a web browser. User inputs captured into WAs are accepted through websites and they are usually first points of exploitation in an attack. There are two types of websites, static and dynamic. Static websites are typically information driven. They are written in Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) (McMahon, Seaman, & Buckingham, 2011). Oftentimes, the information displayed on static websites do not change (Ali, Khan, Baig, & Umer, 2015). Examples of static sites are company or university information sites. Dynamic websites on the other hand contain information that constantly changes in realtime (Christ, 2005). They are built with server-side programming languages such as JavaScript and Hypertext Preprocessor (PHP) to enable the information they display to be modified in runtime. They are data-driven, where the information is fetched from a server. They can manage sessions and retrieve data from a database. They are used in transactional instances. Examples include: social media sites, ecommerce sites and online forms (Nixon, 2014). In comparison to dynamic websites, WAs are more interactive and authentication is required for basic user interactions where security is crucial in its implementation.

ST is a significant part of a WA that has been widely discussed in research and its interpretation and approach varies across software teams. It is a distinct type of testing used to verify that an information system meets its core security requirements. (Gottipalla & Yalla, 2014). It could also be defined as a process used to ascertain that the security attributes of a WA defined in its planning and design stage is in harmony with its implementation (Tian-Yang, Yin-Sheng, & You-Yuan, 2010). Security in application development is done to ensure that the application can function correctly even under a malicious attack (McGraw, 2006).

For general software, ST is usually performed to prove that the software is free from errors. In WAs, the approach to ST is quite different. Its goal is to reveal errors and possible points of failures in the application using some structured framework or methodology (Türpe, 2008). Feature parts of the application (e.g. firewalls, authentication and authorisation subsystems, access control), are critically tested for security functionalities, efficiency and availability. Concurrently, the back-end of the

application, such as the code base and integrations between APIs (Gokhale & Sharma, n.d.) are also tested to discover unexpected vulnerabilities that could have been introduced by design, architecture errors and/or coding inaccuracies. Possible attacks are simulated with the use of tools. These are the main approaches adopted in the testing phase of the SDLC of a WA (Schieferdecker, Grossmann, & Schneider, 2012).

For WAs, most security vulnerabilities discovered are usually of high priority (Gokhale & Sharma, n.d.). This is mainly due to their architecture and the dynamic nature of their operating environment. They are also unique because of the sensitive information they store and transmit and their interoperability across different platforms (Avancini, 2012). Furthermore, studies have discovered that fundamental software testing principles are not completely suitable for WAs due to their uniqueness (Doğan et al. 2014; Li et al. 2014). Several traditional software testing methods such as equivalence partitioning, path testing, and structural testing are not effective in identifying security bugs. This is because software testing assumes ideal scenarios for an application to run. For WAs, there are no ideal scenarios because the motive of the attackers is not always predictable. Different attack possibilities need to be simulated and different scenarios tested (Gupta & Singh, 2013). Test automation is another basic software testing principle which has been highly effective in finding functional software bugs but may not be suitable for certain security tests for WAs. Automated testing cannot effectively map applications to simulate possible attacks and cannot fully simulate certain security scenarios (Erdogan, 2009). Test automation processes adopted in ST need to be fully supplemented by manual techniques for effective application in ST (Stuttard & Pinto, 2011).

2.2.1. Top known vulnerabilities for web applications

WAs typically have some vulnerabilities unique to them. The OWASP is an open-source organisation that collates reported vulnerabilities across the internet. The OWASP releases an annual updated list of top ten vulnerabilities of WAs (Avancini, 2012). These vulnerabilities are as follows:

i. **Structured query language injection (SQLi):** An SQL injection (SQLi) is an attack that can cause damage to a database and corrupt sensitive data in an

application. It has been widely reported as the most dangerous web vulnerability in the last decade (Bau, Bursztein, Gupta, & Mitchell, 2010; Vernersson, 2010). In a SQLi, a malicious code is introduced into user input variables that are linked to SQL commands and serves as input to an application. This is executed and gives an attacker access to the application database. The attacker can then modify data in the database by exploiting the SQL vulnerability (Dukes, Yuan, & Akowuah, 2013). It can be tested for manually by injecting attack vectors with user inputs and verified if SQL exceptions are returned to the user. ST techniques as with code analysis and penetration testing are also used to detect SQL injections (Meucci & Muller, 2014).

ii. **Broken authentication and session management:** This occurs when authentication and session management features of a WA are not implemented appropriately by development teams. Attackers can then gain illegal access to keys, tokens and passwords and impersonate other people's identities to further exploit the application (Patil & Phil, 2014).

iii. **Cross site scripting (XSS):** This is another popular web vulnerability that has been known to cause extensive damage to WAs. It is particularly dangerous because it provides a medium for other types of attacks (Qian, Wan, Chen, & Chen, 2013). In a cross-site attack, a script is integrated with a user input on the client side and executed in a web browser of an application. The attacker then uses this to find points of vulnerabilities in the WA. The injected compromised scripts are also used by the attacker to steal confidential information and even hijack user sessions (Malviya, Saurav, & Gupta, 2013). This script then permanently stays on the application server and continuously retrieves user information for the attacker as in the case of Stored XSS (Pelizzi, 2016). Another form of XSS is Reflected XSS, in which the malicious script is sent to users in embedded links via emails. The browser then executes this script when clicked in the email by users and the attacker can then gain access to private user information in the application. XSS can be prevented by validating user inputs using static code analysis tools (Dukes et al. 2013; Qian et al. 2013).

iv. **Insecure direct object reference:** This is a defect in an application design. The access to sensitive features of the application such as the database or directory is not outrightly protected, thus exposing the feature to attacks through the application (Saripalli & Walters, 2010).

v. **Security misconfigurations:** These are often introduced into an application when the patches and security features are not updated regularly. Since WAs are usually integrated with some other objects such as web servers and operating systems, these objects need to be updated regularly with latest security patches. In addition, frameworks integrated when building the applications need to be adequately configured, updated and tested. Lack of regular updates and misconfiguration could lead to exploitation by attackers (Eshete, Villafiorita, & Weldemariam, 2011).

vi. **Sensitive data exposure:** This is a vulnerability introduced in an application when confidential data stored in parts of the application such as the database are not well protected. Sensitive data as with passwords and credit card information can be detected if not properly encrypted and stored in the database. Transmitting such data to WAs without appropriate encryption channels can lead to interception by attackers. To prevent this, these sensitive data need to be protected using appropriate ST strategies similar to the data protection API by Microsoft (Howard & Lipner, 2006).

vii. **Broken access control:** Broken or compromised access control is also dangerous to an application and attackers can gain unauthorised access into an application. They can access other user accounts, and retrieve sensitive data, modify and create access rights. Appropriate security policies need to be set and enforced to ensure that access control rights are properly implemented in the application (Jaiswal et al. 2014).

viii. **Cross site request forgery (CSRF):** CSRF deceives a victim into loading a compromised web page. The victims' web browser is then forced to send requests to a valid website along with any other authentication information and

the application executes the requests with the assumption that they are legitimate (Ding, 2013). The attacker can gain any information while the victim session is still online. They can be prevented with the use of ST tools to defend against attacks. Tokens are also used to validate requests between clients and applications (Jovanovic, Kirda, & Kruegel, 2006).

ix. **Using components with known vulnerabilities:** Many developers and administrators use outdated software even although they might be vulnerable. Lack of regular updates to web servers and software used on WAs can lead to compromise and become points of exploitation if not updated to the most recent secure stable and secure version (Atashzar, Torkaman, Bahrololum, & Tadayon, 2011).

x. **Unvalidated redirects and forwards:** WAs regularly redirect and transfer users to different third-party websites depending on location, web browser type and some other factors. However, functions analysing the users' data can be exploited by attackers if not properly validated. Users can then be redirected to phishing or vulnerable websites (Atashzar et al. 2011). These listed vulnerabilities can affect WAs and hence adequate ST strategies need to be used to detect and prevent them.

2.3. Security testing techniques, frameworks, methodologies and tools for web applications

The approach to ST varies across teams. Several security frameworks have been developed by companies and regulatory bodies over the years and adopted in different scenarios (Prandini & Ramilli, 2010). The following sub-section describes ST techniques, frameworks and tools that exist as reviewed from the literature.

2.3.1. Security testing techniques

Approaches and practices of software testing differ across different development teams. From existing research, (Kang, Cho, Shin, & Kim, 2015; Scarpino, 2010; Stuttard & Pinto, 2011), there are some known techniques that are widely for WAs. Code review,

penetration testing, vulnerability scanning and fuzzing are some techniques that are discussed below.

i. **Code analysis:** Code analysis or review is an ST process used to inspect an application source code to detect errors and defects in the code before its final integration and deployment (Stuttard & Pinto, 2011; Vernersson, 2010). Code review is used to ascertain the level of quality and security of an application without actual execution of its source code (Mao, 2009; McGraw, 2006; Shema, 2011). Usually, it is carried out by developers or leaders of development teams and requires high technical skills to be done effectively. Oftentimes, it is intensive to conduct manually and might require the use of tools. Static code analysis is the automated aspect of a code review process with the use of tools (Avancini, 2012). These tools identify potential points of vulnerabilities in the WA source code. It is also known as white-box testing.

ii. **Penetration testing:** This is a widely known and adopted approach to ST. It is executed by simulating attacks with tools and hitting the target application to observe and discover vulnerable points in its build (Dukes et al. 2013). Discovered points of vulnerabilities in the application are exploited and the behaviour of the application under critical attack conditions is observed. This helps development teams to build in more security into the WA so that it still functions even under malicious attacks (Dukes et al. 2013; Shema, 2011; Vernersson, 2010). Penetration testing also provides metrics for evaluating the secure state of an application for easy tracking for regression tests and re-fixes. It is also referred to as black-box testing. Independently these test techniques need to be integrated into an SDLC for effectiveness (Kang et al. 2015; Scarpino, 2010; Stuttard & Pinto, 2011).

iii. **Vulnerability scanning:** This involves the use of tools (i.e. vulnerability scanners) to explore an application to identify vulnerabilities. The vulnerability scanner automatically analyses a website or WA interface in a bid to discover exploitable vulnerabilities like SQLi and XSS, CSRF etc. These scanners can also offer mitigation advice and generate compliance reports for detected WA

vulnerabilities (Bau et al. 2010). A restriction is that there is limited coverage to the detection rate of scanners and some produce false positives (Stuttard & Pinto, 2011).

iv. **Fuzzing:** This is a testing technique that is generally applied in ST for different types of applications. This technique involves applying random input and data into WA to test its resilience and attempt to 'break' the application (Felderer, Büchler, Johns, Brucker, Breu & Pretschner 2016). The data input in the application could be applied directly to the WA using command line inputs or compiled programs. The input could also be generated using tools to create a wider range of inputs as in the case of mutation-based fuzzing. (Federer et al. 2015).

2.3.2. Security testing frameworks and methodologies for web applications

Security methodologies have been discovered to be the most structured, organised and applicable approach for ST especially for WAs (Srinivasan & Sangwan, 2017). They consist of developed abstract models that portray security scenarios. They are used to identify vulnerabilities as well as procedures for developing appropriate test plans from the models (Prandini & Ramilli, 2010). They are quite comprehensive and suitable for both small and large-scale scenarios. Some popular methodologies have been identified in the security community and used in different scenarios with some level of accuracy and assertion. Widely known examples are the open source security testing methodology manual (OSSTMM) and the OWASP which has been mentioned earlier. (OWASP, 2016; Prandini & Ramilli, 2010; Srinivasan & Sangwan, 2017).

Security frameworks are used to assess the security risks in a system. Usually, developed with standards by regulatory governing bodies in the IS security community. A typical example is the NIST-SP 800 framework developed by the National Institute of Security Standards. Another known security framework is the Penetration Testing Standard Framework (PTES). These frameworks usually form the basis of most security tests. They could predict how a WA would tolerate an actual attack, as well as the level of complexity needed by an attacker to weaken the system. They are also known to depict countermeasures that could be used to subdue and prevent threats

against the WA. A typical framework usually consists of series of steps which transition from the planning phase to the exploit and reporting phases in application development (Howard & Lipner, 2006).

A methodology differs however from a framework. Herzog (2010) describes a security methodology as a security process that incorporates the necessary security plans, documentation and designs. It also includes the needed data (i.e. possible vulnerabilities) to validate these security mechanisms. A methodology can be used to make comparisons between test results and validate test outputs. It can minimise false positives based on sets of controls and rules that are dependent on the process and technologies involved. It can also provide a concise report with valid metrics more comprehensive than a framework (Herzog, 2010).

Although widely used in WA development, frameworks can be limited in their adoption due to the fact that they are rapidly updated as new vulnerabilities and security tests are gradually discovered in practice. Furthermore, different frameworks are independently designed but not applied in practical industry scenarios and companies stand a huge risk in adopting them (Srilatha & Someshwar, 2011). The levels of awareness of methodologies and frameworks will be examined in what follows.

2.3.2.1. *The Open Web Application Security Project Framework (OWASP)*

The Open Web Application Security Project (OWASP) is known for its advanced work in the security industry. Established in 2001, the OWASP has established itself as a body fully versatile in security research. Aside from publishing the annual top-ten security vulnerability report, it has developed several standards, guidelines and methodologies that have been applied in ST, across both mobile and WA platforms (Meucci & Muller, 2014).

The OWASP testing guide is the most adopted framework in WA development (Kang et al. 2015; Srilatha & Someshwar, 2011). It is a guide that contains best practices for ST and it is particularly unique because it scales ST down to the lowest levels in the SDLC. It is rich and includes activities that span all SDLC phases (Kang et al. 2015).

As an effective framework, it continues to evolve and enjoy contributions from various researchers across the world.

The OWASP testing guide has defined activities throughout the SDLC phases. These phases are described as follows:

a. ***Pre-development phase/planning***: The first step in the OWASP seeks to identify a typical SDLC that allows for an integration of a security feature in each of its phases. Since this stage involves the planning, the necessary documentations available for the project are reviewed and security policies and standards are identified. These standards serve as a guideline to development teams before building the application. Specific secure coding standards for certain programming languages are identified (i.e. Java) (Tittle, Stewart & Chapple, 2006), and cryptographic measures are set if need be (McGraw, 2006). In addition, security measurement metrics are defined to provide defect visibility in the SDLC and the final application.

b. ***Design phase***: In this phase, security requirements set for the application are reviewed and tested to verify the inferences and possibilities assumed for possible gaps. Requirements are clarified based on known security mechanisms such as confidentiality, integrity, authentication, authorisation, session management, legislative and compliance standards, among others. The architecture and design of the application is also reviewed to ensure that they accommodate the security levels in the requirements. Identified flaws are easier to fix in this phase, as modifications and changes to design are less costly compared to latter phases in the SDLC (McGraw, 2006). Visual models illustrating the architecture of the application are usually created with the Unified Modelling Language (UML) to help SDTs better understand how the application is expected to work (Doğan et al. 2014).

c. ***Development phase***: In this phase, the application code is carefully developed with the policies and standards set in the planning and design phases. Planned security requirements are built into the code of the application by the developers. Code walkthroughs are carried out within development teams to understand the

structure of the code from a high-level and identify possible unexpected inputs as well as invalid classes that might have been inputted in the code by developers (Meucci & Muller, 2014) Code reviews are also performed by testers to check for security defects. Certain security checklists from OWASP are used during the code review to ensure proper coding practices have been followed. Language-specific issues are also checked and confirmed to be accurately applied (Erdogan, 2009).

d. ***Post deployment:*** Regardless of all the planning and implementation in the other phases of development, some security bugs may have evaded discovery in any of the previous SDLC phases. After the application has been deployed, post-deployment scans and checks are important to check for missed bugs in previous phases (Graff & Van Wyk, 2003). Penetration tests with tools are carried out first in this phase and then configuration management tests are performed to check if the integrations with infrastructure is compatible and secure. Integration with other hardware or database could create vulnerability points that might be exploitable (Khan & Singh, 2012).

e. ***Maintenance and Operations***: The maintenance phase includes all the ST processes needed during the operational phase of the WA. It involves periodic checks that are scheduled to verify the security state and overall health of the application. Change management and updates are done carefully and security tests are carried out to verify that the security of the WA has not been compromised (Williams, 2006).

The OWASP guide is comprehensive and has been asserted as the most appropriate methodology for WAs. It has been adopted in many studies relating to the securing of WA and even cloud computing (Kang et al. 2015; Kaushik & Mohan, 2013).

2.3.2.2. *The open source security testing methodology manual (OSSTMM)*

The OSSTMM is a security methodology developed in 2001 by the Institute for Security and Open Methodologies (ISECOM). It was the first openly-available security methodology and became widely adopted among emerging companies who needed to

have some measure of validating their security processes (Vernersson, 2010). The OSSTMM helps to measure security at an operational level. Its emphasis lies in identifying variables for testing, defining the assets to be protected, and categorising the protection mechanisms and controls for testing. It also defines measures for detection of the vulnerabilities, weaknesses, concerns and points of exposure which could be possible limitations in an operational setting (Herzog, 2010). It encompasses tests from human and physical channels, telecommunications and data communications. It possesses a set of attack surface metrics that provide a graphical representation of changes in state of attacks, integrating with a dashboard for management inspection. It is well-suited for use in virtual infrastructures, cloud computing and mobile communications as well as high-risk security scenarios. The OSSTMM is particularly used in auditing and quantitative risk management and can be easily integrated into laws and policies. It is also adaptable for penetration tests, security assessments and ethical hacking (Erdogan, 2009). An advantage of the OSSTMM is that it defines the target clearly during assessment and the results are also easy to reproduce. It is free and suitable for an overall security assessment from an operational level. Furthermore, it encourages a large amount of documentation and has good indications for quantifying results and planning (Erdogan, 2009). Its application is however limited to post-deployment and maintenance practices.

When compared, the OWASP and the OSSTMM are different. While the OSSTMM is widely applicable to a range of scenarios and primarily used for auditing and operational security testing of various information systems not limited to WAs alone (Shanley & Johnston, 2015). On the other hand, the OWASP has established itself as the framework best suited for WAs (Stuttard & Pinto, 2011). The OWASP is better for WAs because it addresses vulnerabilities and threats specific to WAs. The OSSTMM is also more suited for designing organisational security strategies for infrastructure, networks, hardware and facilities within an organisation.

2.3.3. Limitations of security testing methodologies and frameworks

A known challenge with the adoption of these frameworks is usability. Despite the continuous developments of ST frameworks, SDTs often face difficulties in using the frameworks due to the inadequate documentation and complexity (Srinivasan &

Sangwan, 2017). Because of inadequate documentation, the frameworks are not completely clear and descriptive enough for teams to understand. As a result of these limitations, frameworks are often neglected (Srinivasan & Sangwan, 2017). Smaller development teams also face challenges in using security methodologies because of the structure and intensive process that is required in its implementation, which can be time consuming (Williams, 2006).

Certain researchers however have recommended that careful assessments should be made at the beginning of the SDLC to decide which methodology will be most applicable for each enterprise and would produce the most efficient results (Kang et al. 2015). In addition, the compatibility of the selected security methodology and the SDLC type is important for best results as there is no 'one-size fits all' methodology for optimal results. (Erdogan, 2009).

2.3.4. Security testing tools and guidelines

The ST process is iterative and intensive. Tools and guidelines have been known to ease the process and produce results. They will be discussed below:

2.3.4.1. *Tools/Toolkits*

Tools are used to automate the ST process as they are necessary to carry out the repetitive and extensive security tests in the SDLC (Howard & Lipner, 2006; McGraw, 2006). Tools ease the complexity of the testing process so that development is faster and the applications are deployed within allocated time schedules. They also provide a visual metric for understanding points of attack as they can generate reports for testing activities. They are used in any phase of the SDLC for different purposes (Dukes et al. 2013; Finifter & Wagner, 2011). There are tools for analysing source code (Jovanovic, Kruegel, & Kirda, 2006), vulnerability scanning, configuration analysis, and database scanning (Bau et al. 2010). While some of these tools are specific, others are generic and can perform multiple tasks (Erdogan, 2009). Some tools are also used to detect specific web vulnerabilities such as XSS, CSRF and the SQLi (Bau et al. 2010).

An extensive review of the tools available for ST, and categorised according to function and vulnerability detection has been offered in previous research studies (Kang et al.

2015; Srinivasan & Sangwan, 2017). A major limitation of security tools for static and dynamic ST is that most tools produce substantial rates of false positive or false negative results in vulnerability detection (Howard & Lipner, 2006). In addition, security tools are not effective without experienced personnel to validate whether the findings and results generated from the tools are legitimate.

Tools are useful in scaling the application security process and enforcing policy compliance (Howard & Lipner, 2006). Typical examples of tools are the WebScarab by OWASP, Acunetix, Netsparker, Burp Suite and many others (Dukes et al. 2013). There are also toolkits which are developed packages consisting of sets of testing tools for different functions.

2.3.4.2. *Guidelines*

Guidelines are steps used to organise the process and structure of ST based on applied best practices identified from experiences obtained from the field of security. These outcomes and practices are first filtered and refined and then used to develop appropriate guidelines. Although they can be practical and suitable for certain scenarios, guidelines usually lack the level of technicality needed to develop a detailed security test plan for a WA. Worthy of mention is the NIST guideline for information security – the common criteria for information technology security testing – which specifies procedures and requirements for accrediting testing laboratories (Horlick, 2005; Prandini & Ramilli, 2010).

2.3.5. Challenges and limitations of tools and guidelines

A major challenge with tools in ST has been the general lack of standards for ST tools development. Most tools in the market are developed without a proper conduct and assessment of real-world scenarios. Appropriate validations are not done to validate and verify most tools (Türpe, 2008) and as a result, classifying them for use in varying scenarios is a huge difficulty (Erdogan, 2009). Most of these tools and frameworks have also produced results with false positives and negatives and do not proffer a method for mitigating these false positives (Hope & Walther, 2008; McGraw, 2006).

While there are many companies building and deploying tools into the tech space, most of these are limited in performance. Although they help to scale the ST process and enforce policy, they do not secure the application. These tools are also designed to be suitable for general applications and may only discover generic problems. For individual application projects, they may not have enough ability to be able to adequately detect flaws (Meucci & Muller, 2014).

2.3.6. Impact and significance of security testing in the SDLC of web applications

The SDLC is the basic prototype for development of software artefacts. It describes a series of sequential processes that need to be considered in building a software application and defines the critical steps and phases of such development. It also highlights the goals of each stage of a software application life cycle from conception to final deployment to the end user (Singh & Kaur, 2019).

2.4. The Software Development Life Cycle (SDLC)

The SDLC is the basic prototype for development of software artefacts. It describes a series of sequential processes that need to be considered in building a software application and defines the critical steps and phases of development (Massey & Satao, 2012). It also highlights the goals of each stage of a software application life cycle from conception to final deployment to the end user (Cruzes et al. 2017,).

Due to the varying nature of IS and the rapid advancement in their structures, a variety of SDLC models have been developed. Notable examples are the Waterfall model, the V-model, RAD model, and Agile/Scrum model (Singh & Kaur, 2019). Some of these models are sequential and require a step of the process to be completed before another, while others are highly iterative with phases having been interwoven and repetitive. The sequential models are directed toward producing an integrated working software application towards the end of the life cycle, while the iterative models are executed with a focus to produce working features of a software at intervals in the cycle (Munassar & Govardhan, 2010).

Several studies (Srilatha & Someshwar, 2011; Erdogan, 2009) have sought to compare and identify the advantages of most widely-adapted SDLCs and discover the limitations for each with respect to WAs. These SDLCs are discussed below in respect of their significance and limitations for security. The two major SDLC notably popular for WA development are the Waterfall and Agile/Scrum models (Erdogan, 2009).

Figure 2.1 depicts a typical SDLC and its phases (Erdogan, 2009).

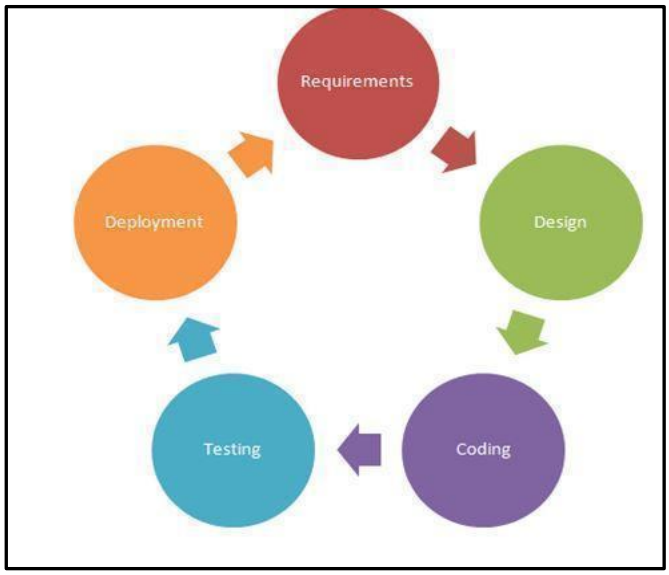

Figure 2.1. The software development cycle (adapted from Massey & Satao, 2012)

2.5. Known SDLC types used for Web applications

Each of the known SDLC types have processes for integrating security in each of their phases. These SDLC methodologies have also known limitations with respect to their approaches and activities. Their limitations will be discussed in the section which follows.

2.5.1. Waterfall SDLC and security

The Waterfall model is the oldest and most adopted SDLC model (Lewis, 2016; Paul, 2016). It is widely used in aspects of software engineering such as mobile and web development. It accentuates early planning and design so that flaws can be identified as each of its sequential phases. In addition, it has well-defined requirements (Lewis, 2016), proper documentation and review. It is also easy to comprehend and design and

widely suitable especially for small teams and mature products. It specifies deliverables for each stage and reinforces ethical software engineering practices which involves appropriate quality and security measures (Munassar & Govardhan, 2010).

In terms of ST, a major disadvantage of the Waterfall model is that although the planning phase seeks to identify the needed requirements for security, the testing comes after the application has been developed. Errors made in the earlier phases of the cycle aggregate until the latter phases. Accumulated errors are expensive to rectify as cost of rework is usually high towards the end of an SDLC (Singh & Kaur, 2019). Many security bugs would have escaped discovery, and this could lead to drastic consequences after deployment. Another limitation of the Waterfall model is that testing and development are kept as separate entities (Lewis, 2016) as opposed to the ideal process of integrating testing with development.

2.5.2. Agile SDLC and security

The Agile methodology is a contemporary and developmental model. It was designed to eliminate many of the challenges of other traditional SDLC models with much simpler and efficient methods. The Agile manifesto was first created in 2001 (Fowler & Highsmith, 2001) and its ideology focuses on functional software over comprehensive documentation. It also places emphasis on individuals and interactions and deems these more important than processes and tools (Broström, 2015). Its continuous and highly iterative nature makes it suitable to ensure a faster delivery of software to the market and end users, leading to its wide adoption by many development teams (Cruzes et al. 2017). It also helps to foster collaborations across teams.

A major downside however of the manifesto that drives the Agile methodology is that it does not have any strategies for security (Chóliz et al. 2015). The basic approach to ST in a typical SDLC involves detailed planning before development, implementation of plans and use of tools in certain instances. On the other hand, the Agile manifesto focuses on user interaction over processes and tools. It also stresses responding to changes over following a plan (Wotawa, 2016). ST in Agile development is done at different phases. The security tests in the phases are discussed as follows:

i. **Unit tests:** These are tests that are run to check the smallest components of an application before its integration with other components. Small units of code in the application are tested to certify that it does its desired function (Broström, 2015). In the Agile SDLC, these tests could be code-compiled programs that are run to validate security endpoints in the WA. The tests are continuously done on feature units within the application and bugs can be easily identified and fixed (Hope & Walther, 2008; Khan & Singh, 2012).

ii. **Integration tests:** These are end-to-end tests carried out after all the individual units and parts of the application have been integrated (Cruzes et al. 2017).

iii. **Automated regression tests:** These are tests suites that are run repeatedly on an application after deployment. These test suites ensure that previously fixed security bugs do not reoccur after a new build or feature is added to the application (Andrews & Whittaker, 2006).

In order to be effective while still ensuring a fast rollout time to market for a given product, integrating security into the Agile SDLC should be performed with the most apt approaches. A general approach that has been widely researched is the use of static code analysis on the source code for each build or unit of the application. This is also called automatic security testing. Studies have however called for a more formal approach to ST within the Agile development life cycle (Cruzes et al. 2017).

2.5.3. Security testing practices in each phase of the SDLC

Each unique phase in the SDLC has specific security goals that have been discovered through research and practical applications in the web development industry. In addition, each phase has its own challenges. These include:

i. **Requirements phase:** This phase typically involves identifying security requirements, determining known flaws and attacks for each requirement and then mapping them to the specific application to be developed. Requirements and design documents usually involve security goals composed of a unique identifier, a title and well stated description (Assad, Katter, Ferraz, Ferreira, & Meira, 2010). Possible attacks are obtained from attack libraries and from

observing attack patterns of similar applications and included in the requirements (Mouratidis & Giorgini, 2007). Checklists for verifying security goals of confidentiality, integrity and authentication are also set (McGraw, 2006). In some systems, several security constraints are defined for stakeholders with security reference diagrams (Vernersson, 2010) (Mouratidis & Giorgini, 2007). A major challenge however is that requirements and design documents are hardly available for WA projects and thus it becomes difficult to define the behaviour expected. This challenge leads to the introduction of application logic flaws (Felmetsger, Cavedon, Kruegel, & Vigna, 2010).

ii. **Design phase:** Threat modelling and design review is usually performed in this phase. After analysis, identified attack surfaces, input and output nodes are modelled into attack trees. This attack trees illustrate the paths from an attacker to an input interface within the WA. Paths that depict successful attacks are modelled (Vernersson, 2010). Possible points of exploitation are identified across interfaces and measures to mitigate them are outlined and documented. This is important as it provides an understanding in the effect of any modifications to the architecture of the application. Documented threats and security decisions are also classified by severity and impact (Lebanidze, 2006). This phase is particularly important because flaws in this design phase are responsible for fifty-percent of security issues in applications (McGraw, 2006).

iii. **Implementation phase:** This stage involves examining application code for defects or coding mistakes without an actual execution. Code is reviewed and security tools are used to find bugs which are fixed consecutively (Dukes et al. 2013). This technique is effective in finding security bugs in code (McGraw, 2006; Stuttard & Pinto, 2011).

iv. **Testing phase:** This phase involves testing the application to find out security bugs and defects. Random inputs are applied to the WA to observe certain behaviours and responses in a bid to break the system as in the case of fuzzing. Penetration testing tools are used to access parts of the application and find possible points of failures and vulnerabilities (Srilatha & Someshwar, 2011).

v. **Maintenance phase:** This involves all post-deployment security checks and scans. Configuration management tools are used here to perform a version control of the application build (Khan & Singh, 2012).

Although each phase has significant security goals, most organisations typically delay security processes to the end of the life cycle. Gary (2006) describes this as a "penetration and patch" approach. In as much as it could be applicable in some scenarios, it is a process that is reactive rather than proactive. Often, the security problems of applications are embedded deep within the code making this reactive approach highly unsuitable (McGraw, 2006).

Security is a critical quality that must be built into the application development process. Each phase has its unique vulnerability potential and these should be considered by all stakeholders involved in the life cycle of the WA (Gupta & Singh, 2013; Kaushik & Mohan, 2013). Figure 2.2 illustrates the ST efforts estimated in each phase of the development life cycle. Security efforts in the planning and design phases should be larger than other phases of the SDLC.

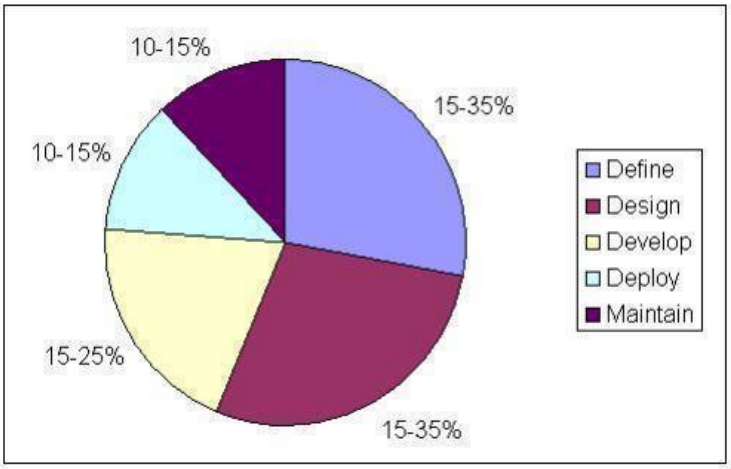

Figure 2.2. Test Effort required in the stages of the SDLC (adapted from the OWASP Testing guide Williams, 2006)

There are ST activities in each phase of the SDLC and these activities produce certain software artefacts. Figure 2.3 illustrates the various ST activities typically applied in each phase of the SDLC and the test artefacts produced from each activity.

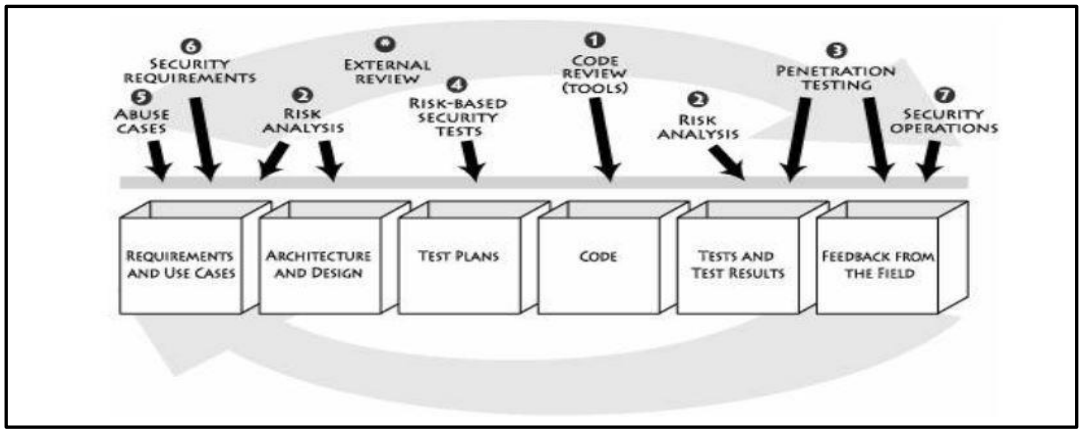

Figure 2.3. Security testing activities from software artefacts in each phase of the SDLC (adapted from McGraw, 2006)

Security requirements and abuse cases are defined in the requirements gathering phase. Abuse Case defines interactions between a user and an application that results in a security flaw to a resource associated with other users of an application, or the application itself (McDermott & Fox, 1999). With the Abuse Case, risk analysis is performed to identify the critical features of the application to be prioritised. This is used in the architecture and design of WAs. Test plans are used to define how to test WAs and risk-based tests to be executed are outlined in the test plan. The code which is the output of the development phase is examined using ST techniques such as code review with the use of ST tools. After the development phase, the testing commences and produces results of tests carried out on the WA. These test outputs are further examined and a proper risk analysis is performed to determine specific areas of vulnerability in the WA. Penetration tests are then applied to the feature parts identified. After the WA has passed tests based on the risk analysis, it is deployed to business users as in the case of User Acceptance Testing (UAT). The feedback is then collated from the users and reviewed again to further improve the security of the WA. Further security operations are then carried out after the WA is integrated with infrastructure and other types of applications. As depicted in Figure 2.3, a structured security approach throughout the SDLC helps to protect the application from known risks and certify quality (Gottipalla & Yalla, 2014).

2.5.4. Importance of security testing in the SDLC

ST plays a significant role in the quality and security of WAs. Certain factors indicate the important value that ST has in a typical SDLC. These are as follows:

i. **Cost of software bugs:** From research to industry feedback, application development is a costly venture. The quality characteristic to quantify in ST is cost. Most organisations consider security a costly investment with very little return and business value (Sowunmi & Misra, 2015). However, because corrections and bug fixes cost twenty to a hundred time more during implementation, testing and maintenance, overall savings outweigh the extra expense that such security could incur (Lewis, 2016). Figure 2.4 graphs the cost effect of fixing bugs through each phase of the SDLC.

   An exploitation or attack on a WA could lead to unavailability of the application to critical users and clients. The impact of this inaccessibility is high on business operations (Patil & Phil, 2014). This can be measured in terms of quantity or quality. Quantitatively, revenue is being lost for every minute that the application is unavailable. The cost to fix the downtime issues and restore the application back to normal operations can be very high. Consequently, the qualitative effects of these exploitations largely affect businesses. Figure 2.4 provides a graphical Illustration of how the cost of fixing defects increases towards the end of the SDLC.

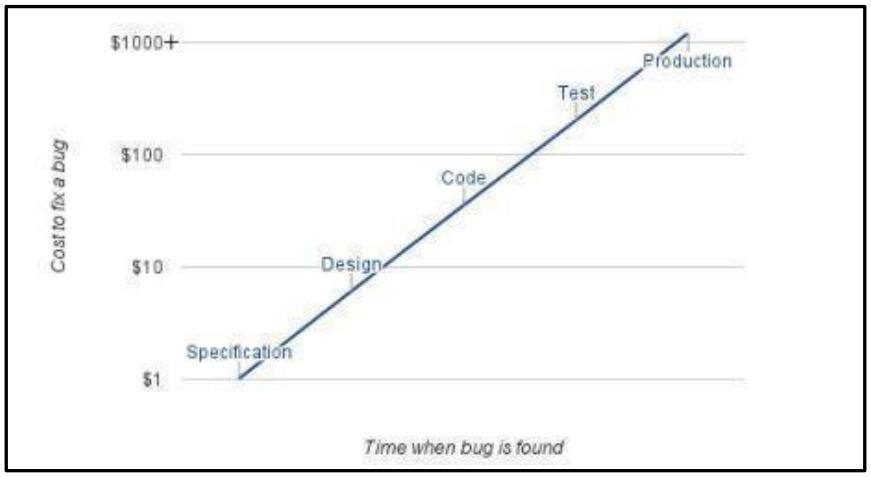

Figure 2.4. Cost of fixing software defects at each stage of a software development life cycle (Graff & Van Wyk, 2003)

Besides monetary costs, loss of credibility with customers and loss of brand reputation are qualitative costs that could adversely affect organisations due to security defects. Hence, it is highly imperative to involve all stakeholders and ensure that appropriate security practices are applied throughout the life cycle of WAs (Paul, 2016).

ii. **Risk:** The development of devices, applications and their inter-related operations bring certain risks. These revolve from many factors and are constantly on the increase. Security weaknesses and attacks are also becoming more varied and sophisticated. While many researchers are trying to find better ways to understand the nature of these attacks and proffer better ways of defending applications, organisations with so much at stake and limited resources resort to speedier options such as penetration tests rather than follow a methodology or framework (Kang et al. 2015).

2.6. The Nigerian IT industry and software development challenges

Software engineering is a rapidly growing industry in Nigeria (Boakye, 2014; Ogunsola, 2016). Many organisations are gradually switching from the conventional outsourcing style to developing their own software to produce services for their clients (Casado-Lumbreras, Colomo-Palacios, Ogwueleka, & Misra, 2014). Of this class, WAs top the list of software types popularly developed (Ogunsola, 2016). The application domains range from ecommerce, digital marketing and advertising, payments, development and health.

The practice of software development in general has however faced some challenges over the years. One distinct characteristic of the Nigerian software industry is that the larger percentage of software development companies within the country are largely start-ups and small to medium scale enterprises (SMEs) (Obasemo, 2015). The teams are small and usually consist of a few individual developers who build applications and set targets to roll out to clients in time (Ogunsola, 2016). Due to this, proper development measures and other ethical considerations involved in software engineering are largely neglected (Binuyo, Oyebisi, Olayinka, & Afolabi, 2015; Sowunmi & Misra, 2015).

Another constraint revealed through research has been the inadequacy of quality resources available to companies. These resources are both human and non-human. The challenge of finding skilled human resources has been a major issue in Nigeria's software industry over the years (Binuyo et al. 2015). Most companies simply recruit graduates of computer science or IT/IS and believe that they can perform on the job. A large percentage of these graduates are not properly trained with the needed skills for the industry and oftentimes find it hard to appropriately implement secure development strategies. Experienced professionals also struggle with the challenge of applying ethical and standardised measures in building their products because of the demands and pressure during development (Sowunmi, Misra, Fernandez-Sanz, Crawford, & Soto, 2016; Ume & Chukwurah, 2012). Most SMEs cannot finance appropriate technical training required for increased productivity on the job due to limited revenue (Dukes et al. 2013).

The popular approach to software testing among Nigerians places emphasis on manual testing. While automated testing has a wide range of advantages, the importance of manual tests in ST should not be disregarded. Vulnerabilities are highly diverse and unpredictable and can require the careful inspection and expertise of a tester in some instances. Manual testing has been stated to be outrightly important for ST (Dukes et al. 2013). It should be conducted systematically alongside automated tests so that more vulnerabilities can be found in the application.

Generally, software testing in Nigeria has grown consecutively alongside WA development. The dynamic nature of WAs, constant change requests and new customer demands has created a need for software testing engineers to be involved in the SDLC to detect bugs and prevent flaws. Errors and flaws in the application need to be highlighted by testers and reported for early fixes. After application deployment, software testers spend a great deal of time to ensure that the WA continuously works as expected. Regular updates are usually applied to upgrade and improve on WAs and SDTs and testers run regression tests at intervals to ensure that the updates do not affect or break any of the previous core functionalities of the application.

2.6.1. Factors that affect software development practice in the Nigerian IT industry

The structure and practice of software testing in Nigeria has faced many challenges over the years. Due to constraints of project size, time, and resources most software development processes have been found to lack quality and security (Sowunmi & Misra, 2015). An assessment of software engineering ethics in Nigeria software companies was made in 2015 (Sowunmi & Misra, 2015) and the results revealed a number of challenges that have affected software development in the country. These include:

i. **Inadequate support:** Nigeria occupies a significant position in information systems development in Africa (Odufuwa, 2012; Ogunsola, 2016). The awareness and increased growth in its software development industry has been consistent and rapid in recent years. However, inadequate support from management and officials of individual companies and that of the government has been a huge factor limiting the adoption of appropriate IT processes of which security is a part (Irefin, Abdul-Azeez & Tijani, 2012). Technology advances continue to influence political and economic decisions, policy making and distribution of needed IT resources to empower organisations. Consequently, there is need for involvement on the part of the government (Gotterbarn, Miller, & Rogerson, 1999) and business management of companies. WAs are very useful for businesses and have gradually removed the constraint of reaching out to customers; however, many organisations still have some form of resistance because of security concerns. Support from both business management and the government is thus urgently needed (Gupta & Singh, 2013; Oyetoyan, Cruzes, & Jaatun, 2016). ST is an important part of software development.

ii. **Inadequate adoption of appropriate software engineering ethics:** A major challenge that constantly faces the typical software engineer in the Nigerian tech industry is the lack of a proper code of ethics to guide the practice of software engineering (Ume & Chukwurah, 2012). Because there is no authoritative regulatory body, certain problems have emerged which confront software

development teams. The choice of the development methodologies to use, what defines the final deliverables and the people to be involved in the development process are usually of major concern (Osho, Misra, & Osho, 2013). Over the years, teams have found a way to create their own idea of appropriate practices, but since there are no regulatory standards, the quality of applications built with this process are compromised (Binuyo et al. 2015). Standardised and regulated ethical guidelines help to improve the performance of teams and consequently impacts on the quality and security of a software application developed. In 2012, Ume and Chukwurah identified the importance of a structured ethical guide and stated that it was important to guide the profession by setting out well laid-out policies for members of a development team to follow. The inadequacy or lack of such ethical standards is costly to the profession as individual members will have no pattern or model to follow in implementing their duties (Ume and Chukwurah, 2012).

A lack of ethical guidance in the industry affects the perceptions of individual members of teams about the usefulness and importance of ST. Consequently, this leads to an apathetic disposition and indifferent attitude by members of SDTs in implementing basic security processes in development (Sowunmi & Misra, 2015).

iii. **Inadequate implementation of policies:** The processes of IT in development and practice in Nigeria are being managed by The National Information Technology Development Agency (NITDA). In 2012, the NITDA released a national ICT policy to help fulfil the nation's IT developmental objectives (NITDA, 2012). This policy consisted of different focal points in ICT, such as infrastructure developments, local manufacturing of software and hardware, and national security among others. The policy contained different strategies which were highly aspirational, yet had no clear-cut details on how they would be implemented (Odufuwa, 2012). The policy was developed to facilitate socio-economic growth but has not been effective across markets. Because of the continuous and invasive growth in the industry, policies need to be set, as well as constantly reviewed and upgraded to accommodate emerging trends and

changes so that the potential of the ICT industry can be maximised (Manjula, Bagchi, Ramesh, & Baskaran, 2016).

Policy is important as it helps to protect data from being compromised. Furthermore, it helps to form a system of acceptable practices for users of technology where implementing such policies is an essential aspect of information security. However, strategic policy implementation has been a critical hindrance to growth within the Nigerian economy (Odufuwa, 2012). A typical example is the cashless policy introduced by the Central Bank of Nigeria in 2012. It was originally developed to encourage the adoption of electronic-based payment transactions and to gradually regulate the distribution and circulation of currencies. With this policy, the Federal Government aimed to decrease the risk and high cost of cash transactions and tackle corruption. The policy was also created to encourage traceability of transactions, manage inflation and promote all round economic growth (Eze, 2013). Although this policy took effect for a while after its creation, its implementation has drastically declined over time (Eze, 2013). This policy had a potential of increasing the adoption of electronic transactions and motivate organisations to adopt WA technologies in their businesses (Oyewole, El-Maude, Abba, & Onuh, 2013). However, many organisations still refuse to implement the policy. Several researchers have studied this problem and discovered that security has been one of the major factors affecting the adoption of the policy among organisations (Ume & Chukwurah, 2012). Several other studies related to the Nigerian IT industry have also affirmed that policy implementation is still a challenge which consistently impacts on the quality of the output and growth of the industry (Ogheneovo, 2014; Sowunmi & Misra, 2015; Binuyo et al. 2015).

iv. **Project timelines:** WAs are usually developed with a specified project timeframe in most structured organisations. The scope of complexity, time to market, client expectations, market competition and available resources are some determinants of a project timeline. In theory, team members and stakeholders meet at the inception of a project to plan and set targets and milestones. As the project commences, several unexpected and unplanned

scenarios often occur that affect the set timeline target and cause delays. Instances of unclarified requirements or new change requests can occur and need to be merged into the project. Gradually, these changes cause delays which accumulate towards the end of the SDLC, and most times affect the testing phase (Binuyo et al. 2015). Likewise, stakeholders and management may also set unrealistic timelines due to market demand and competition (Sowunmi & Misra 2015). As a result, SDTs are often constrained even before the implementation of a project. In a bid to meet these timelines, teams are forced to compromise on certain important parts of the SDLC, such as security towards the end of the cycle (Binuyo et al. 2015). Some teams adopt the use of penetration testing tools to exploit the WA for bugs and security vulnerabilities as a last resort. These approaches usually lead to late discovery of accumulated bugs that are often very costly to fix at the end of the cycle. Some important security bugs could also have been evaded due to the constrained timelines. In other instances, some organisations do not even bother to do any form of ST in the SDLC (Stuttard & Pinto, 2011). They simply complete a development cycle and out-source the ST process to external security consultants. Indigenous and independent security consulting firms are invited to offer their ST services and identify security flaws. Research has revealed that these firms use techniques that are involve developing exploit libraries from regularly occurring and widely known attacks, and then re-using these libraries to verify new applications (Atashzar et al. 2011). This method is ineffective and may be unsuitable for some applications with unique architectures and functionality.

v. **Lack of skilled professionals:** The software industry in Nigeria has been known to experience a shortage of skilled IT professionals especially in application development (Binuyo et al. 2015). The art of ST itself is a skill that is very complex to master and thus requires a high level of expertise. In a recent study Binuyo, Oyebisi, Olayinka and Afolabi (2015) have revealed that inadequate skilled personnel are a major factor affecting the quality of software products. Software teams are constantly overwhelmed with huge workloads and in some instances, are required to work extra hours because of the lack of skilled members. Such skill shortages slow down work processes, and reduces quality

in WAs as the few skilled team members cannot effectively combine multiple responsibilities together. In other instances, team members are involved in virtually every step of the SDLC which makes them ineffective (Sowunmi & Misra 2015). SDTs are expected to be familiar with a wide range of web technologies and web interfacing terminals. They are also required to understand how basic implementation flaws can introduce vulnerabilities into a built application. Yet, because of the huge demand on businesses, organisations are left with no alternative than to employ semi-skilled personnel in the hope that they will grow and learn on the job (Binuyo et al. 2015). This seldom happens because the demands of the job become overwhelming over the long term. Accordingly, ST becomes relegated to an add-on task that is generally neglected as most teams believe that it requires special skills to be performed (Ogheneovo, 2014). Another important requirement is the ability to carry out both manual inspection and be equipped to use several security tools to identify the security vulnerabilities in the earliest possible time.

vi. **Cost of Security Infrastructure:** ST involves both manual approaches and the use of tools for automating the process. The tools used for aspects of ST such as static code analysis and penetration testing are normally commercial off-the-shelf tools from external security vendors (Wotawa, 2016). These tools are designed to detect known WA vulnerabilities and can be configured for use to suit the need of different organisations. They are quite expensive and often the budget for the WA project may not be able to accommodate the extra costs involved in acquiring these tools (Mao, 2009; McGraw, 2006; Scarpino, 2010).

Besides the cost of purchasing, the cost of installing and integrating these tools to fit into each WA project is also to be considered. Some might require the external consultants to be present within these teams for training and clarification. In many instances, many would rather avoid ST. Cost is a major factor especially in the Nigerian software industry (Binuyo et al.).

vii. **Insufficient on-the-job training:** The field of WA development is a widely growing one. Around the world, several new features, libraries and technology are constantly being introduced through research and in practice. Research and

development teams in big enterprises constantly develop new tools and approaches for ST in WAs. New technologies are being introduced to the field. A study by Sowunmi & Misra (2015) revealed that although these technologies exist, many SDTs in Nigeria cannot effectively make use of them because the training is lacking. In addition, bureaucracy and politics within organisations override the demand for training for teams. Binuyo et al. (2015) corroborated this study and stated that on the job training was generally lacking in most Nigerian companies. In their study, a large percentage of the participants agreed that training was a major factor affecting the quality of the software products developed within their organisations. This is a huge challenge as the ST implementation should include a thorough training for developers (Mahendra & Khan, 2016).

viii. **Nature of business:** Security is very important in WA development. However, organisations place security at different individual priorities. While some software applications are about safety and could lead to a loss of life, if not properly secured, others applications are business critical software which could lead to loss of data, clients, and revenue, if not properly secured (Ume & Chukwurah, 2012).

WAs are usually considered business-critical software. They are used to receive, store and process customer information. They are instrumental in the financial growth of a business as they promote sales and profit. Because they are widely used across different application domains (e.g. health, retail, ecommerce, etc.), there is a need for security to be integrated in all phases of the development cycle. These applications could pose a great risk if left vulnerable (Binuyo et al. 2015). In some organisations, top stakeholders do not understand the importance of ST in the SDLC because there is a misconception that the ST process could affect the timelines set for projects (Ogheneovo, 2014). The workload and pressure in some fast-paced domains, affects the adequate implementation of security strategies by development teams (Sowunmi & Misra, 2015).

Having carefully reviewed some of the existing challenges identified in software testing in the Nigerian IT industry, this study will further examine whether these factors are also applicable to ST adoption or identify new factors and challenges.

2.7. Overview of the technology acceptance model (TAM)

The Technology Acceptance Model (TAM) was originally developed in 1986 by Fred Davis (Davis, 1986). It was originally derived from the Theory of Reasoned Action (TRA) developed in 1967 by Martin Fishbein and Icek Ajzen (Fishbein & Ajzen, 1975). It hypothesises that a user's attitude towards using an information system (IS) determines its actual usage. It also maintains that attitude toward using is a function of two beliefs, namely, perceived use (PU) and perceived ease of use (PEOU), with PEOU having a causal effect on PU. The TAM also stipulates that according to the TRA model, design features of an IS are categorised as external variables which influence PU and PEOU (Davis, 1986). These design features are indicated as X1, X2 and X3 in Figure 2.5.

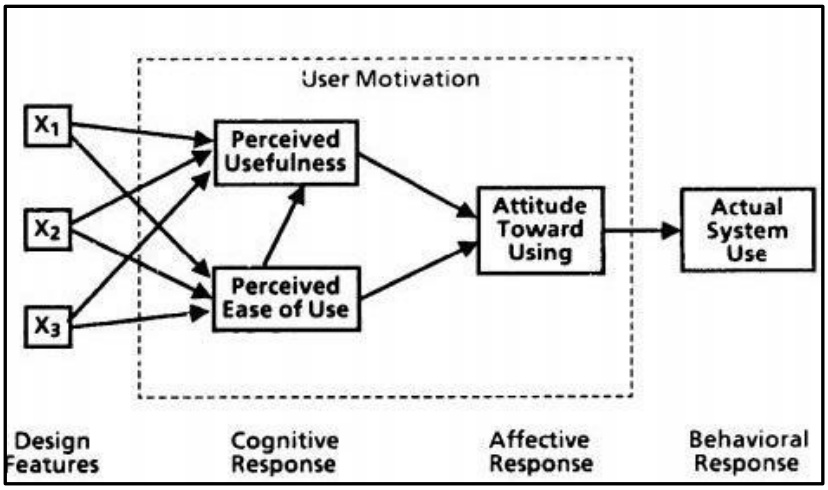

Figure 2.5. The original TAM developed in 1986 (Davis, 1986)

The TAM has evolved over the years as more research has been done to understand how its constructs predict acceptance of technology. The second iteration of the TAM (TAM2), was developed by Davis in 1989, which postulates that some external variables could influence the PU and PEOU to predict system use (Davis, 1989). It also introduces the behavioural intention to use (BIU) which mediates between ATT and USE. Figure 2.6 illustrates the TAM2 developed in 1989.

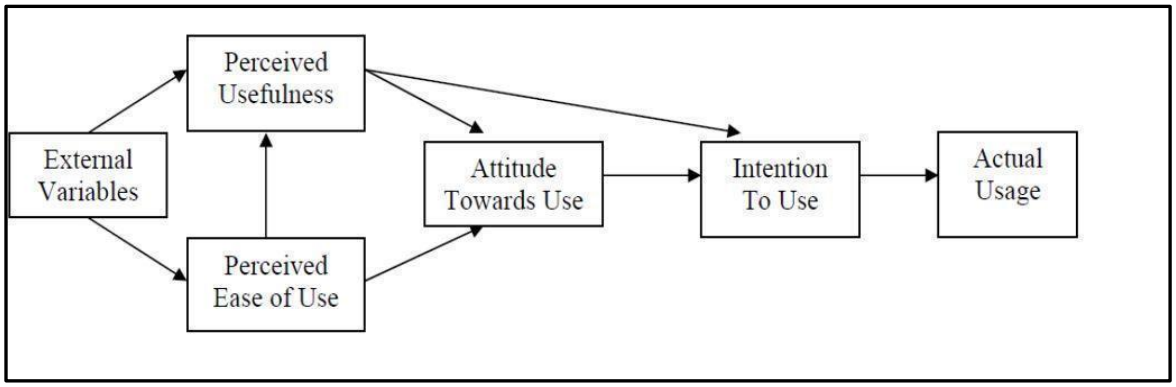

Figure 2.6. The first modified TAM (TAM2) developed by Fred Davis in 1989 (Davis, 1989)

In 1996, the final TAM (TAM3) was developed and the ATT construct was eliminated because some studies showed that users' intention to use a system was a better determinant of usage than attitude. Figure 2.7 shows the TAM3.

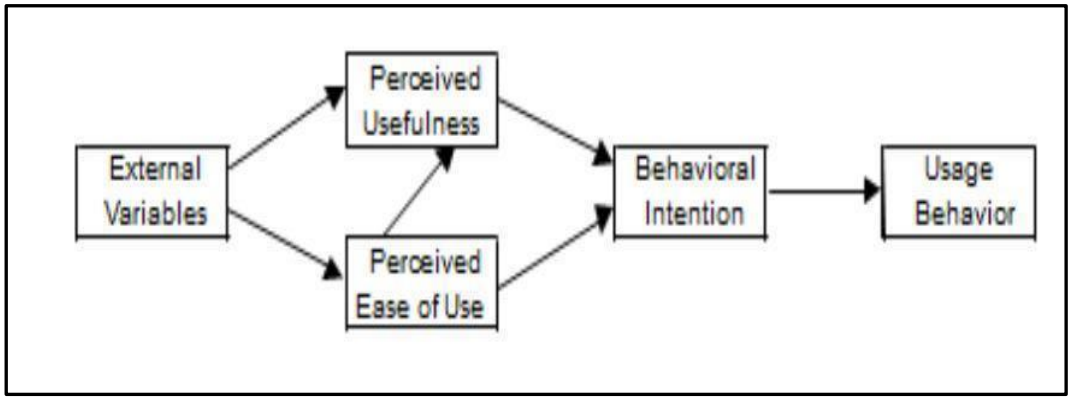

Figure 2.7: The final version of the TAM (TAM3) developed in 1996 (Venkatesh & Davis, 1996)

The TAM2 was adapted for this study because related research in the Nigerian IT industry revealed that willingness and attitude could influence the state of software quality practices in the Nigerian IT industry (Sowunmi & Misra, 2015). It was chosen and adequately modified to reflect the objectives of the study. While PU will help to understand developer's perception about the usefulness of ST, PEOU will also help to understand the ease of integrating ST for each team in the sample. The ATT construct was included to understand if it could influence USE in this sample as far as ST is concerned among development teams in Lagos.

From the reviewed literature of this study, some factors that affect quality software testing among Nigerian development teams include the complexity of the processes involved and the perceived value of the testing approaches by the teams (Sowunmi & Misra, 2015). These relate to PU and PEOU which are the two cognitive beliefs for understanding adoption of technology (Davis, 1989). There are several IS models that have been used to study user behavioural intention to adopt new technologies. A typical example is the Unified Theory of Technology of Acceptance and Use of Technology (UTAUT) (Venkatesh, Morris, Davis, & Davis, 2003). It holds that four constructs, performance expectancy, effort expectancy, social influence and facilitating conditions are direct determinants of usage behaviour. It has moderating factors of age, gender, experience, and voluntariness of use. However, some constructs of the UTAUT such as performance expectancy and social influence are not significant to this study because as per the reviewed literature, these may not have significant impact on software testing processes (Patil & Phil, 2014; Zia, 2015). Nevertheless, moderating factors of gender and experience are included as external variables in the conceptual framework (CF) for this study.

The TAM2 is simple and is a natural fit for this study. It has been used in similar studies of ST of WA (Erdogan, 2009; Huang, Tsai, Lin, Huang, Lee, & Kuo, 2005; Scarpino, 2010). Its external variables have been modified to understand adoption in IS domains such as electronic banking, education, web and cloud computing, within organisational settings (Kripanont, 2007; Lim & Ting, 2012).

Stewart (2013) applied the TAM3 to understand technology acceptance in some organisations. The PU and PEOU successfully explained the acceptance rate of the technology within an organisational setting (Stewart, 2013). In a similar study in 2015, the TAM was extended to explain how some social, cultural and organisational factors influence user behavioural intention to use the internet. In the study, the TAM2 was extended to include external variables from the theory of planned behaviour (TPB), the UTAUT and TRA (Abbasi, Tarhini, Hassouna, & Shah, 2015). In 2005, Johnson carried out an information security study using the TAM2 to understand an organisation's decision to invest in information security. In the study, the PU and PEOU together with some external variables explained user's decision to accept and invest in information security (Johnson, 2005). The TAM has been compared with many other theories such

as the Theory of Planned Behaviour (TPB) model and the TAM offers an empirical advantage over TPB. It is simpler, easier to use, and is a very powerful model to explain users' technology acceptance (Lee, Kozar, & Larsen, 2003).

2.8. The conceptual framework

The framework to be used to achieve the objectives of this study is adapted from the TAM2 developed in 1989. This is depicted in Figure 2.8.

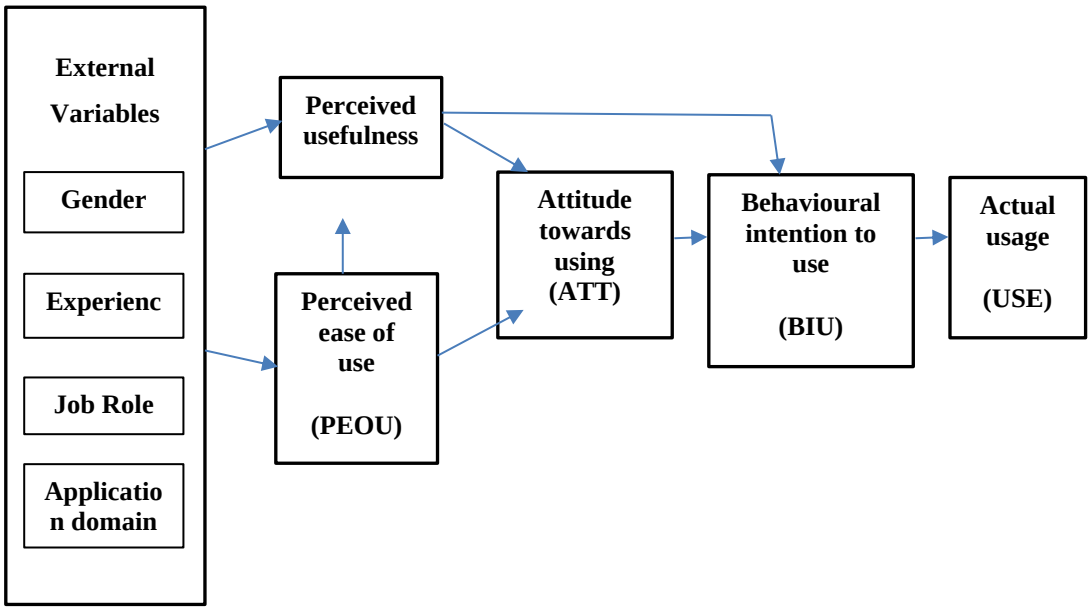

Figure 2.8. Conceptual framework (adapted from Davis, 1989)

The major constructs and external variables of the CF are defined as follows:

i. **Perceived usefulness (PU)**: This construct defines the extent to which a person believes that making use of an IS or technology could improve job performance. Independently, it predicts attitude towards using (ATT) and behavioural intention to use (BI) simultaneously. It is also a dependent construct because it is predicted by perceived ease of use (PEOU) (Erasmus et al., 2015). For this study, it would help to understand the perceived usefulness of ST in the development processes.

ii. **Perceived ease of use (PEOU)**: This defines the extent to which a user believes that the use of a specific IS or technology will be free of effort. An IT

application/process that is perceived to be easier to use will be easily accepted by a user. PEOU would help to understand ease of using and integrating ST in development how it impacts on the attitude of developers.

iii. **Attitude towards use (ATT)**: This construct is influenced by perceived usefulness (PU) and perceived ease of use (PEOU) of the technology. From the research, a high PU of a system or technological process could give a user a more positive attitude towards using the system (Johnson, 2005). In this study, it will be used to evaluate the user's attitude towards ST adoption and help discover ways of improving it.

iv. **Behavioural intention to use (BIU)**: This defines a user's readiness to accomplish a given behaviour. It is presumed to be an immediate precursor of USE and is determined by attitude and weighted for its importance in relation to the behaviour and population of interest (Venkatesh et al. 2003). This will help to identify possible ways of improving the challenges that affect ST adoption from the behavioural intentions of the developers.

v. **Actual system use (USE)**: This indicates the usage in the TAM model. The predictions of the model will help to explain the degree of usage and adoption of ST approaches in development (Erasmus et al. 2015).

vi. **External variables**: These variables are inputs that can affect the PU and the PEOU and predict adoption of technology. For this study, the chosen external variables will represent unique characteristics of the sample that could impact the adoption of ST (Venkatesh et al. 2003). Several external variables have been known to significantly influence technology adoption. In terms of this study, the researcher chose gender, experience, job role/job fit, and organisational type as variables based on previous IS security research.

    a) **Gender:** Gender differences have been found to have a significant impact on technology adoption in organisational settings. It influences ATT and BIU. Different factors motivate adoption of technology for men and women respectively and may be significant in this study (Venkatesh, Morris, & Ackerman, 2000).

b) **Experience**: This has been discovered to influence technology adoption because it has a direct impact on ease of use. The addition of experience to TAM is said to be very significant from previous studies (Szajna, 1996; Venkatesh et al. 2003).

c) **Job role**: This is a variable that can largely impact technology adoption. It affects the PU and PEOU as it is expected that users with more technical roles within SDTs would have a more positive attitude towards secure development because they understand its value in improving the quality of the applications they develop (Damanpour, 1991).

d) **Application domain**: The nature and sensitivity of the applications developed in a company can largely determine the extent of technology adoption and secure processes. Companies that develop products for high risk systems domains such as healthcare and finance will tend to adopt ST more because they value it as useful in improving their business processes (McGraw, 2006).

These variables will help serve as inputs that may influence PU and PEOU to predict the use of ST in the Lagos Nigerian IT industry.

2.9. Chapter summary

ST is a process that is critical in the SDLC of WAs. Many studies have identified different techniques and frameworks which are suitable for integrating ST into the SDLC of applications. While the SDLC approach adopted across organisations in the IT industry are different, each approach has defined ST strategies that could be applied to applications irrespective of application domain. Certain factors which have been known to affect the attitude and willingness of individual members of SDTs in software quality assurance in the Nigerian IT industry have also been identified from related research. The CF used in this study and the constructs measured in line with the research objectives have been defined. This present research investigates whether the constructs and variables of the CF have significant impacts in determining ST usage among WA developers in the Lagos Nigeria IT industry.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1. Introduction

This chapter describes the methods and approach used to achieve the objectives of this study. First, a comprehensive explanation of the research design which includes the data collection methods and instruments is presented. Second, a description of the participants for the study and the sampling techniques used to identify them in the sample is given. Third, the analysis techniques used are discussed in the concluding section of the chapter.

The present study is a business research project focused on providing answers to an organisational challenge. Sekaran and Bougie (2016) defines this type of research as "an organised and critical or objective investigation into a business problem" (Sekaran & Bougie, 2016). The outputs of business research are used in making decisions that will resolve the underlying problems in business and improve the current state of certain practices in the business.

3.2. Research design and paradigm

A research design is a strategy for investigating a research problem to find answers and achieve the desired objectives of the study. According to Sekaran and Bougie (2016), there are different rationales that form the motive of a business research and these determine the research design. As this present study seeks to understand a research interest that has not been widely researched, it is exploratory in nature. An exploratory research approach was undertaken in order to gain insight and understanding into a research context that has not been widely studied and assisted in gathering the needed information (Sekaran & Bougie, 2016). This present study also takes a descriptive approach because the outcomes of the study will help to describe and identify unique characteristics of the sample with respect to the research objectives (Olivier, 2009).

A mixed methods research design was chosen in order to carry out a detailed and comprehensive study to reveal the challenges of ST practices and adoption. Mixed methods research usually involves a blend of both qualitative and quantitative aspects of a study (Bryman, 2006). There are many reasons for mixing methods in research. A survey conducted by Bryman (2006) of some existing IS literature that utilised the mixed methods research approach identified the reasons why it was chosen. The study concluded that a mixed methods research approach is used to corroborate, complement, expand, explain, and provide credibility of the findings in a research study. Other reasons for using a mixed methods approach were to confirm, discover and provide diversity to a research context that has not been widely researched (Venkatesh, Brown, & Bala, 2013). In terms of the present study, a mixed methods research approach was chosen for the following reasons:

i. **Triangulation and validity:** Qualitative and quantitative research methods were combined and the findings from each research method were triangulated. The findings from the qualitative aspect of the study were used to validate the findings from the quantitative aspect of the research.

ii. **Completeness:** The study was exploratory and required careful investigation so that a detailed and comprehensive account of the research problems could be discovered and reported. Using a mixed methods approach ensured the completeness and richness of the study.

iii. **Expansion:** The four research questions (RQs) required different methods of inquiry and hence a mix of quantitative and qualitative methods was needed. The first and third research questions were answered using qualitative methods. The second research question was answered quantitatively. The fourth research question was answered after the findings of both qualitative and quantitative analyses were merged.

The mixed methods research design has been used in many IS studies because it adds rigor and gives insight to issues in practice which might not be achievable with a single approach (Jokonya, 2016). This research design has been used to study technology adoption in ecommerce (Pavlou & Fygenson, 2006) and in security related research

(Peng & Nunes, 2009).

Creswell and Clark (2007) highlight six major design strategies that are used in a mixed methods study. These are as follows:

i. Convergent parallel design;
ii. Explanatory sequential design;
iii. Exploratory sequential design;
iv. Embedded design;
v. Transformative design;
vi. Multiphase design.

Each of these research design types have different goals and purposes. While some involve mixing data sources at different stages of the data collection process, others interweave quantitative approaches in qualitative methods and *vice versa*. The interchange could be sequential or concurrent, depending on the objectives of the research.

For this present study, the overall research design was a convergent parallel design. This mixed methods design involves a simultaneous implementation of both quantitative and qualitative approaches. The data collection and analysis in each approach is done concurrently and independently of one another. The findings and results of each analysis are then merged and discussed in line with the objectives of the study (Creswell & Clark, 2007). Figure 3.1 depicts the design approach followed by this study.

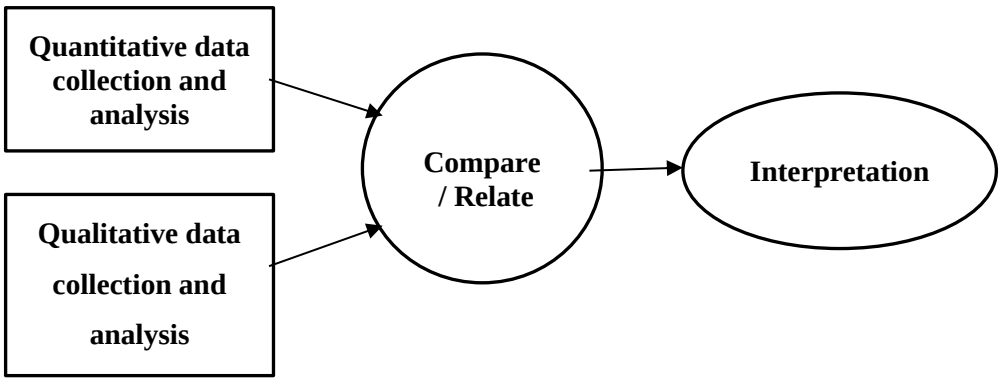

Figure 3.1. The design approach using mixed methods research (adapted from Creswell and Clark, 2007)

The convergent parallel design allows the researcher to combine the different strengths and weaknesses of the quantitative and qualitative research methods. It was chosen for this study because of the small sample size and the generalisation bias of the quantitative method was neutralised by the in-depth inquiries undertaken using the qualitative approach. It is suitable for research focused on teams and its efficiency is its major strength (Creswell & Clark, 2007).

The research paradigm is a model or organising structure or a philosophical stand or perspective in social science research (Feilzer, 2010; Creswell & Clark, 2007). For this present study, a pragmatist research paradigm was adopted. Pragmatism accepts that there are many unique and numerous realities that can be used to inquire, analyse and solve problems in the practical world (Johnson & Onwuegbuzie, 2004; Creswell & Clark, 2007).

3.3. Target population

The target population is the collection of people with similar characteristics for which the research study was directed towards. These include IT companies who build WAs in Lagos, Nigeria. The population spans across different application domains (e.g. banking/finance, retail/ecommerce, travel, health) (Soriyan, Mursu, Akinde, & Korpela, 2001) and sizes (e.g. SMEs) (Li et al. 2014).

3.4. Sampling strategies

Due to the exploratory nature of the research, non-probability sampling techniques were used to identify participating companies for this study. In terms of the data collection, purposive sampling was used to identify companies from the target population because the required information was only obtainable from target group companies who build WAs. Soriyan and Heeks (2004) have shown that IT companies in Lagos, Nigeria were in the domain of small, medium and large enterprises. Accordingly, purposive sampling was used to identify companies in each category of SMEs.

This sampling technique also allowed the researcher to select participants based on their expertise and familiarity with the problem being researched (Sekaran & Bougie, 2016). This technique was used so that the specific companies who specialise in WA development are selected to participate in the study so that they would be able to give the relevant feedback required for the study.

3.4.1. Sampling and sample size

Sampling is a process of selecting a part of a population to depict the entire population. A sample is a fragment selected from a population and the sample size is the total number of units in the sample for a study (Leedy & Ormrod, 2005). Relevant companies were selected using purposive sampling. Request letters for participation in the study were sent out to the identified companies. This letter contained a brief explanation of the study goals and objectives. Samples of the data collection instruments were also sent. Gatekeepers letter were obtained from each company prior to data collection.

For the quantitative data, simple random sampling was used to identify members in development teams of each participating company to administer questionnaires to. The findings from this technique can be generalised to the sample as each participant has an equal chance of being selected. It is believed that using the sampling method neutralises bias and provide credible results to achieve the objectives of the study (Creswell & Clark, 2007).

For the quantitative data, software developers, testers and project managers of development teams were administered questionnaires. For the qualitative data,

interviews were conducted with team leaders and heads of units. The target company types were IT companies in the small to medium (SME) scale range. These SMEs constitute about 70 % of the companies in the Nigerian IT industry (Soriyan & Heeks, 2004). Three companies were selected to participate based on purposive sample. Table 3.1 provides an estimate of the population and sample size.

Table 3.1. Sample population and distribution

S/N	Web development companies	Total population for teams in each company	Total questionnaires shared	Total questionnaires collected
1	Coy A	40	40	33
2	Coy B	35	35	29
3	Coy C	25	25	20
	Total	100	100	82

From Table 3.1, the questionnaires collected from the sample was eighty-two (82) and this was the sample size for the study. According to a study by Krejcie and Morgan (1970), an estimate sample size of eighty (80) out of a population of one hundred (100) is acceptable. Sekaran and Bougie (2016) also affirm that a sample size of eighty out of a population of one hundred is sufficient for a quantitative study. This high response rate was due to the pilot study carried out previously to enlighten participants on the research subject.

For the qualitative aspect of the study, eight respondents who were experts and heads in the development teams were selected to participate. The number of participants in the interviews was based on the availability of the participants. The required permission and approvals were acquired from each participating company before conducting the research. Ethical clearance was also granted from the University of KwaZulu Natal Research Office before data collection commenced.

3.5. Data collection methods

Data is acquired from participants in a research process and the questionnaires and in-depth interviews to be used for gathering data are the data collection instruments (Creswell & Clark, 2007). For this present study, questionnaires and in-depth interviews were used as instruments. This choice was based on the nature of the mixed methods research design. Questionnaires were used to gather data to achieve the quantitative objectives of the study and in-depth interviews with experts in the industry were used to gain the qualitative data.

3.5.1. Questionnaires

These are research instruments with a list of structured questions used to derive responses from a chosen sample (Leedy & Ormrod, 2005). The questionnaire consisted of twenty (20) close-ended questions that were developed after studying literature. The participants selected responses based on a set of predetermined response scales. The first section contained questions to obtain the descriptive statistics of the participants. The second section had questions developed using the Likert scale design in which the participants had to state the extent to which they agreed or disagreed with the statement being asked. (1-Strongly disagree, 2-disagree, 3-Neutral, 4-Agree, 5-Strongly agree).

Table 3.2. Number of Items of measurement per construct

Construct	Number of Items
PU	5
PEOU	4
ATT	4
BIU	4
USE	3
Total	20

The questionnaires were reviewed and validated by a statistician before final administration to participants after one month. The responses to the questionnaire were

recorded using numerical codes to uniquely identify each of its items. A known limitation in using questionnaires for data collection is that the data produced from the survey may lack depth (Kelley, Clark, Brown & Sitzia, 2003). This present study was however strengthened with the qualitative interviews to provide depth and substance.

3.5.2. Interviews

These were instruments of research used to explore and gain data from perceptions and experiences and allowed the researcher to generate understanding through dialogue (Creswell & Clark, 2007). A clear understanding about the current perception and practices about the ST process was derived from domain experts in the industry. The interview schedule consisted of open-ended questions. Participants gave their responses based on experiences in ST implementation within their SDTs. These interviews allowed for a comparison to be made between different participants because standardised questions were used to gain rich and relevant data for the study.

The interview questions were developed after a careful review of the existing literature to understand which aspects of the ST practice needed to be investigated. There were a set of three (3) questions each relating to the constructs PU, PEOU, ATT, BIU, and USE. There were fifteen (15) questions in all. The interviews sessions were scheduled and each session was digitally recorded. A total of eight (8) interviews were conducted. Three (3) from Company A, three (3) from Company B and two (2) from Company C. The recorded interviews were later transcribed for thematic analysis. The real names of the companies have been withheld in terms of field research ethics to ensure confidentiality.

A known challenge of interviews is the availability and accessibility of participants. Each participant was contacted prior to the interview sessions and appointments were booked based on their schedule (Leedy & Ormrod, 2005).

3.5.3. Interview schedule

The interview schedule was divided into two major sections. The first section concerned the demographics of the participants and also represents the external variables in the CF. The second section contained questions based on each of the constructs.

The PU questions were designed to probe how each of the teams viewed ST as useful and important to their SDTs. Questions were asked about the current processes that exist in the teams and how important the ST process was in the SDLC. The questions for the PEOU construct were focused on understanding how the ST process was being integrated into the SDLC. The questions were also designed to carefully understand the phases in the life cycle where ST was prioritised as this could determine the effort required in integration. Participants were prompted to mention some factors that influence the choice of the ST approaches. The ATT questions were designed to understand the disposition of team members towards adopting ST. It probed into understanding the willingness of teams to integrate ST despite the factors that were known to affect adoption and the possible complexity of the ST process. The BIU section had questions to identify the readiness and behavioural intent of the leads and heads of teams to use ST practices in the SDLC. Questions were asked to understand the preparations each team made towards ST per project and if there was adequate technical support for each team particularly from a managerial perspective. The USE section had questions that probed into how the ST approaches can be made more effective to improve its actual usage among each team. The participants were asked about compliance and regulatory policies that exist and how it influences the ST adoption.

3.5.4. Data collection

The data instruments were collected within a month of administering the questionnaires. This was due to the schedule of most of the participants in the different selected companies. The interviews were digitally recorded and later transcribed.

3.6. Data analysis

Data analysis in mixed methods research is both deductive and inductive and there are different approaches required for each type of data collected. While the qualitative data analysis uses a deductive approach to arrive at conclusions based on the qualitative research questions, the quantitative aspect uses an inductive approach. For the quantitative data, analysis was achieved using a statistical package for social sciences (SPSS).

For the qualitative data analysis, several activities were carried out for proper processing. The feedback data from the interviews were captured for easy storage and retrieval for analysis (Flick, 2014). Digital voice recordings were taken for each session. The recorded interviews were then transcribed to text before analysis. The transcription of data was done by the researcher and the text was cleaned and formatted in readiness for analysis. The thematic analysis of the data was done using Nvivo® software, which is used for sorting and analysing qualitative data. A careful read through the transcribed text was done to ensure that errors were avoided and data had been adequately captured. Pseudonyms were assigned to each participant for easy identification and confidentiality in terms of field research ethics. After a careful examination of the transcribed text, codes were developed from the text based on the RQs. Notable and frequently occurring words were identified using Nvivo® and some were developed into codes. The generated codes were arranged and labelled as child themes. Similar child themes were aggregated and grouped and parent themes emerged for each grouping. These parent themes then represented main themes that emerged to answer the research questions. The themes extracted were then reviewed to reveal the findings from the study.[1]

3.7. Pilot study

A pilot study was undertaken by conducting a mini survey between members of the Nigerian IT industry. The questionnaires were administered with the help of online surveys and printed surveys by hand delivery. The online survey link was shared among members of the Nigerian software testing board and local developer communities in the Lagos IT industry. The pilot survey was used to design interview questions for the qualitative feedback. The pilot results were not used in the final analysis.

3.8. Chapter summary

In this chapter, the methods and design approach used for the study were explained. The sampling strategies were described and the overall approach and methods used for

[1] These are further explained in chapter five.

the research data collection were then explained. The details of the data collection instruments were also given.

The next two chapters will present a comprehensive explanation of the analysis techniques used for the data collected as well as the various statistical tests that were applied to determine the results from the data. The frequencies and distribution of the results of the analysis will also be examined.

CHAPTER FOUR

QUANTITATIVE ANALYSIS

4.1. Introduction

This chapter contains details of the analysis of the quantitative data collected in this study. A description of the statistical tests used to generate the results will be carefully discussed and the results of the analysis and statistical tests given. Descriptive statistics will be presented followed by a set of inferential analysis.

4.2. Statistical tests applied in analysis

Several statistical tests were carried out at different instances to examine the relationships in the conceptual framework. The tests applied were as follows:

i. **Reliability analysis:** This was used to measure the validity and consistency of the variables that are defined in a scale using the Cronbach's alpha test (Sekaran & Bougie, 2016). It was applied to measure the validity of the constructs of the conceptual framework and their relevance for the study.

ii. **Cross-tabulation with chi-square test:** In statistics, crosstabulation tests determine trends and patterns of an occurrence within a sample (Sekaran & Bougie, 2016). In this study, it was used to examine if any pattern or significant relationships exist between the external variables and the security testing activities in each phase of the SDLC. Chi-square tests were further carried out on the cross-tabulations to determine the extent of significance between the two variables represented in the cross-tabulation. Fisher's exact test of independence was used when conditions of the Chi-square tests were not met.

iii. **One sample t-test:** This test was conducted to find out if the mean score generated for the items in the constructs were significantly different from the hypothesized scalar value (Sekaran & Bougie, 2016). It was conducted to understand any statistical differences between the hypothesized mean and the actual mean of each of the constructs.

iv. **Regression analysis:** This was used to assess the relationship between independent and dependent variables to understand any significant relationships between them (Creswell & Clark, 2007). Analysis of variance (ANOVA) test was applied to determine if the model was fit to examine the security testing challenges in the sample.

4.3. Descriptive statistics

Descriptive statistics describe the distribution and summaries about a sample. In this section they were used for the external variables in the conceptual framework. Frequencies and distribution summaries of the external variable, gender, age, years of experience and application domain are provided.

4.3.1. Descriptive statistics for external variables

The first section of the questionnaire was designed to collect demographic information from the participants. This section also indicates the external variables in the conceptual framework.

i. **Gender:** Data collected indicated that the frequency distribution of the gender variable disclosed that the majority (73%) of the participants were male, with only 27% being female (Figure 4.1). Figure 4.1 depicts the distribution.

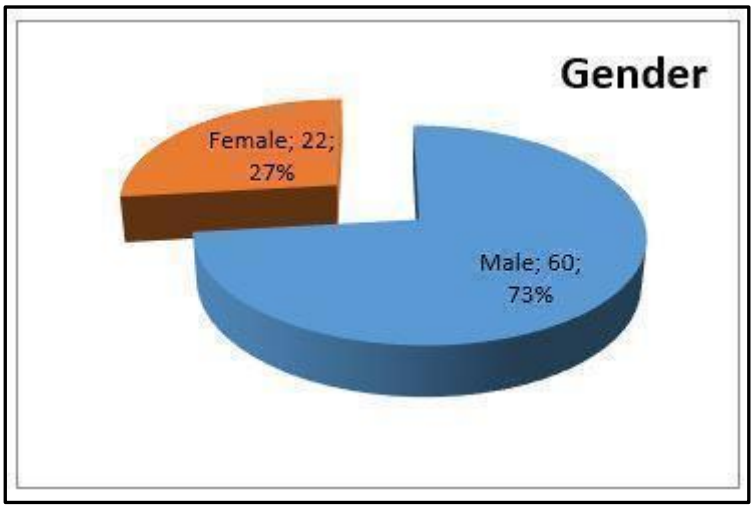

Figure 4.1. Frequency distribution for gender

This statistic suggests that a large percentage of SDTs in the Lagos IT industry are males. This pattern is closely related to that of Sowumi and Misra's 2015 study on software quality practises in Nigeria (Sowunmi & Misra, 2015). This indicates that there are more males than females in most SDTs in Lagos.

ii. **Years of experience:** The next variable measured concerned the years of experience each participant had in the IT industry. The participants were each asked to select from a set of options. Figure 4.2 provides a bar chart of the participants and the number of years of experience in their respective job roles in the industry.

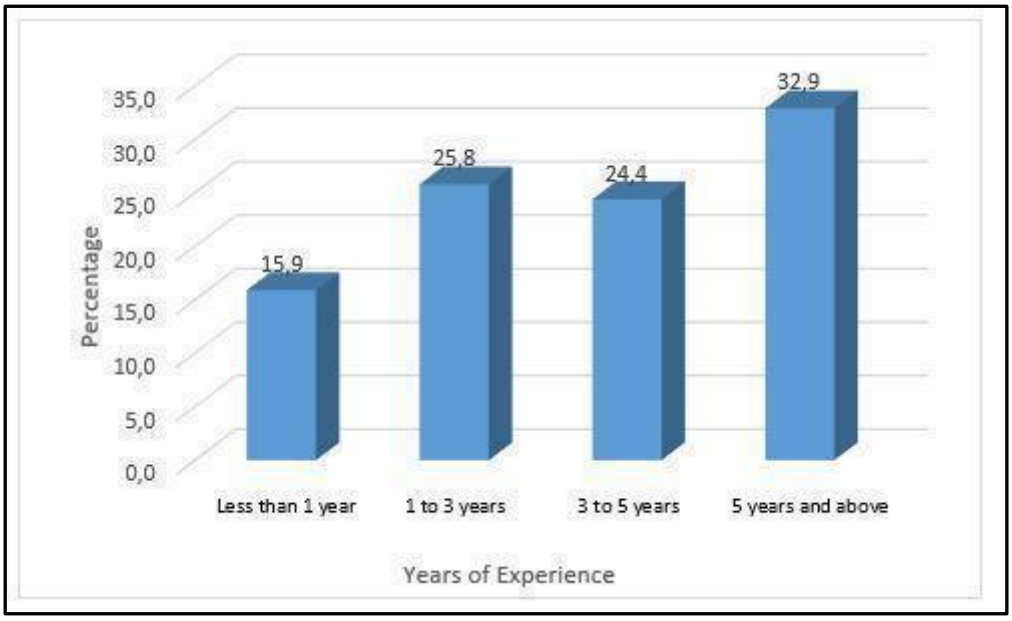

Figure 4.2. Frequency distribution for experience

From Figure 4.2, the highest percentage (32.9%) of participants have more than 5 years of experience building WAs. This percentage is followed closely by participants with 1 to 3 years' experience (25.8%) and participants with 3 to 5 years' experience (24.5%). A clear indication of this was that most participants had some experience and could give valid responses to the study based on their knowledge. The lowest percentage (15.9%) was indicated by participants with less than 1-year experience. This implies that there is a general inclusion of both skilled experts and new entry level graduates among SDTs in Lagos.

iii. **Job role:** The next variable measured was job role. There were three roles specified in this category. Software developers (also called software engineers), software testers (also called software quality assurance engineers) and project managers. These three roles were identified as they are the main roles notable in a typical development team in Nigeria (Sowunmi & Misra, 2015). The frequency distribution among these three-job roles is shown in terms of a bar graph in Figure 4.3.

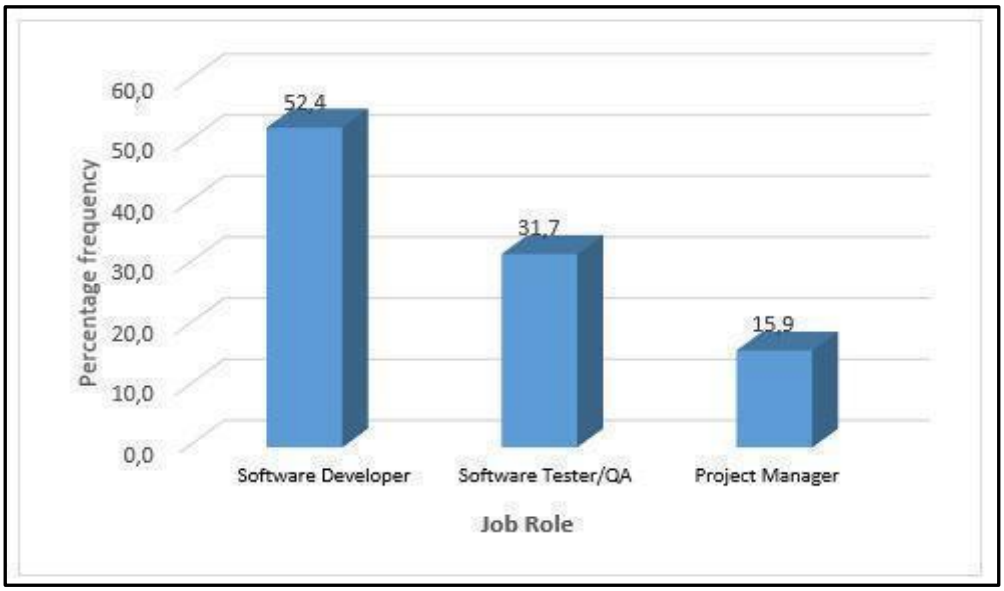

Figure 4.3. Frequency distribution for job role

The largest percentage (52.4%) of the participants indicated that they were software developers whose role was specifically to design and build the WA. The next category were the software testers who were 31.7% of the entire population. Their role in the SDLC is to test and verify that a WA is error-free and built to requirements, report bugs and discrepancies in development. The next job role were project managers which made up 15.9% of the population. Project managers oversee the entire SDLC and ensure proper planning and delivery is done.

iv. **Application domain:** The application domain variable was used to identify the domains that participants built WAs for. Figure 4.4 shows the frequency distribution for the five application domains specified in the questionnaire.

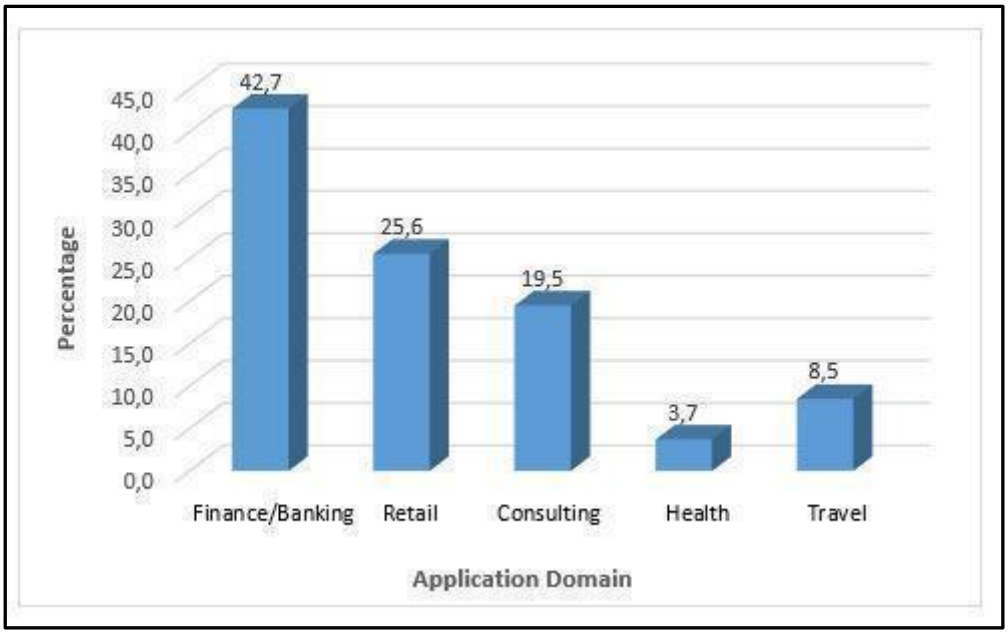

Figure 4.4. Frequency distribution for application domain

From the bar chart in Figure 4.4, it is clear that a large percentage (42.7%) of the participants actively built WAs for the finance/banking industry. This was not surprising as there have been many recent IT innovations in the finance and banking domains in Nigeria in recent years (Sowunmi & Misra, 2015). The ecommerce/retail application domain comes next with 25.6%. Participants who belong to the service/consulting domain were about 19.5% of the population while 3.7% of the participants built WAs for the healthcare and pharmaceutical industry. The remaining 8.5% built WAs for the travel/hospitality domain.

4.3.2. Security testing efforts in the phases of the SDLC

From the literature review it was discovered that there are different levels of effort required for ST implementation in each phase of the SDLC. Figure 4.5 indicates percentage distribution of ST activities applied in each phase of the SDLC as indicated by the research participants.

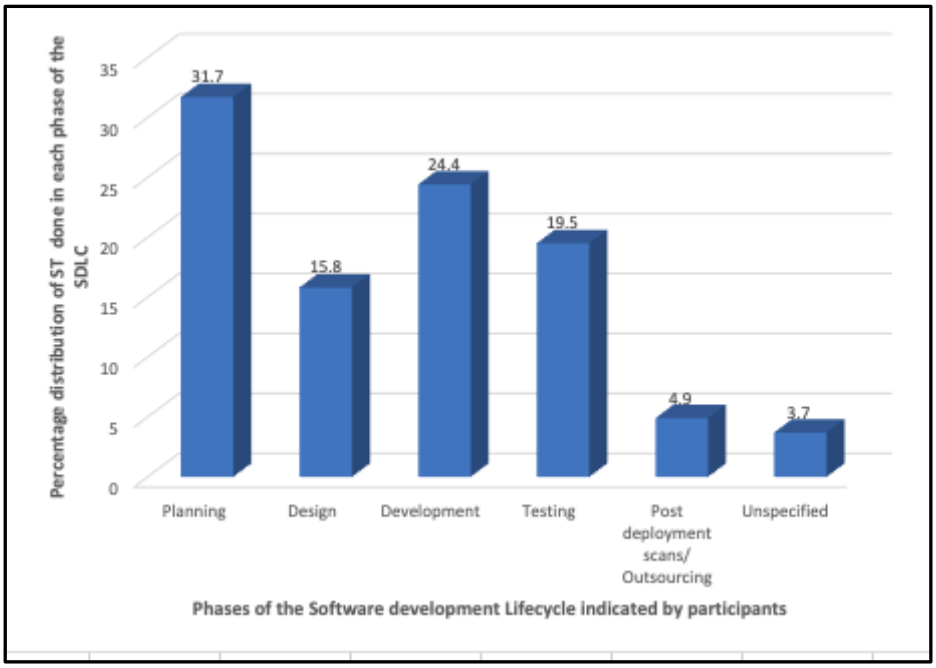

Figure 4.5. Frequency distribution for phases of the SDLC in which ST is applied

The ST activities in each phase of the SDLC have been described extensively in the literature review section. As shown in Figure 4.5, the majority of the participants (31.7%) indicated that they apply ST in the planning phase of the SDLC. This implies that most teams actually plan in advance for security before project implementation commences. From Figure 4.5 it can be seen that only 15.8% of the sample indicated that ST activities were done in the design phase of the SDLC. This may indicate that most teams do not transform their plan into a design phase. This could affect proper implementation of security requirements as conceptual designs guide proper implementation of ST. In the development phase of the SDLC, 24.4% of the participants indicated that ST approaches were intensified in this phase. This percentage also indicated that much effort goes into building in security in the application code. An implication of this is that security responsibilities will be placed more on software developers who build the code as they are involved in the development phase of the SDLC.

The testing phase of the SDLC involves ST techniques such as penetration testing and vulnerability assessment tests on the developed application using tools. A total of 19.5% of the participants indicated that ST efforts were intensified in this phase. In the post-deployment and maintenance phase, the ST activities involve post-deployment scans on the applications. ST during this phase could also be outsourced to external security

consultants and 4.9% of the sample selected this option. This low percentage could indicate that teams focus efforts towards the early phases of the SDLC. The remaining 3.7% of the participants did not respond to this question. Percentages in the earlier phases of the SDLC implied that teams understood that ST costs more towards the end of the SDLC and hence more activities were carried out in the early phases.

4.3.3 Impact of external variables on choice of security testing activities in the SDLC

To determine the associations between external variables and choice of security testing activities in the software development lifecycle (SDLC), cross-tabulation tests were carried between each external variable and the each of the security testing activities in the SDLC phases, planning, design, development, testing and post-deployment. Each of the ST activities in each phase were listed in the questionnaire and choices were made based on the options. The ST activities in each phase are itemised below.

- Planning: Developing security requirements and planning for it in the requirements gathering phase.
- Design: Building security into the design / model of the application before actual development.
- Development: Reliance on Inbuilt code technologies and implementing unit-based security frameworks during application development.
- Testing: Penetration testing and vulnerability assessments tests with tools during the testing phase after development
- Post-deployment: Post deployment scans and outsourcing to external security consultants.

It was observed that the external variables, gender, years of experience and application domain do not impact on the choice of ST activities in the SDLC. However, there was a significant relation observed between the external variable, job role and the testing approach carried out,. Table 4.2 below shows the results of the crosstabulation test carried out.

Table 4.2a: Crosstabulation tests between job role and security testing activities in each SDLC phase

		Planning	Design	Development	Testing	Post deployment	` Total
Software developer/ engineer	Count	14	7	17	2	2	42
	Expected Count	13.8	6.9	10.6	8.5	2.1	42.0
	% within Job role	33.3%	16.7%	40.5%*	4.8%	4.8%	100.0%
	Std. Residual	.0	.0	2.0	-2.2	.0	
Software tester/ Quality analyst	Count	6	0	3	13	2	24
	Expected Count	7.9	3.9	6.1	4.9	1.2	24.0
	% within Job role	25.0%	.0%	12.5%	54.2%*	8.3%	100.0%
	Std. Residual	-.7	-2.0	-1.2	3.7	.7	
Project manager	Count	6	6	0	1	0	13
	Expected Count	4.3	2.1	3.3	2.6	.7	13.0
	% within 3. Job role	46.2%	46.2%*	.0%	7.7%	.0%	100.0%
	Std. Residual	.8	2.6	-1.8	-1.0	-.8	
Total	Count	26	13	20	16	4	79
	Expected Count	26.0	13.0	20.0	16.0	4.0	79.0
	% within 3. Job role	32.9%	16.5%	25.3%	20.3%	5.1%	100.0%

Table 4.2b: Chi-square test of independence for job role and security in SDLC phase

	Value	df	Asymp. Sig. (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)	Point Probability
Pearson Chi-Square	41.521	8	.000	.000		
Likelihood Ratio	44.988	8	.000	.000		
Fisher's Exact Test	37.722			.000		
Linear-by-Linear Association	.128	1	.720	.727	.384	.044
N of Valid Cases	79					

From the table 4.2a and 4.2b above, it can be seen that there is a significant relationship between job role and security testing activities in the SDLC (Fisher's exact = 37.722, p<.0005). A significant number of software developers rely on inbuilt code security technologies and implement unit-based security tests in the development phase. Software testers rely on penetration tools and vulnerability scanners and implement this in the testing phase. Project managers on the other hand ensure that security models and designs are implemented in the design phase much earlier in the SDLC. This suggests that job role may influence the choice of ST activities carried out in the SDLC among software development teams in Lagos.

4.4. Inferential statistics

Inferential statistics is used to make inferences about a sample based on the data. In this section, Cronbach alpha test, one sample t-test, and ANOVA tests were applied.

4.4.1. Reliability analysis

As earlier discussed, reliability measures the inter-item consistency reliability between constructs in a model using the Cronbach's alpha test (Sekaran & Bougie, 2016). A reliability coefficient of 0.70 (alpha > 0.7) is considered a reliable and acceptable measure for social science research (Sekaran & Bougie, 2016). Table 4.3 shows the reliability statistics for each construct and the overall reliability of the constructs. The

individual and overall reliability indicates Cronbach's alpha values greater than 0.7, which are acceptable.

Table 4.3. Reliability analysis of conceptual framework constructs

Construct	Cronbach's alpha	Number of items
PU	0.749	5
PEOU	0.695	4
ATT	0.799	2
BIU	0.842	4
USE	0.807	3
Overall	0.778	19

From the results collected in Table 4.3, all the constructs except the PEOU have values greater than 0.7. The PEOU has a Cronbach's value of 0.695. Although this is less than 0.7, it is not far below. It can be accepted as reliable enough because it is very close. The results in the table indicate that each of the constructs of the conceptual framework are reliable to be measured.

4.4.2. One sample T-tests for conceptual model constructs

For the one sample t-tests, analysis was done to find the frequencies, mean and standard deviation (STD) of items in each construct. One sample t-tests were then applied to test for significant agreement or disagreement between the items. The average mean was 3. The mean value was used to compare agreement levels across the five constructs. Therefore, a mean greater than 3 indicated significant agreement, and a mean less than 3 indicated a significant disagreement. Values presented in the one sample t-test tables are shown below 0

i. N indicates number of items

ii. STD indicates standard deviation

iii. T indicates t-test value (t-test statistic) for a one sample t-test.

iv. Sig. (2-tailed): Also known as p-value. It is the probability that the hypothesis (t-test value) in each construct is significant. A p-value given as .000 is very small and is reported as $p< 0.001$. In the one sample t-test tables, results are statistically significant if p-value ≤ 0.001. Results are not statistically significant if p-value > 0.001

v. Degrees of freedom (df): It is the critical value of a *t*-distribution with $(n - 1)$ degrees of freedom

vi. Mean (M): the statistical mean value of the items in each construct

vii. Confidence interval: This is a range of likely values for the population mean. For this sample, 95% was the confidence interval.

Detailed tables showing the one sample t-test for each of the conceptual framework constructs are shown in sections 4.2.2.1 to 4.2.2.4.

4.4.2.1. *Analysis of PU items*

The perceived usefulness (PU) had five items of measurement. The items were focused on understanding how participants perceived software testing (ST) to be important. For all five items, the mean was greater than 3. The results of the one sample t-test carried for PU is shown in Table 4.4.

Table 4.4. One-sample test results for PU items

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
1. Security testing is important in the Development process (SDLC) of Web applications	27.04	81	.000	4.74	1.62	1.87
2. Using security testing practices prevents security vulnerabilities in the SDLC of Web applications	22.32	81	.000	4.51	1.38	1.65
3. Using security testing is useful in discovering application defects early in the SDLC	16.44	81	.000	4.39	1.22	1.56
4. Fixing application defects is easier when using security testing approaches	8.80	81	.000	4.02	.79	1.26
5. Using security testing approaches to fix application defects saves time	4.976	81	.000	3.59	.35	.82

From Table 4.4, the p-values for all the items in the PU construct are less than 0.005 and their means are above 3 and this indicates significance. This means that there is a significant agreement that ST is important in the development process of WAs ($M = 4.74$, $p < 0.001$), ST prevents security vulnerabilities in the development cycle of WAs ($M = 4.51$, $p < 0.001$), and ST is useful in discovering application defects early in the development cycle ($M = 4.39$, $p < 0.001$). It also further indicates that there is a

significant agreement that fixing application defects is easier using ST approaches (M = 4.02, p < 0.001) and using ST to fix application defects saves time (M = 3.59, p <0.001). Since there is a significant agreement for all the PU items and the overall mean is, the PU construct is significant to determine ST usage in the sample.

4.4.2.2. *Analysis of perceived ease of use (PEOU) items*

The PEOU consisted of four items designed to identify how participants perceived ST to be easy to integrate into the SDLC. Table 4.5 presents the results of the one-sample t-test on the PEOU items.

Table 4.5. One-sample test results for the PEOU items

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
1. Security testing techniques in the SDLC are simple and easy to learn	-.76	81	.449	2.93	-.26	.12
2. Security testing frameworks are easy to integrate into the SDLC of applications	3.95	81	.000	3.33	.16	.50
3. It is easy for me to become skilful at using security testing techniques in the SDLC	3.70	81	.000	3.35	.16	.54
4. Security testing practices are easy to adopt as a part of my responsibilities in the SDLC of applications	5.79	81	.000	3.52	.34	.70

In Table 4.5, the p-value for the first PEOU item is greater than the 0.001. (p>0.001). This implies that statistically, there is no significant agreement that security testing techniques are simple and easy to learn (M= 2.93, p >0.001). The other items of the PEOU however have mean values higher than 3 and p-values greater than 0.001. This implies that there is a significant agreement in the sample that ST frameworks are easy to integrate into the SDLC (M = 3.33, p < 0.001), that it is easy to become skilful at using ST techniques in the SDLC (M = 3.35, p < 0.001), and also that ST is easy to adopt in the SDLC (M = 3.52, p < 0.001).

4.4.2.3. *Analysis of Attitude (ATT) items*

Four items of measurement were used to determine Attitude (ATT). However, two negatively worded items had to be reverse coded so that same type of responses could be obtained across all items. The items focused on understanding what influenced the attitude of participants towards adopting ST in development. Table 4.6 captures the results of the one sample t-tests.

Table 4.6. One-sample test results for the Attitude (ATT) items

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
1. I like to use security testing in the SDLC because it helps to understand the application design better	4.54	81	.000	3.52	.29	.75
2. I like to use security practices because it helps my role in the SDLC and will help improve my team's processes and work output	7.04	81	.000	3.71	.51	.91

3. I prefer not to use security testing in the SDLC because it can delay my work deadlines	-1.11	80	.272	2.86	-.38	.11
4. I prefer not to not adopt security testing practices because it can make my work complex and cumbersome	-2.72	81	.008	2.71	-.51	-.08

From Table 4.6, the third item of the ATT had a p-value greater than 0.001. This statistically implies that there is no significant agreement that using ST in the SDLC can delay work deadlines ($M = 2.86$, $p > 0.001$). From the table however, there was a significant agreement that participants use ST because it helps them to understand the application design better ($M = 3.52$, $p < 0.001$) and also that using ST would help their roles and improve their team's processes ($M = 3.71$, $p < 0.001$). There was also a significant agreement that most participants prefer not to adopt ST approaches because it can make work complex and cumbersome ($M = 2.71$, $p < 0.001$).

4.4.2.4. *Analysis of behavioural intention to use (BIU) items*

There were four items in the BIU construct. The questions were posed to understand the intention and willingness of participants towards using ST. The results of the one-sample t-tests are shown in Table 4.7.

Table 4.7. One-sample test results for the behavioural intention (BIU) Items

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
1. I will likely apply security testing in building applications in order to adhere to ethical and good coding practices	9.37	81	.000	3.89	.70	1.08
2. I plan to use security testing in future in all developments because it is useful to my career	10.52	81	.000	4.00	.81	1.19
3. I will use security testing in the SDLC when it is critical to the nature of the application	12.93	81	.000	4.15	.97	1.32
4. With the necessary training and support, I intend to use adequate security testing approaches in the required stages of the SDLC	17.92	81	.000	4.38	1.23	1.53

From Table 4.7, all the items had a mean greater than 3 and a p-value less than 0.001. This statistically implies that there is a significant agreement that ST would likely be applied in application development in order to apply to ethical and good coding practises ($M = 3.89$, $p < 0.001$) and that SDTs plan to use ST in future developments because it is useful to their careers ($M = 4.00$ $p < 0.001$). There was also a significant agreement that ST will be used when it is critical to the nature of the application ($M = 4.15$, $p < 0.001$). There was also a significant agreement that with the needed training

and support, participants would use ST in the required phases through the development life cycle (M = 4.38, p < 0.001). Since all the items indicated significance when compared to mean, BIU is a significant construct.

4.4.2.5. *Analysis of actual Usage (USE) items*

The USE construct had three items of measurement. Questions were asked to determine whether participants used ST in the phases of application development and if each there were activities to encourage awareness and adoption. The participants were also asked questions to ascertain if ethical and regulatory issues influenced the adoption of ST in development. One sample t-tests were applied to the data and the results are given in Table 4.8.

Table 4.8. One-sample test results of the Actual Usage items

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
1. I adopt security testing in all stages of application development	2.80	81	.006	3.28	.08	.48
2. I regularly engage in activities to encourage security testing awareness and learning in my organisation	3.34	81	.001	3.38	.15	.60
3. I frequently apply and use regulatory policies that exist to support security testing practices in the SDLC	.94	77	.349	3.10	-.11	.32

For the first two items of the USE construct, the p-value was less than 0.001($p<0.001$). This means that there was no significant agreement that ST was adopted in the phases of application development ($M=3.28$, $p > 0.001$). Table 4.8 also shows that there was no significant agreement that participants regularly engage I activities to encourage ST awareness and learning in their organisations ($M = 3.38$, $p = 0.001$). The table also shows that statistically, participants do not significantly agree that they frequently apply regulatory policies that support ST practises in the SDLC ($M = 3.10$, $p > 0.001$).

4.4.3. One sample t-tests to determine overall significance of CF constructs

For all the constructs of the framework, one sample t tests were applied to determine the overall significance of each construct. Figure 4.6 represents the degree of significance measured for PU, PEOU, ATT, BIU and USE.

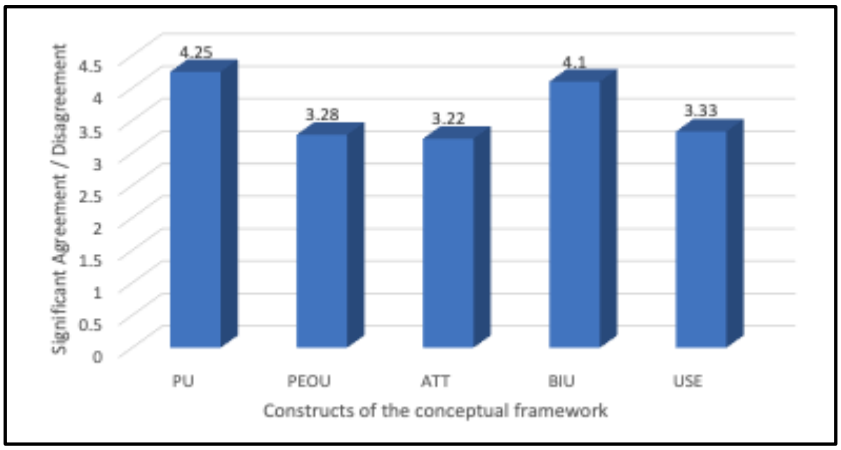

Figure 4.6. Significance of conceptual model constructs to predict adoption of security testing

As shown in Figure 4.6, each construct had a mean greater than 3, and this means a general significance was observed. This indicates that measuring each construct was significant to predict ST adoption in the sample. The results of the one-sample tests for the conceptual framework constructs are given in Table 4.9.

Table 4.9. One-sample t test results of the conceptual framework constructs

One-Sample Test						
	Test Value = 3					
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean	Lower	Upper
PU	19.03	81	.000	4.25	1.12	1.38
PEOU	4.29	81	.000	4.28	.15	.42
ATT	2.09	81	.039	3.22	.01	.43
BIU	15.02	81	.000	4.10	.96	1.25
USE	3.37	81	.001	3.33	.13	.52

As recorded in Table 4.9, PU, PEOU, BIU and USE indicate statistical significance with varying levels of significant agreement ($p < 0.001$). However, the ATT is not statistically significant based on the results in table 4.7. ($p > 0.001$)

Since the PU construct was the most significant with a mean of 4.25, it implies that perceived usefulness is very significant in determining ST usage. Table 4.9 also indicates that BIU with a mean of 4.10 is very significant. This might be due to the high mean of PU which directly impacts BIU. The PEOU and the USE also have some degree of significance in the conceptual framework. In the next section, the causal relationships between each of these constructs will be analysed to further explain the relationships between them.

4.4.4. Regression analysis

To test the causal relationships between the constructs, ANOVA tests were applied. ANOVA is a type of statistical test conducted to compare and understand the differences among multiple groups (Sekaran & Bougie, 2016). This test was applied to determine if there was any significant impact of the predictor variable (construct) and the dependent variable (construct) in each examined relationship. Below is a short explanation of values indicated in the model summary and ANOVA tables.

i. Multiple correlation (R): this is the correlation between predicted and observed values. R square (R^2) indicates square multiple correlation coefficient. It is the proportion of variance in the dependent variable which can be explained by the independent variable (Pallant, 2016).

ii. Adjusted R square: This is an adjustment of the R-squared that penalises the addition of extraneous predictors to a model (Pallant, 2016).

iii. Std. Error of the Estimate: this is the root mean square error (Pallant, 2016).

iv. Sum of squares: these are the sum of squares related to the three sources of variance which are total, residual and model (Pallant, 2016).

v. Degrees of freedom (df): These are the degrees of freedom associated with the sources of variance.

vi. Mean square: These are the sum of squares divided by their respective degrees of freedom (df). (Pallant, 2016).

vii. F statistic (F): This is the mean square (regression) divided by the mean square (residual) (Pallant, 2016).

viii. Significance (Sig): This is the p-value associated with the F-statistic. It indicates statistical significance (Pallant, 2016). For all the ANOVA tables, the superscript 'a' indicates the predictor and superscript 'b' indicates the dependent variable.

4.4.4.1. *Relationship between PU and PEOU*

Table 4. 10. Model summary for PU and PEOU

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.398[a]	.159	.148	.54946

a. Predictors: (Constant), PEOU

b. Dependent variable: PU

Table 4.11. ANOVA table for PU and PEOU

Model	Sum of Squares	df	Mean square	F	Sig.
Regression	4.553	1	4.553	15.080	.000[a]
Residual	24.152	80	0.302		
Total	28.705	81			

From Tables 4.10 and 4.11, PEOU accounts for 15.9% (R^2= .159) of the variance in PU, F (1.80) = 15.080, p <0.001. This implies that PEOU is a predictor of PU.

4.4.4.2. *The relationship between PU, PEOU and ATT*

Table 4.12. Model summary for PU, PEOU and ATT

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.115[a]	.013	-.012	.95524

a. Predictors: (Constant): PU, PEOU

b. Dependent variable: ATT

Table 4.13. ANOVA table for PU, PEOU and ATT constructs

Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	0.962	2	0.481	0.527	0.592[a]
Residual	72.087	79	0.912		
Total	73.049	81			

As per Table 4.13, F (2, 79) = 0.527, p<.001. Since the R^2= .013, it is insignificant. This implies that neither PEOU nor PU is a significant predictor of attitude. However, to further examine the PU and PEOU impact on ATT, the coefficients table for PU, PEOU and ATT are given in Table 4.13.

Table 4.14. Coefficients table between PU, PEOU and ATT

Coefficients*								
Model		Un-standardised Coefficients		Standardised Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	2.979	.824		3.616	.001		
	PEOU	-.155	.193	-.098	-.803	.425	.841	1.188
	PU	.176	.194	.111	.907	.367	.841	1.188

* Dependent Variable: ATT

The t-statistic vales for the PU and the PEOU show that when considering this specific sample, an increase in PU results in an increase in ATT, while an increase in PEOU results in a decrease in ATT.

4.4.4.3. *Relationship between PU and BIU*

Table 4.15. Model summary for PU and BIU

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.353[a]	.124	.113	.62659

a. Predictors: (Constant): PU

b. Dependent variable: BIU

Table 4.16. Analysis of Variance for PU and BIU constructs

Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	4.460	1	4.460	11.360	0.001[a]
Residual	31.409	80	.393		
Total	35.869	81			

From the results given in Table 4.15 and Table 4.16, BIU accounts for 12.4% (R^2 = .124) of the variance in PU, F (1, 80) = 4.460, p<.001. Hence PU is a significant predictor of BIU.

4.4.4.4. *Relationship between ATT and BIU*

Table 4.17. Model summary for ATT and BIU

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.159[a]	.025	.013	.66109

a. Predictors: (Constant): PU

b. Dependent variable: BIU

Table 4.18. Analysis of Variance for ATT and BIU constructs

Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	.906	1	.906	2.072	.154[a]
Residual	34.963	80	.437		
Total	35.869	81			

From the results given in Table 4.17 and Table 4.18, ATT accounts for 2.5% (R^2 = .025) of the variance in PU, F (1, 80) = 2.072, p<.001. Although not as significant, ATT still has an impact on BIU.

4.4.4.5. *Relationship between BIU and USE*

Table 4.19. Model summary for BIU and USE

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.250[a]	.063	.051	.86333

a. Predictors: (Constant): BIU

b. Dependent variable: USE

Table 4.20 Analysis of Variance for PU and BIU constructs

Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	3.982	1	3.982	5.343	.023[a]
Residual	59.628	80	.745		

From the results given in Table 4.19 and 4.20, BIU accounts for 6.3% (R^2 = .063) of the variance in PU, ($F(1, 80) = 5.343$, $p<.001$). This implies that BIU is a significant predictor of USE. A detailed explanation about the significant relationships between the constructs are presented in the discussion chapter.

4.5. Chapter summary

In this chapter, detailed results of the outputs of the quantitative analysis were given. The descriptive statistics displaying the frequency distribution of the characteristics of the sample were also provided. The results of the various statistical tests carried out on the constructs in the conceptual framework and the significant relationships between were also examined. Further discussion regarding the findings from this chapter will be presented in chapter six.

CHAPTER FIVE

ANALYSIS OF QUALITATIVE DATA

5.1. Introduction

The details of the analysis performed on the qualitative data collected is discussed in this chapter. Details about how the thematic analysis was done and how the data was coded will also be explained.

5.2. Thematic analysis

There are several types of techniques to analyse text data in qualitative research. Content analysis involves investigating texts and giving results of analysis in numerical descriptions or descriptive images. Thematic analysis is a type of content analysis which involves a critical assessment of the qualitative aspects of the data rather than the numerical indications (Vaismoradi, Turunen, & Bondas, 2013). It can be used as a standalone method and as a tool along with different methods. It is independent of theory and gives the researcher a form of theoretical freedom, while providing a rich, comprehensive and detailed account of data (Braun & Clarke, 2006). A major limitation of thematic analysis in research is the lack of a specified guideline or framework for conducting it appropriately (Aronson, 1995). However, several researchers have over the years presented a standardised way to process thematic analysis in qualitative studies (Muir-Cochrane & Fereday, 2006; Thomas & Harden, 2008).

5.2.1. Coding the data

According to Saldana (2015), coding involves identifying words or phrases to symbolically assign an aggregate characteristic for a portion of data. In the first coding cycle of the analysis, an exploratory labelling of notable ideas and patterns observed was performed in line with the research objectives. In the second coding cycle, the identified labels were grouped into categories which emerged into themes. These themes can be grouped to develop a framework or theory (Saldana 2015). In this study, the data relevant for analysis were identified and labelled from the data set. In Nvivo®,

this type of coding method is referred to as structural coding. This process helps a researcher to familiarise her/himself with the participants' perceptions and language (Guest & MacQueen, 2008; Morse, 1994).

Second cycle coding methods were used to re-organise and analyse data coded in the first cycle. Similar categories were grouped into coherent themes from the data. Pattern coding method was used to identify similarly coded data based on attributes and meaning.

5.2.2. Biographical information of the interview participants

Three companies were selected to participate in the study, and from each company participants with different role functions were interviewed. Each participant will be identified and referred to in the study according to their job function followed by the company name, e.g. DevMan A refers to the development manager of company A.

i. DevMan - Development manager.
ii. ProMan- Project manager.
iii. TestMan - Test manager.

Table 5.1. Profile of interview participants

	Company	Role	Gender	Experience	Application Domain	Pseudonym
1	Coy A	Software development manager	Male	3 to 5 years	Consulting	DevMan A
2	Coy A	Project manager	Male	1 to 3 years	Consulting	ProMan A1
3	Coy A	Project manager	Male	5 years and above	Consulting	ProMan A2
4	Coy B	Software development manager	Male	3 to 5 years	Health	DevMan B
5	Coy B	Software test manager	Male	5 years and above	Travel	TestMan B
6	Coy C	Software development manager	Male	5 years and above	Ecommerce/ Payments	DevMan C
7	Coy C	Software test manager	Male	5 years and above	Payments	TestMan C
8	Coy C	Project manager	Male	3-5 years	Payments	ProMan C

5.3. Theme development using NVIVO®

The analysis of the transcribed interview data was carried out using the Nvivo® software developed by Qualitrix. The interview questions are given in Appendix B. After a careful read-through of the text, codes were developed using the earlier stated coding methods of analysis. The relevant information identified through the transcribed text

were grouped together into themes. Similar themes were then categorised into larger parent themes, as indicated in the study. Table 5.2 indicates the themes and categories that emerged from the analysis.

Table 5.2. Valid themes identified from the constructs of the conceptual framework

Constructs	Themes
PU	1. Security testing approaches used among teams 2. Importance of ST to application domain 3. Degree of technical expertise to support ST
PEOU	1. SDLC approach adopted by team 2. SDLC phase in which ST is most intensified 3. Factors that affect ST used in the SDLC
ATT	1. Willingness to adopt ST 2. Impact of limiting factors on ST 3. Complexity of ST and attitude
BIU	1. Behavioural disposition towards adopting ST 2. Preparation and planning towards ST implementation 3. Investment towards ST by teams
USE	1. Use of ST among teams 2. Ways to improve ST usage among teams 3. Impact of ethical and compliance policies on ST adoption

Table 5.2 provides the emerged themes based on the CF constructs. From the reviewed literature, the choice of ST approach can impact the quality of WAs developed by teams in each SDT and consequently their job performance. Also, the perception of SDTs about the importance of ST and the degree of technical expertise available to support ST in each team can impact job performance. The PEOU themes relate to the extent to which a user believes ST will be free from effort. As discussed in chapter two (Figure 2.2), more effort is required in the earlier phases of the SDLC. Hence, the choice of SDLC approach and the phase of the SDLC can determine the degree of effort required

in implementing ST. The factors that affect ST could also impact the effort required in ST adoption and this theme directly answers the third research question of this study.

For the themes in ATT, the willingness to adopt ST is a theme that emerged and this could be an indication of a positive attitude toward ST adoption. The impact of limiting factors could also affect the attitude of SDTs towards ST. In addition, as observed in the research participants' feedback, the complexity of the ST process has an impact on the ATT of teams in ST adoption. For BIU, the preparation and planning made by SDTs could signal a general intention to use ST. The disposition and the degree of investment made towards ST also gives a hint towards intention to adopt ST. For the construct, USE, the themes that emerged provides an indication of the actual use of ST among teams. Ways to further improve ST adoption emerged as a theme based on user experience of the participants. The impact of ethical and compliance factors which emerged will also be used as a supporting theme for factors that affect ST adoption.

Only themes from the PU, PEOU and USE were found to directly explain the research objectives for this study and they are listed in Table 5.3.

Table 5.3. Themes related to the research objectives for the study

Construct	Themes
PU	Security testing (ST) approaches used among teams ● Basic security implementation following security principles ● Risk based security testing ● Security Testing through the SDLC phases
PEOU	Factors that affect ST adoption in the SDLC ● Human resource factors ● Project constraints ● Ethical and Compliance factors
USE	Ways to Improve ST adoption among teams

The themes directly related to the research objectives and the way they emerged from the transcribed data are discussed in the sections which follow.

5.3.1. PU themes related to research objectives

As indicated in Table 5.2, three themes emerged from the PU construct. They were as follows:

i. Basic security implementation following security principles.
ii. Risk based security testing.
iii. ST through the SDLC phases.

Figure 5.1 illustrates how these themes emerged.

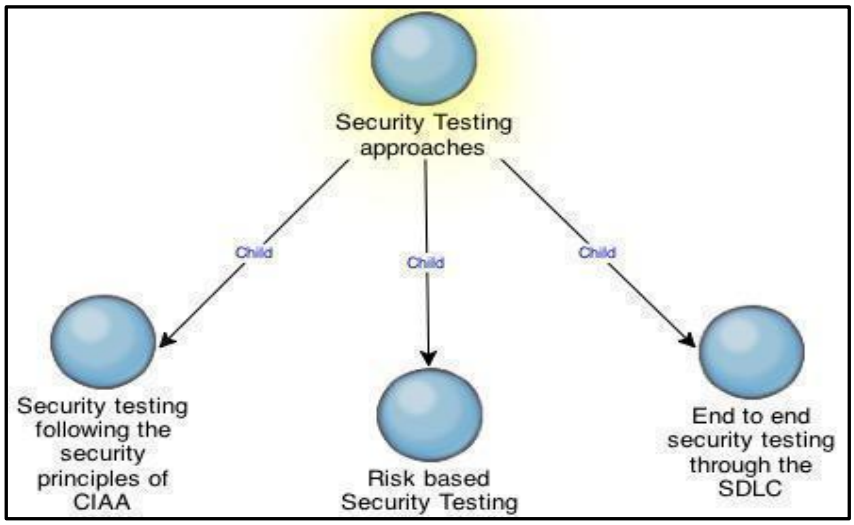

Figure 5.1. Security testing approaches used among Lagos web developers

Further discussion concerning these themes will be given in the next chapter.

5.3.1.1. *Basic security testing implementation following the known security principles*

This ST approach involves implementing security in development in specific areas of the application based on the security principles. According to one of the research participants:

> *We focus on the security principles of CIA (Confidentiality, Integrity and Availability) to guide our security testing implementation. Specific features like the authentication and access control features that are*

critical are properly developed and tested to ensure that the CIA goals are met (DevMan B).

Tittle et al. (2006) explains the importance of the top three security goals in guiding security testing implementation. Typically known as CIA goals, Confidentiality (C) is a security goal that indicates the level of security assurance in an application. It also provides a measure of protection for the application data against unauthorised access. Integrity (I) ascertains that the data the application processes is unaltered. Availability (A) is used to verify that authorised people can access resources and data in an application at all times when a request is made (Tittle et al. 2006).

In Coy C, the project manager explained that their team applies other security goals such as authorisation and non-repudiation to guide their ST process:

Usually in the SDLC while planning, we develop security requirements. As such, we want to target the system following the goals of security CIAA. We focus on confidentiality, availability and non-repudiation (ProMan C).

These security goals are suitable for guiding the approach to ST because WAs typically have critical features such as authentication and access control. These features serve as input and output points for data flow into the application and are prone to exploitation by malicious hackers. In Coy B, they also adopted the security goals in implementing ST:

Most times we focus on the security principles of CIAA (Authentication and confidentiality) to guide our security testing implementation because we are not aware of known ST frameworks that are specific to our applications or how to implement (DevMan B).

5.3.1.2. *Risk-based Security testing*

A second approach noted from the themes is the risk-based ST approach. Here, security is prioritised in the critical high-risk areas in the development process. The project

manager in company C explained that this approach was also used in some instances and on some projects:

> *In some project life cycle, risk assessment is done to determine the security testing approach. So, this helps to determine how much effort we have to put in security with respect to the available time (ProMan C).*

Research participants from Coy A and Coy C described some techniques used alongside the risk-based approach to ease security implementation in the development life cycle:

> *In the design phase, we do some form of analysis and model attack trees especially for critical projects* (DevMan C).

> *Before the development phase is completed, code review is done to check if there are no obvious points of security vulnerability in the code or issues. This is carefully done by managers and heads of teams* (DevMan A).

> *During the phase of development, we have code review sessions where senior engineers go through the code and identify vulnerable points* (DevMan C).

5.3.1.3. *Security testing through the entire software development life cycle*

The third approach to ST implementation among the teams involved a complete end-to-end program through the development cycle. A research participant in Coy C provided a careful description:

> *After the entire team identifies the requirements and designs the model for the application, the developer working on the application works and implement security features in the application build. In the second stage, the application developed is pushed to the staging (testing) environment and then the quality assurance team tests again in this staging*

environment. The final stage, often times there is a periodic security sweep done in production level to ensure that the application is safe even after post-deployment (DevMan C).

This ST approach is comprehensive and has specified ST activities and techniques for each phase of the development life cycle.

5.3.2. PEOU themes related to research objectives

After the thematic analysis of the interview data, three categories emerged for the different factors that affect ST adoption. These were then grouped into a major parent theme that emerged to indicate the factors that affect ST adoption. Figure 5.2 illustrates how these factors were grouped.

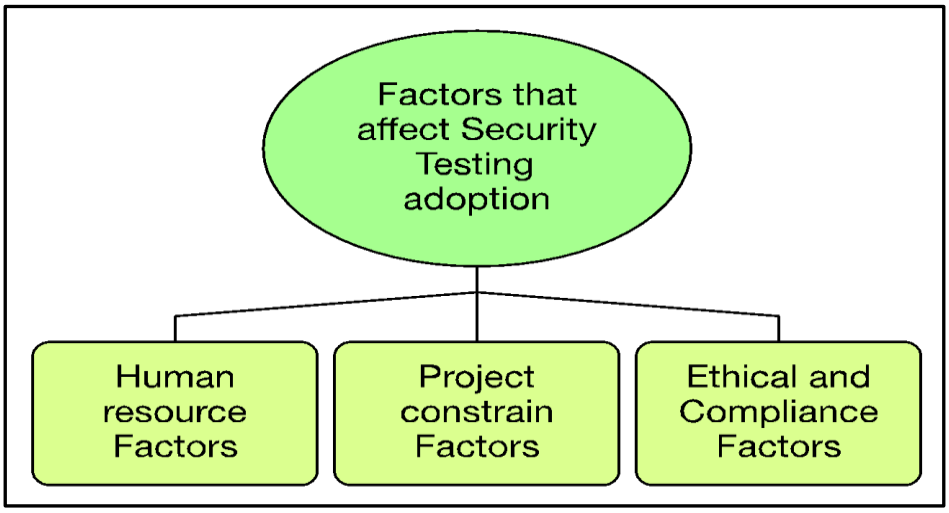

Figure 5.2. Factors that affect security testing adoption

Each of these three themes also had child themes that emerged. Figure 5.3 illustrates the human resource factor theme and its child themes.

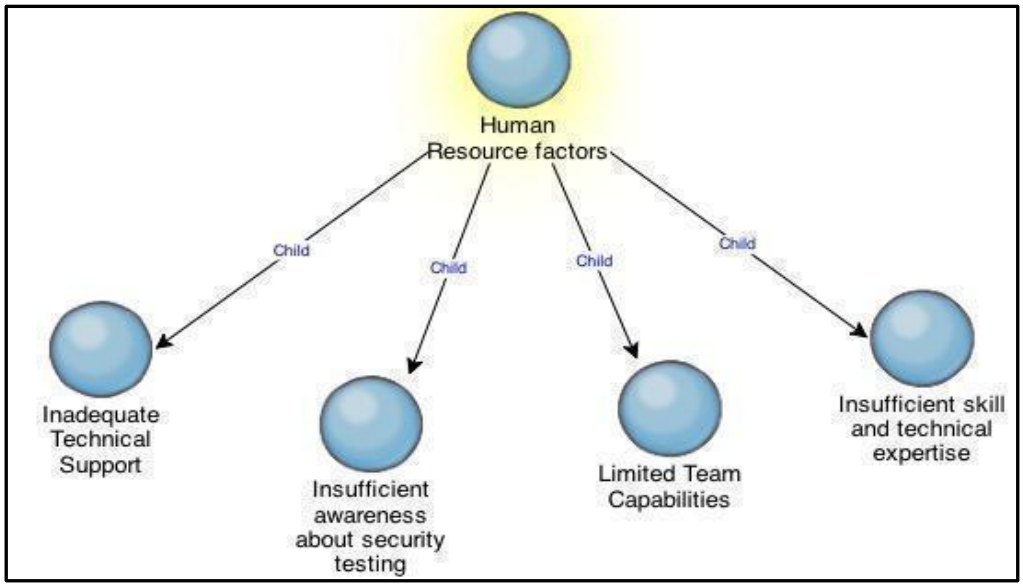

Figure 5.3. Human resources factors

5.3.2.1. *Human resource factors*

Four child themes emerged from this theme. These were as follows:

i. **Insufficient awareness:** This was a factor stated by research participants from Coy B:

Lack of awareness largely affects the security testing process (TestMan B).

We are not aware of known ST frameworks that are specific to our applications or how to implement (DevMan B).

ii. **Limited team capabilities:** It was also observed that team size and capability are factors that affect the effectiveness of ST implementation. In this respect, research participant ProMan A2 provided a detailed explanation:

The nature of the working environment and the size of the team affects the proper implementation of security testing approaches and frameworks. The size of the team on a particular project is always small compared to the workload. Hence the team members prioritize on roles

and oftentimes, security is affected because of the work pressure (ProMan A2).

iii. **Insufficient technical skills and expertise:** This was another factor noted by one of the research participants:

Oftentimes we are constrained in finding the needed expertise in the industry so we just spend more time to train people on the job (TestMan B).

iv. **Inadequate technical support:** Inadequate technical support was also stated as a factor that limits ST adoption by one participant:

There are limiting factors to be considered like training and technical support that affects the attitude of team members towards security testing (ProMan A1).

Figure 5.4 depicts the project constraint factors theme that emerged and its child themes.

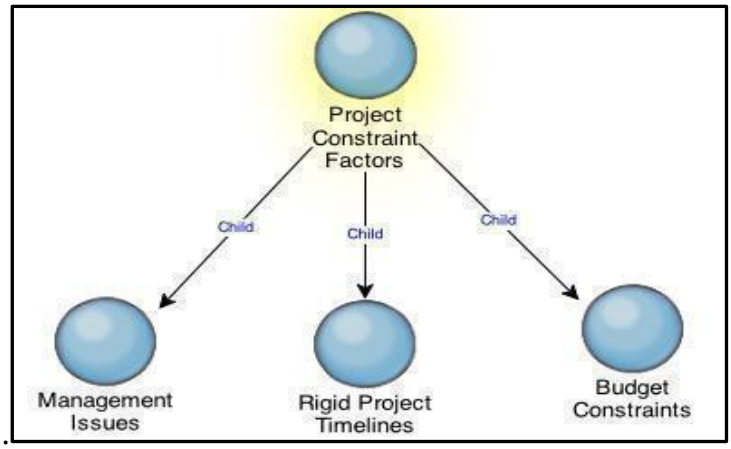

Figure 5.4. Project constraint factors

5.3.2.2. *Project constraints factors*

This theme also had three child themes that were aggregated from it. These were as follows:

i. **Management issues:** In the discussions with the team leads, research participants from Coy B and Coy C emphasised the importance of management involvement in supporting ST adoption:

There are non-technical approaches and part of them is compliance and regulatory policies. These should be enforced by the management to ensure security testing adoption (ProMan C).

Security is affected because of the pressure on team members due to huge business demands from management (ProMan B).

ii. **Rigid project timelines:** According to the research participants, this was a child theme derived from the project constraints theme:

The timeline also affects the quality of the ST process. Budget affects the resources needed to ensure appropriate implementation (ProMan A1).

We plan, and prepare well for security testing, but named factors of project timeline and huge business demands could pose a challenge to the proper execution of the plan (TestMan B).

iii. **Budget constraints:** The final theme that emerged from the project constraint theme was the budget constraint factor. This relates to the cost implication of ST. According to one research participant:

Security is cost intensive and often times the budget allocated to application projects can be very limiting (ProMan B).

The third parent theme that emerged under factors affecting ST adoption was the ethical and compliance factors from which two patterns were observed.

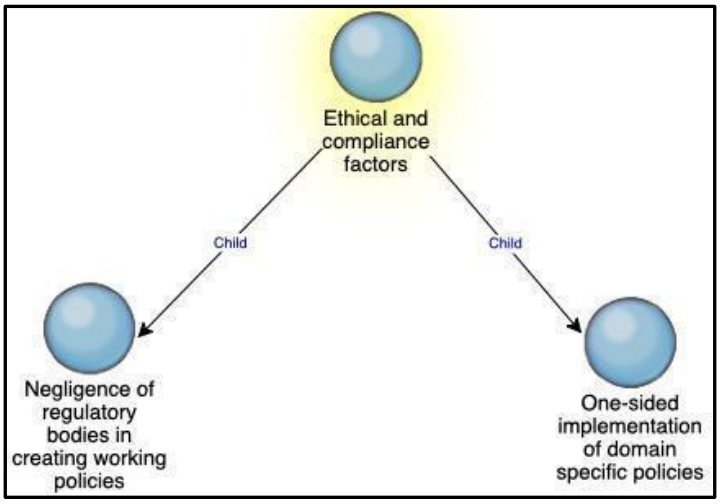

Figure 5.5. Ethical and compliance factors

5.3.2.3. *Ethical and compliance factors*

In Figure 5.5, two patterns were observed from the ethical and compliance factor. The analysis of the interview data revealed that although some teams were unaware of the available ethical and compliance strategies that guided the art of security in development with respect to the Nigerian IT industry, research participants from Coy C reported some known regulatory bodies that had developed specific compliance policies that related to the development process:

> *In the Nigerian IT industry, there are many ethical and compliance issues. This could be due to negligence of the necessary governing bodies to set viable regulations and policies* (ProMan A1).

> *We also have regulatory bodies like the PCI (i.e., payment card industry) that have strict policies we have to follow* (DevMan C).

From the thematic analysis of the PEOU, certain variables that influenced ST usage such as team size and technical competency were recounted by the research participants and this could have a significant impact on PU and PEOU.

5.3.3 USE themes related to research objectives

For the USE construct, a theme emerged that directly explains the fourth research objective of the study. Specifically, the theme indicates ways to improve security testing adoption among development teams in the Lagos IT industry. Figure 5.6 illustrates the five different child themes that emerged.

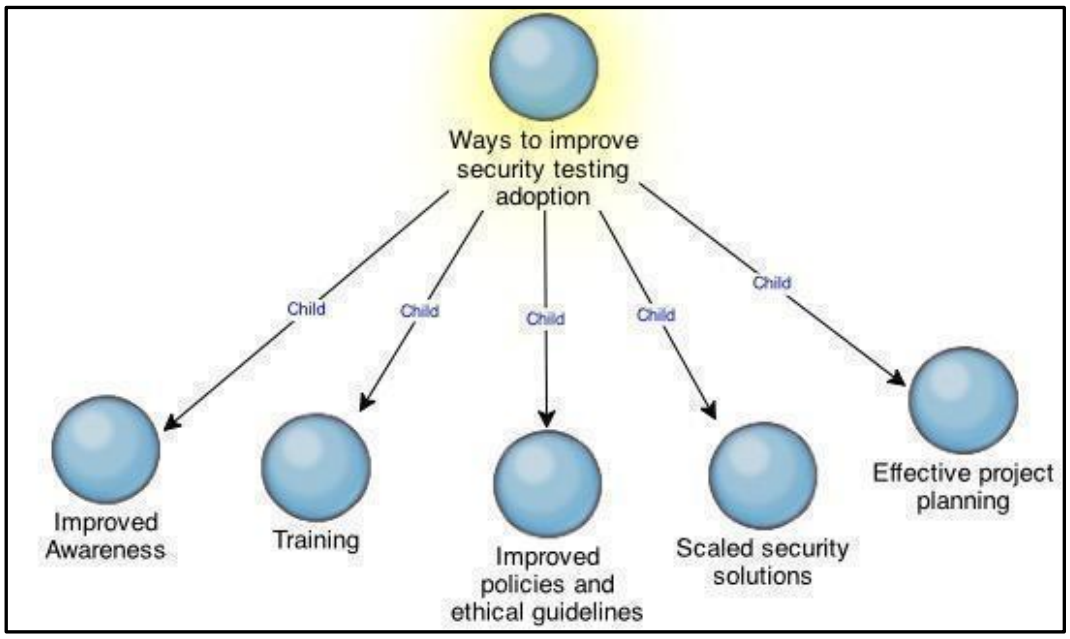

Figure 5.6. Ways to improve security testing adoption

As can be seen from Figure 5.6, the five child themes that emerged were as follows:

i. Improved awareness;
ii. Training;
iii. Improved policies and ethical guidelines;
iv. Scaled security solutions;
v. Effective project planning.

Excerpts from the research interviews are given below in support of these emerged themes:

i. **Improved awareness:** Improved awareness was reported by two of the research participants:

Constant awareness should be raised among development teams. Practical hacking sessions should also be done to ensure that teams are gaining hands on knowledge about implementing security tests (ProManA1).

More awareness and training need to be available. Tech writers, blogs and tech evangelists should share more discussions and information regarding preventing vulnerabilities among the many forums so that the ecosystem can be more aware and grow stronger (TestMan B).

ii. **Training:** The importance of training was reported by two of the research participants:

After awareness, we need to empower team members by training and then making the security tools to support the security testing process readily available (DevMan A).

Training should be intensified, and team members should be encouraged to be industry compliant to known standards (TestMan C).

iii. **Improved policies and ethical guidelines**: One research participant narrated that ethics was a major challenge in the Nigerian IT industry:

Ethics is a major challenge that needs to be improved in the Nigerian IT industry. Negligence of the necessary governing bodies to set viable regulations and policies affects security testing adoption (ProMan A2).

iv. **Scaled security solutions:** One research participant stated that companies needed to employ scalable security tools and approaches that were both fit for purpose and affordable:

Companies should look for a way to find scalable security testing tools and approaches that works for them and is still affordable to them (ProMan C).

v. **Effective project planning:** Two research participants recounted the need for effective project planning:

> *If the timeline is flexible and there is enough time allocated for the development and testing phase, then appropriate Security testing techniques will be applied* (DevMan A).

> *Security requirements developed for each project should be factored into project plan to guide the security testing implementation* (ProMan A2).

5.4. Chapter summary

This chapter discussed how the various themes that relate to the research objectives emerged. A detailed discussion of these themes and suggested ways of improving ST adoption and usage will be given in the chapter that follows.

CHAPTER SIX

DISCUSSION OF FINDINGS

6.1. Introduction

A careful review of the analysis done for this study as explained in chapters four and five revealed a number of findings. This chapter synthesises the research findings detailed in chapters four and five in order to answer the research questions. The discussion is presented in line with the research questions as reflected in the introductory chapter. The ST challenges that affect the development teams are also explained here and ways to improve adoption are proffered.

6.2. Discussion of research questions

6.2.1. Security testing approaches used in the Lagos IT industry

From the results of the qualitative analysis, three distinctive child themes emerged as the approaches used. These were as follows:

i. Basic security implementation following known security principles;
ii. Risk-based security testing;
iii. Security testing through the entire development life cycle.

6.2.1.1. *Basic security testing implementation following known security principles*

This ST approach was affirmed by participants from Coy A and Coy B. The security goals of confidentiality, integrity and availability of the implementation of security in specific phases of the SDLC of the application were ably demonstrated in the research data. In 2006, Tittle et al, (2006) has argued that certain basic security principles are known to guide security implementation as they provide an essential starting point for implementation considering that these security principles have been developed through extensive research. While there are many security principles in IS research, this analysis showed that specifically only confidentiality, integrity, availability and authentication

were mentioned by the participants. Talebi and Ayaburi (2016) have argued there are other factors such as security awareness and compliance which are relatively important in implementing security in the development life cycle. This approach helps to also measure the secure state of an application (Nieles, Dempsey, & Pillitteri, 2017).

This approach to ST implementation has been known to be used in large teams for security management (Cruzes et al. 2017). This study however revealed that it is also adopted among much smaller teams and can help to guide security implementation in specific features of an application. The use of security goals to guide the ST process in WA development is strategic because WAs have many critical features, including authentication and access control that could be focal points exploited by malicious hackers. As a result, these features need to be built securely and rigorously tested before deployment (Patil & Phil, 2014).

A major factor that may be responsible for the choice of this ST approach is the nature of the application domain of the development teams. For example, research participants from Coy B who used the ST approach built WAs for the travel and health application domains. The choice of ST approach may have been because the WAs in these domains are not typically high-risk or money critical applications. Another factor that may influence this choice is the lack of awareness about ST frameworks and their implementation. One of the research participants in this category stated that they mainly focused security implementation on specific features of WAs that were popularly prone to malicious attack.

Another observation noted within these teams was that the ST efforts were intensified in the testing phase of the SDLC. However, as the 2016 research by Mahendra and Khan (2016) has shown, there were possibilities of omitting certain security requirements in the earlier phases of the SDLC, which may make the application more vulnerable. This could further imply that certain members of the development team may not be involved in ST processes and there is more pressure on the developers and software testers who work in the development and testing phases of the SDLC (Sowunmi & Misra, 2015).

6.2.1.2. *Risk-based security testing*

A second approach noted from the themes that emerged from the qualitative data was the risk-based ST approach. The critical high-risk areas in the development process are identified and prioritised in risk-based ST. This risk-based approach is usually focused on the design and coding phases of the development life cycle. A research participant, ProMan C reported that it helps to measure the effort applied in testing an application.

From the thematic analysis, participants stated that their team carefully identified high-risk areas that were based on functionality and impact on other features of the application. The impact asset is typically the feature with the highest possibility of an attack determined by a risk assessment guided by the results of the risk assessment. Known ST techniques are then applied to the high-risk areas to prevent vulnerabilities (Landoll, 2005). Some related research also affirms that this ST approach highly optimises the security process because it involves a critical analysis of both the requirements and the features of the application to be developed (Srilatha & Someshwar, 2011; Tian-Yang et al. 2010).

It was observed from this study that there were specific security techniques applied alongside risk-based ST to ease the process. Some techniques mentioned included: code review, threat analysis, modelling attack trees and penetration tests. In a study conducted in 2006, McGraw affirmed the importance of security techniques in implementing ST in each phase of the SDLC as they ease the process (McGraw, 2006).

Another observation noted from the study was that teams that utilised the risk-based ST approach were in the range of small to medium (SMEs) scaled companies with about 25-50 members. These SDTs reported that their choice of this approach was due to their simultaneous involvement with multiple WA projects. Accordingly, there is a need to prioritise resources, while still ensuring the use of appropriate tools and techniques. AS Ume & Chukwurah (2012) have confirmed, the nature of projects and the time allocated for implementation can influence the approach used in testing.

6.2.1.3. *Security testing through the entire life cycle*

The third approach to ST among the teams involved a complete end-to-end process through the development cycle. Activities in this ST approach span from the planning phase to requirements, implementation and through to the post-deployment phase of the application life cycle (Kaushik & Mohan, 2013). One research participant provided a careful description:

> *After the entire team identifies the requirements and designs the model for the application, the developer working on the application works and implement security features in the application build. In the second stage, the application developed is pushed to the staging (testing) environment and then the quality assurance team tests again in this staging environment. The final stage, often times there is a periodic security sweep done in production level to ensure that the application is safe even after post-deployment* (DevMan C).

The ST process involves activities for each stage of the SDLC of the application (Mahendra & Khan, 2016). The process usually commences with stakeholders at the project meeting deliberating on the security requirements for the project in the planning phase. Potential vulnerabilities are also identified and noted. In the design phase, a model for the WA and attack trees depicting possible points of vulnerabilities in the application is created. In development, the previously-designed models serve as frameworks to guide the integration of the various features of the application being developed. Critical code review is performed by experienced or senior members of the team to analyse the code and identify missing security implementation. The team then resolves issues and the project proceeds to the testing team. The testing team analyses the application by using security tools and scanners to check for vulnerabilities and also simulate load and stress to the application to test its performance behaviour. The application can then be deployed based on the approval of the test team. Post-deployment tests are also performed to verify that the application functions as expected.

A major advantage of this approach is that roles and clear responsibilities are well-defined for each member of the team. This approach has been suggested in many studies

(Broström, 2015; Kaushik & Mohan, 2013). Although all the critical phases of the SDLC are covered in this approach, Kaushik and Mohan (2013) have argued that it could be complex to adopt if not aided by appropriate techniques through the SDLC. Some techniques used alongside this ST approach as reported by the research participants include security requirements, modelling attack trees, code reviews and analysis, penetration testing, post-deployment scans – all of which are mentioned in similar studies (McGraw, 2006).

From the interviews, it was observed that known security frameworks/methodologies such as the OWASP which encourages end-to-end ST through the SDLC had been adopted by the team in Coy C. As one participant was to report:

> *In web development, there are available frameworks like the OWASP that we adopt to enable us identify web vulnerabilities like CSRF and XSS. We spend time learning about these frameworks (DevMan C).*

This indicates that compared to the other teams, there was more awareness about ST in Coy C compared with the two other companies visited. Srinivasan and Sangwan (2017) have underlined the importance of security frameworks/methodologies such as the OWASP and emphasised on its suitability for WA development due to its wide adaptability and flexibility to identify vulnerabilities.

6.2.2. Significance of perceived usefulness, perceived ease of use and attitude in influencing security testing usage

As previously discussed in Section 4.2, all the constructs were reliable to measure. It was also observed that the PU construct was the most significant with a mean of 4.25. This implies that there was a general perception that using and adopting ST was important and that it could improve their performance in their respective roles within the SDLC. It also implies that to a large extent, participants in the sample understood that ST improved the quality of WAs produced by their SDTs.

As the items in the PU indicated, fixing application defects early, prevention of security defects and vulnerabilities in time were important to improve participants' job performance in the SDLC. Jaiswal et al. (2014) have stated that although ST is much

more important than just finding and detecting defects, defect prevention is an essential part of the SDLC and the findings from the present study further corroborates their argument. Furthermore, from the thematic analysis of the qualitative data, all job roles represented (i.e. software developers, software testers and project managers) stated that they had some form of approach to ST. This implies that ST is important for performance in their respective job roles. This research finding supports the results of the 1012 study by Ume and Chukwurah which affirmed the importance of security in application development regardless of role. The PU construct has always played a large role in determining adoption in TAM-related information systems studies, (Stewart, 2013; Szajna, 1996) and its significance in this study further shows that perceived usefulness is an important construct that explains the adoption of technology.

The PEOU was significant with a mean of 3.28 ($p < 0.001$). Since the PEOU items were designed to determine the effort required and the ease of adopting ST in the SDLC, the significance of the PEOU implies that to an extent, the research participants understood that using and adopting ST would be free from effort. In terms of secure development, s Talebi and Ayaburi (2016) have noted, the ease by which ST can be integrated into the development life cycle is largely affected by the type of SDLC model adopted for development. As discussed in the literature review, the level of effort needed in implementing ST in the various SDLC approaches in development differs.

From the results, two patterns were observed indicating the types of SDLC typically used by development teams in Lagos. These were:

i. A hybrid of the Waterfall and Agile SDLC
ii. The Agile SDLC

These two patterns used by development teams were clearly inferred by two of the research participants:

> *We adopted Agile recently, and we are still trying to stabilise and adopt it. We still use a hybrid of the Waterfall and the Agile SDLC in our project* (ProMan1a).

> *Oftentimes the Agile SDLC is difficult to adopt in certain situations practically. So, we use a hybrid of the Agile and Waterfall SDLC in our development* (ProMan1b).

The hybrid usage of the Waterfall and Agile SDLCs was observed in Coy A and Coy B. In a related study by Mahadevan, Kettinger, and Meservy (2015), it was observed that since transitioning from the Waterfall to Agile, SDLC was not easy and could disrupt coordination between SDTs. Accordingly, a hybrid approach of the Waterfall and Agile SDLC models was adopted by some teams. A hybrid approach enables teams to map Agile processes to Waterfall processes across each phase of the SDLC and this increases the output within each team. However, for those teams who adopt the hybrid Agile and Waterfall SDLC, a major challenge may the effective mapping out of defined ST strategies (Mahadevan, Kettinger, & Meservy, 2015).

Another pattern observed was the full adoption of the Agile methodology by the team in Coy C. They reported that it was flexible and allowed for progress and visibility in their projects. From the results, it is not surprising that Coy C also had an end-to-end ST approach. Each phase of their SDLC had a corresponding ST activity and hence there was easy integration. This implies that to an extent the inclusion of the Agile SDLC in both patterns indicates that this methodology supports ST integration compared to other methodologies. It also implies that applying ST in Agile is free from effort to some extent. According to Erdogan (2009), an Agile methodology is one of the SDLC models that provides easy integration with security. The Agile SDLC also supports the adoption of known security frameworks/methodologies such as the OWASP which is well-suited for WA security (Chóliz et al. 2015; Türpe, 2008; Broström, 2015).

Although the ATT had a mean of 3.22, it was not however significant that the p-value was higher than 0.001 ($p>0.001$). This implies that the ATT construct was not significant in this sample to predict usage. Nevertheless, from the thematic analysis of the qualitative data, it was observed that while teams generally had a positive attitude towards implementing security in the SDLC, many factors influenced their willingness and attitude. With respect to adopting new ST techniques, some team members expressed negative attitudes towards project timelines and inadequate technical

competencies as they struggled with systems that they may not have mastered. As one research respondent was to recall:

> *I am not typically excited about the ST process because it could be complex and require a certain level of expertise. However, if some form of technical support or training is given, I may be more encouraged* (Proman A1).

According to Binuyo et al. (2015), complexity could be a factor that affects software product development in Nigeria. Srinivasan and Sangwan (2017) have also asserted that the complexity of ST frameworks could adversely affect the adoption of ST. It was in this regard that questions were posed to the participants to understand if complexity affected ST adoption.

While some research participants (e.g. Proman A1) felt complexity adversely impacted their attitude towards ST, others viewed complexity as a means to learn and build more skillsets and agility within their teams regardless of any complexities in the ST process. As two of the research participants were to narrate:

> *Complexity does not impact on security because there are basic security implementations that are simple and easy to integrate in development. Some exist in open source code libraries in some resource groups on the internet* (DevMan B).

> *Security is really important and we find a way around a set of seemingly complex scenarios. We work and brainstorm as a team* (TestMan C).

These divergent views from the thematic analysis further affirm the findings from related research about the significance of the ATT construct in technology adoption as it can be a partial mediator between PU and BIU in determining USE.

Davis, Bagozzi, and Warshaw (1989) argue that although users' acceptance behaviour is solely determined by PU, PEOU and BIU, the ATT could be a partial mediator in the causal relationships in the TAM. In 1996, when reviewing the TAM and creating the

TAM2, the ATT construct was removed (Venkatesh & Davis, 1996). However, more recent studies still adopted the older version of the TAM and the ATT served as a partial or full mediator between PU and BIU. In 2001, Moon and Kim adopted the TAM with the addition of a new variable to understand the adoption of the WWW. In their study, ATT partially mediated between PU and BIU (Moon & Kim, 2001). In 2000, Siponen adopted the TPB and the TAM2 to develop a behavioural science model to study organisational information security awareness. In his research, he expressly stated that explicit measurement of human attitudes in technology adoption may be difficult as the advantage of any indications from the measurement of ATT output may vary between individuals (Siponen 2000). This further affirms the result of the varying output in this study. The role of the ATT in the causal relationships in the CF will be discussed in Section 6.2.2.1.

The BIU was also significant. ($M = 4.10$, $p < 0.001$). This construct has been known to influence technology adoption in different IS studies. It is predicted by PU and ATT. The BIU explains a users' readiness to use a technology. From the items of the BIU in the questionnaires, most participants agreed that they would use ST in the SDLC because it was ethical and useful in their careers. They also significantly agreed that given the necessary support they would use ST.

From the thematic analysis, items in the BIU were posed to understand the behavioural disposition of each team towards ST in practice. It was observed that all teams were inclined towards using ST as they adequately planned for security at the inception of every project. Most managers defined security requirements to be implemented based on each project. The planning and the preparation of each team towards ST indicates a general intention to use. However, the major challenge noted in this area is the lack of technical support and human resources to guide or manage the ST process. Smaller teams do not have the needed support, and larger teams struggle with the cost of training and purchasing of tools and technology needed to support the security process.

USE was the last construct measured and it was also significant. ($M = 3.33$). From the items of the USE, it can be stated that to an extent that most teams adopted ST in most phases of the SDLC and also regularly engaged in ST learning and awareness. From the thematic analysis, it was observed that all participants actually used ST. Some team

leads reported that they championed the process in their teams. The significance of the USE construct is not surprising because a positive indication of the BIU leads to a positive indication in USE. However, a major factor that impacts this construct was an inadequate awareness about ethical and compliance policies which have an effect of ST usage and adoption. This factor is further discussed later in the chapter.

6.2.2.1. *Relationships between the constructs of the conceptual framework*

The relationships between the constructs were examined to determine how they predict the actual usage. The results of the ANOVA tests shown in Section 4.2.4 showed that some relationships were not significant to predict USE.

In the original TAM, external variables serve as inputs to the PU and PEOU constructs and could provide some variance to these two constructs (Venkatesh & Davis, 2000; Lee et al. 2003). As shown in section 4.2.3, the results of the crosstabulation tests revealed that though none of the external variables had an impact on PU and PEOU, job role is a significant determinant of the choice of security testing activity applied in the SDLC. This similar pattern has been observed in similar IS studies in which external variables did not provide any significant impact on PU or PEOU (Burton-Jones & Hubona, 2006).

For the relationship between PU and PEOU, the ANOVA test results revealed that PEOU had a significant impact on PU. The PEOU accounts for 15.9% ($R^2 = 0.159$) of the variance in PU ($f(1.80) = 15.080$, $p<.0005$). This indicates that the degree to which ST is free of effort largely impacts on the degree to which ST will enhance the job performance of the individual members in the respective SDTs.

The relationship between PU, PEOU and ATT was examined and the analysis was carried out using ANOVA tests. The predictors were PU and PEOU, and the dependent variable was ATT. The results indicated however that neither PU nor PEOU are significant predictors of attitude (ATT). However, in this sample, it was observed that an increase in PU produced an increase in ATT, and a decrease in PEOU led to a decrease in ATT. This implies that that PU may impact positively on ATT while PEOU may impact negatively. The 2005 study by Johnson indicated that a high PU could give

a positive attitude towards technology use. A similar pattern was also observed in a 2017 study by Inci in which an increase in PU also caused an increase in ATT on adoption.

The relationship between PU and BIU was measured with BIU being the dependent variable. The results showed that PU significantly impacts on BIU and is a direct predictor. This implies that the degree to which ST improves job performance is a significant determinant of a user's readiness and willingness to use ST. This pattern has been observed in similar IS studies on adoption indicating that PU is a significant predictor of BIU (Lim & Ting, 2012).

The results of the analysis of the relationship between ATT and BIU also indicated that ATT does not predict BIU. This implies that a user's willingness to use ST is not determined by their attitude.

The last causal relationship examined was between BIU and USE. The results revealed that BIU significantly predicts USE. An implication of this significance is that a users' readiness and willingness to use ST determines the actual adoption and use of ST in the sample. This could largely be due to the significant impact of PU on BIU.

The significant relationships between the constructs of the CF were as follows:

i. PU is significant in predicting USE.
ii. PEOU is significant in predicting USE.
iii. PEOU impacts on PU in predicting USE.
iv. PU is a direct predictor of BIU.
v. BIU is a significant predictor of USE.

These significant relationships imply that PU, PEOU and BIU are key determinants in the adoption and usage of ST among web development teams in the Lagos IT industry.

6.2.3. Factors that affect the effective use of security testing approaches among web developers in Lagos

During the thematic analysis, specific factors were identified which affect the effective use of ST in the sample. From chapter five, three factors were identified. These were as follows:

i. Human resource factors.
ii. Project constraint factors.
iii. Ethical and compliance factors.

6.2.3.1. *Human resource factors*

Human resource factors were grouped as factors that directly affect individual members of the teams. Four specific factors identified under this category were identified. These were as follows:

i. Insufficient awareness.
ii. Limited team capabilities.
iii. Insufficient technical skills and expertise.
iv. Inadequate technical support.

i. **Insufficient awareness:** From the thematic analysis, a major factor that was observed to affect individual team members was a minimal awareness about ST frameworks and techniques within the SDLC. An implication of this was that the necessary technical and financial investments for implementing appropriate security strategies may not be adequately provided by the management of the teams (Patil & Phil, 2014).

ii. **Limited team capabilities:** As observed from the interviews, the size of each team seemed to have an impact on the choice of ST approach and the effectiveness of its implementation. Participants from Coy A and Coy B reported that team size and capabilities were influential in their choice of ST approach and techniques. Because their team sizes were small compared to the

projects they handled, they used risk-based ST approaches to enable them to prioritise their efforts in WA development. In comparison with Coy C which was the largest of the three teams, roles were well-defined and appropriate ST techniques could be applied in each phase of development. Similar to these results, Irefin et al. (2012) conducted a study that revealed that most companies in the Nigerian IT industry were constrained because of their small size. They also noted that many individual team members had to combine multiple functions and roles which put pressure on the quality of their outputs. According to a study by Choliz et al. (2015), the collaboration of teams through the SDLC was important to ease the ST process and ensure effective and proper coverage.

iii. **Insufficient technical skills and expertise:** This is a very crucial factor that affects all SDTs. All participants in the interviews stated emphatically that inadequate skills and expertise affected their ST implementation in some way or other. A similar study in the Nigerian context by Binuyo et al. (2015), revealed that skills shortage and lack of expertise has been a huge impediment to quality software development in the Nigerian IT industry. This corroborates with the findings from this study. According to Patil and Phil (2014), ST is an intensive process that requires adequate skills and expertise for proper implementation. This is because experienced members within teams can provide the needed guidance and support especially in the use of tools and appropriate frameworks (Patil & Phil, 2014; Howard & Lipner, 2006). This also corroborates with the findings from this study.

iv. **Inadequate technical support:** From the responses provided by the research participants, most SDTs enjoy insufficient technical support to guide their ST processes. This finding complements related studies in IS that have identified inadequate technical support as a challenge faced by teams in security implementation (Janes, Lenarduzzi, & Stan, 2017; Kaushik & Mohan, 2013). Choliz et al. (2015) maintain that security tools and frameworks developed through research cannot be effectively adopted if there is a lack of technical support to teams. A study by Mahendra and Khan in 2016 revealed that teams needed separate security teams to conduct the needed security evaluation within

organisations. Besides obtaining expertise from an external security teams, the findings from this study indicate the need for individual SDTs to be empowered with adequate technical skills.

6.2.3.2. *Project constraint factors*

Aside from the human resource factor which largely impacts on the individual members of each team, there were factors that affect a project or application in development. These were as follows:

i. Management issues.
ii. Rigid project timelines.
iii. Budget constraints.

i. **Management issues:** As the SDLC commences, business demands from management induce pressure on SDTs to ensure that project timelines are met. This pressure impacts on the quality of the security process and the effectiveness of team members. Related studies in the Nigerian context have shown that there was a lack of awareness among most management about the criticality and importance of many technical processes in the SDLC (Osho, Misra, & Osho, 2013; Sowunmi & Misra, 2015).

   There are non-technical approaches to security in the SDLC which relate to the managerial function and compliance. Business management drives business goals and often focuses on achieving these goals irrespective of technical limitations. It is important that the management in each company understand that ST is a crucial part of the business process. Such awareness will ensure the inclusion and financial investment in ST.

   In some related IS studies, management support significantly impacts on the adoption of technology within organisations (Irefin et al. 2012; Vernersson, 2010; Mouratidis & Giorgini, 2007).

ii. **Rigid project timelines:** Each activity in the SDLC is planned and scheduled to fit into a project timeline. From the results, research participants reported that the project schedule was usually inflexible and that this made it difficult for ST to be implemented through all the phases of the SDLC. The project timeline largely impacts on the activities in the SDLC, the SDLC methodology to be used, and consequently the ST approach. Although there is adequate planning, business demands interrupt the planned schedules and teams are constrained to neglect many aspects of the ST process. This could lead to vulnerabilities being introduced into the WA. This finding corroborates that of Sowunmi and Misra (2015) where rigid timeline constraints were shown to affect the implementation of adequate software quality practices in the Nigerian software industry.

iii. **Budget constraints:** The ST process is a costly and intensive one. The technologies that support the implementation of techniques and use of tools that need to be purchased from security firms can be cost intensive (McGraw, 2006; Sowunmi & Misra, 2015). From the results in this study, it was observed that the budget for WA projects was both strict and minimal. Most business stakeholders do not effectively factor-in the cost implications of implementing security when planning for projects. Tools to support implementation are costly, and the training required for members of development teams are also cost intensive. One research participant offered the following explanation:

Security is cost intensive and often times the budget allocated to application projects can be very limiting (ProMan B).

A related study by Wotawa (2016) emphasised the importance of an adequate budget to support the security process in organisations. Important resources such as money, time and expertise should be included in the budget planning (Wotawa, 2016).

6.2.3.3. *Ethical and compliance factors*

An interesting discovery emerging from the results was the lack and neglect of ethical and compliance strategies to guide the art of security in development with respect to

the Nigerian IT industry. While there are some generic policies and frameworks unique in some application domains such as in banking and ecommerce, teams that work in other domains are often unaware of certain policies and frameworks which are available to guide the implementation of security. For financial systems, regulatory bodies around the world develop standardised frameworks and ensure their compliance. These regulatory bodies are very effective as they verify that companies have the necessary frameworks implemented before they issue practicing licenses. Some examples include the NIST, OWASP, PTES, and PCI among others (Manjula et al. 2016).

Only research participants from Coy C attested to the fact that they followed the security policies of regulatory bodies such as the PCI. Other teams were inadequately aware and this relates to the lack of awareness factor discussed earlier. Different patterns and approaches to ST emerge because there is no clear-cut generic approach that relates to the Nigerian IT industry. From related research in the Nigerian context, ethics and compliance are factors that affect the actual usage of appropriate software engineering practices in Nigeria (Ume & Chukwurah, 2012). In another study, it was reported that most regulatory bodies in the IT industry are ineffective (Sowunmi et al. 2016; Ume & Chukwurah, 2012). This could result in companies adopting random practices that are not standardised.

6.2.4. Ways of improving security testing usage among web developers in the Lagos IT industry

From the previous section, certain factors that limit the use of ST among developers and their teams were discussed. These were as follows:

i. Improved awareness.
ii. Training.
iii. Improved ethical guidelines.
iv. Scaled security solutions.
v. Effective project planning.

i. **Improved awareness:** From the earlier stated factors, a deeper level of awareness is needed among teams as well as generally across the industry as

many teams collaborate on projects. Awareness needs to be created among SDTs about the right security tools and techniques to be used. As collaborations between SDTs occur, ideas can be shared and challenges in implementing ST can be discussed. Awareness needs to extend to management teams in each organisation so that needed investment to improve the ST process can be provided. An increased awareness across the industry will also improve the quality of the applications developed. As more quality applications that solve socio-economic problems are developed, this will lead to economic growth and create opportunities for investors. Consequently, there will be a reduction in software importation and an improvement in the degree of expertise among SDTs because awareness creates motivation and a willingness to learn.

ii. **Training:** The ST process in development is always evolving. As new vulnerabilities are discovered, security companies are constantly building solutions to mitigate and prevent vulnerabilities. New technologies are also introduced and SDTs must learn how to apply them effectively. SDTs need to be trained to understand how these new ST techniques and frameworks integrate into the SDLC. Training would empower SDTs with the needed technical skills required. Investing in training will enlighten team members about current security vulnerabilities and the tools and techniques that can be used to mitigate them.

iii. **Improved policies and ethical guidelines:** Policies provide a guide for the various activities and processes within an industry. Policies also assist in setting up measures and standards as well as providing justification for the ethical use of technology within an industry (Ume & Chukwurah, 2012). It is very important that appropriate policies suitable to guiding the art of ST in WA development companies be set by governing and regulatory bodies so that there will be a proper and unanimous approach to ST within the IT industry in Lagos, Nigeria. Management within teams should research and discover suitable standards and policies that are unique to their individual application domains to guide ST implementation. Policy and regulation is a non-technical aspect that drives appropriate implementation of ST and has a huge impact on the

techniques and frameworks to be applied (Broström, 2015; Nieles et al. 2017). Certain regulatory rules that enforce companies to disclose cyber breaches need to be set in place to ensure that the appropriate security is built into developed WAs. These will increase awareness within the industry as well uncovering such breaches and mitigating against further attacks (Manjula et al. 2016).

iv. **Scaled security solutions:** From the analysis, it was observed that security is expensive to implement and this is a factor that affects most teams. While there is no one-way approach to ST that outrightly applies to all teams, each team needs to find ways of adopting security in such a way that the additional costs do not become a limitation. Research should be conducted into developing security solutions that are scaled and fitted for adoption with respect to each application domain and team size. Scaled security solutions are designed specifically by security companies to suit certain development scenarios. These are both cost effective and beneficial in terms of helping teams achieve their goals regardless of the uniqueness of their application domain or limited team size.

v. **Effective project planning:** The project timeline constraint was a challenge stated by every participant in the interviews. As Sowunmi and Misra (2015) have argued, this challenge can be dramatically reduced if teams begin early in their planning for every activity and eventuality in the development process. Project managers need to adequately factor-in the time for ST activities in the SDLC implementation and development process. Additionally, from the feedback given by the participants, the project plan should include security responsibilities for each member of the team. With proper planning, teams can apply the necessary ST techniques and approaches and build more resilient and quality applications.

6.3. Chapter summary

In this chapter, a detailed discussion of the results from the analysis has been provided in line with the stated research objectives. The security approaches used by teams in the

Lagos IT industry differ across teams and this was explained in the first section of the chapter. It was observed that while most teams have a default ST approach applied in development, there are factors that are responsible for their choices. PU, PEOU and BIU were constructs that were significant and largely impactful in the causal relationships in the CF. Furthermore, certain factors that affect the usage of ST in the population were identified and grouped into human resource, project and ethical factors. In the concluding sections of the chapter, ways of improving the state of ST practice in the IT industry in Lagos, Nigeria, such as increased awareness, training and adequate planning were discussed. Finally, other ways to limit identified challenges such as improved budgeting for security and applying scaled security programs were suggested.

In the chapter that follows, a conclusion and summary of the study will be offered.

CHAPTER SEVEN

CONCLUSIONS AND RECOMMENDATIONS

7.1. Introduction

In the previous chapter, a detailed discussion of the results from the findings was given. Certain observations were made and the research questions were comprehensively answered. The main objective of this study was addressed, namely, to understand the factors that affect the adoption of ST in practice among web development teams in the IT industry in Lagos, Nigeria.

7.2. Summary of results

From the findings, it was observed that a ST approach is adopted by each of the three teams that participated in the study. Basic security implementation following known security principles, risk-based security testing and ST through the entire development life cycle are the three approaches used for ST. Although there is an awareness, the results revealed that across the industry there were no structured guidelines available for ST implementation across teams. This was due to several limiting factors, including ethical and compliance factors. Other limiting factors identified included human resource and project constraint factors. Challenges that affect ST implementation were identified as insufficient awareness, limited team capabilities, insufficient skills and technical expertise, inadequate technical support, rigid project timelines and budget constraints.

For this study, PEOU, PEOU and BIU were key determinants in predicting USE based on the causal relationships in the conceptual framework. It was also noted that though none of the external variables had a significant impact on PU and PEOU, job role could determine the choice of ST activities in the SDLC. This finding differs from similar TAM studies that revealed that each variable could be significant in predicting technology adoption (Borkovich, Skovira, & Breese-Vitelli, 2015; Paquet, 2013). This study also noted two new variables that may influence the PU and PEOU as indicated

by the experts interviewed. These variables were team size and technical competency. These variables may be further examined in future research.

Several ways of improving the adoption of ST in the Lagos IT industry were also identified. These include an improved awareness, regular training for SDTs to improve their technical skills and keep them up-to-date with current trends and technologies in the industry. More ethical and compliance policies suitable for adoption in the Nigerian IT industry should also be developed by the regulatory bodies. Adequate project planning should also be affected among teams and adequate time required for ST implementation should be created in the SDLC.

7.3. Recommendations from the study

From the results of the study, certain recommendations can be made. These are as follows:

7.3.1. Increased awareness about ST techniques and frameworks

From the findings of the study, it is clear that there must be an increased awareness about ST techniques and frameworks that are suitable for WA development. Security is a structured process that must be implemented with the right techniques and frameworks to build in resilience against WA attacks. Awareness must be improved across the Lagos IT industry so that SDTs can improve their skills and management teams can provide the needed technical support. As the awareness increases, teams will be able to understand the availability of suitable techniques and frameworks that can be applied in each development process and consequently improve the quality of applications built.

7.3.2. Resolution of limiting factors to ST

The second recommendation is that many of the limiting factors to ST should be both resolved and eliminated because similar patterns have been observed in related studies in the Nigerian industry. Rigid project timelines, insufficient skilled developers, inadequate training are factors that need to be eliminated. More training should be given to teams to improve their technical competencies and increase their knowledge.

While more training and investment in security is encouraged, individual developers should make every effort to do research and keep themselves up-to-date with libraries, resources and tools available on a regular basis. WA vulnerabilities are always evolving and attacks are ever more structured. Teams need to grow their skills and improve their competencies by researching and sharing knowledge.

7.3.3. Development of new ST techniques and frameworks

The third recommendation is that there is a need for a ST approach that will be suitable for teams that adopt the hybrid SDLC approach of the Agile and the Waterfall methodology. Each of these methodologies have their own constraints and teams should carefully observe their processes and decide which ST techniques and frameworks will be suitable in certain scenarios.

7.4. Limitations of the study

Certain environmental and financial limitations affected this study. Many companies could not create the needed time to participate in the study and hence the sample size was limited. In addition, the choice of the participating companies and the number of participants was constrained by the research schedule and timeline.

For the interviews conducted, the target audience were known experts in the IT industry in Lagos, Nigeria. Their availability was a major challenge due to their busy schedules. The researcher had to improvise; hence, some of the interviews were conducted at locations where the participants could be reached. The time allocated for the interviews was also often constrained by the availability of participants. Despite these stated limitations, this study offers valuable insights into the ST practices of software teams in Lagos.

7.5. Recommendations for future research

The TAM2 was adopted to design the conceptual framework for this study. However, because some of the external variables applied in the framework were not significant, some other technology adoption frameworks such as the UTAUT and TPB might be considered in future studies. An observation made during this study was that larger

teams had a planned structure towards ST implementation. Organisational size has long been known to be a significant factor in the adoption and implementation of ST technology. Accordingly, the variables of organisational team size and technical competency could be tested in future studies to determine their significance in ST adoption. Furthermore, a more comprehensive research is needed to assess whether the findings of this study would be different if more companies and participants were involved in the sample.

7.6. Final conclusion

In conclusion, the research objectives of this study have been achieved. Future work might involve an integration of the TAM and some other technology adoption model to eliminate the limitations observed in this study.

REFERENCE LIST

Abbasi, M. S., Tarhini, A., Hassouna, M., & Shah, F. (2015). Social, Organisational, Demography and Individuals' Technology Acceptance Behaviour: A Conceptual Model. *European Scientific Journal*, 11(9).

Ali, S. R., Khan, A., Baig, M. M. F., & Umer, A. (2015, December). Implementation of Kano's model in web metrics for information driven websites-KDQI. Paper Presented at the 2015 International Conference on Information and Communication Technologies (ICICT). https://doi.org/:10.1109/ICICT. 2015.7469577.

Andrews, M., & Whittaker, J. A. (2006). *How to Break Web Software: Functional and Security Testing of Web Applications and Web Services.* Boston, MA: Addison-Wesley Professional.

Aronson, J. (1995). A Pragmatic View of Thematic Analysis. *The Qualitative Report, 2*(1), 1-3.

Assad, R. E., Katter, T., Ferraz, F. S., Ferreira, L. P., & Meira, S. R. L. (2010 August). Security Quality Assurance on Web-Based Application through Security Requirements Tests: Elaboration, Execution and Automation. Paper Presented at the 5th International Conference on Software Engineering Advances (ICSEA). https://doi.org/10.1109/ICSEA.2010.48.

Atashzar, H., Torkaman, A., Bahrololum, M., & Tadayon, M. H. (2011 December). A Survey on Web Application Vulnerabilities and Countermeasures. *Proceedings of the 6th International Conference* on *Computer Sciences and Convergence Information Technology (ICCIT)* (pp. 647-652). Seogwipo: IEEE.

Avancini, A. (2012 June). Security Testing of Web Applications: A Research Plan. Paper Presented at the 34th International Conference on Software Engineering. https://doi.org/10.1109/ICSE.2012.6227054.

Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). State of the Art: Automated Black-Box Web Application Vulnerability Testing. Paper Presented at the 2010

IEEE Symposium on Security and Privacy (SP). https://doi.org/10.1109/SP.2010.27.

Binuyo, G. O., Oyebisi, T. O., Olayinka, A., & Afolabi, B. S. (2015). Evaluation of the Factors Influencing the Indigenous Software Products Development in Nigeria. *International Journal on Advances in ICT for Emerging Regions (ICTer), 7*(3) 1-8.

Boakye, K. (2014). *Nigeria: How Africa's Largest Economy is Prioritising Affordable Internet*. Retrieved 08 October, 2016, from https://a4ai.org/wp-content/uploads/2014/07/Nigeria-Case-Study_FINAL.pdf.

Borkovich, D. J., Skovira, R. J., & Breese-Vitelli, J. (2015). New Technology Adoption: Embracing Cultural Influences. *Issues in Information Systems, 16*(3).

Braun, V., & Clarke, V. (2006). Using Thematic Analysis in Psychology. *Qualitative Research in Psychology, 3*(2), 77-101.

Broström, A. (2015). *Integrating Automated Security Testing in the Agile Development Process*. KTH Royal Institute of Technology, Stockholm, Sweden.

Bryman, A. (2006). Integrating Quantitative and Qualitative Research: How Is It Done? *Qualitative Research, 6*(1), 97-113.

Burton-Jones, A., & Hubona, G. S. (2006). The Mediation of External Variables in the Technology Acceptance Model. *Information & Management, 43*(6), 706-717.

Casado-Lumbreras, C., Colomo-Palacios, R., Ogwueleka, F. N., & Misra, S. (2014). Software Development Outsourcing: Challenges and Opportunities in Nigeria. *Journal of Global Information Technology Management, 17*(4), 267-282.

Chao Peng, G., & Baptista Nunes, M. (2009). Identification and Assessment of Risks Associated with ERP Post-Implementation in China. *Journal of Enterprise Information Management, 22*(5), 587-614.

Chóliz, J., Vilas, J., & Moreira, J. (2015 August). Independent Security Testing on Agile Software Development: A Case Study in a Software Company. Paper

Presented at the 10th International Conference on Availability, Reliability and Security (ARES). https://doi.org/10.1109/ARES.2015.79.

Christ, P. (2005). Achieving Seamless Website Transformation: Promotional Implications of Static Versus Dynamic Websites. *Journal of Website Promotion*, *1*(2), 97-109.

Creswell, J. W., & Clark, V. L. P. (2017). *Designing and Conducting Mixed Methods Research*. Thousand Oaks, CA: Sage.

Cruzes, D. S., Felderer, M., Oyetoyan, T. D., Gander, M., & Pekaric, I. (2016 May). How is Security Testing Done in Agile Teams? A Cross-Case Analysis of Four Software Teams. *International Conference on Agile Software Development* (pp. 201-216). Helsinki: Springer Cham.

Damanpour, F. (1991). Organisational Innovation: A Meta-Analysis of Effects of Determinants and Moderators. *Academy of Management Journal, 34*(3), 555-590.

Davis, F. (1986). A Technology Acceptance Model for Empirically Testing New End-User Information Systems: Theory and Results. (Unpublished doctoral dissertation). Massachusetts Institute of Technology, Massachusetts, USA.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly, 13*(3), 319-340.

Davis, F. D., & Venkatesh, V. (1996). A Critical Assessment of Potential Measurement Biases in the Technology Acceptance Model: Three Experiments. *International Journal of Human-Computer Studies*, *45*(1), 19-45.

Ding, C. (2013). Cross-Site Request Forgery Attack and Defence: Literature Search.

Doğan, S., Betin-Can, A., & Garousi, V. (2014). Web Application Testing: A Systematic Literature Review. *Journal of Systems and Software, 91*, 174-201.

Dukes, L., Yuan, X., & Akowuah, F. (2013). A Case Study on Web Application Security Testing with Tools and Manual Testing. *Proceedings of IEEE Southeastcon*. https://doi.org/10.1109/SECON.2013.6567420.

Erasmus, E., Rothmann, S., & Van Eeden, C. (2015). A Structural Model of Technology Acceptance. *South African Journal of Industrial Psychology, 41*(1), 01-12.

Erdogan, G. (2009). Security Testing of Web-based Applications. (Unpublished master's dissertation). Norwegian University of Science and Technology, Norway.

Eshete, B., Villafiorita, A., & Weldemariam, K. (2011 August). Early Detection of Security Misconfiguration Vulnerabilities in Web Applications. Paper Presented at the 6th International Conference on Availability, Reliability and Security (ARES). https://doi.org/10.1109/ARES.2011.31.

Eze, O. R. (2013). Electronic Payment in Cashless Economy of Nigeria: Problems and Prospect. *Journal of Management Research, 5*(1), 138.

Felderer, M., Büchler, M., Johns, M., Brucker, A. D., Breu, R., & Pretschner, A. (2016). Security Testing: A Survey. *Advances in Computers, 101*, 1-43.

Felmetsger, V., Cavedon, L., Kruegel, C., & Vigna, G. (2010). Toward Automated Detection of Logic Vulnerabilities in Web Applications. *Proceedings of the 19th USENIX Security Symposium* (pp. 143-160). Washington: USENIX.

Feilzer, Y. M. (2010). Doing Mixed Methods Research Pragmatically: Implications for the Rediscovery of Pragmatism as a Research Paradigm. *Journal of Mixed Methods Research*, *4*(1), 6-16.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating Rigor using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods*, *5*(1), 80-92.

Finifter, M., & Wagner, D. (2011 June). Exploring the Relationship between Web Application Development Tools and Security. *Proceedings of the 2nd USENIX Conference on Web Application Development* (pp. 99-111). Portland: USENIX.

Fishbein, M., & Ajzen, I. (1975). *Belief, Attitude, Intention and Behaviour: An Introduction to Theory and Research*. Reading, MA: Addison-Wesley.

Flick, U. (2014). *An Introduction to Qualitative Research.* Thousand Oaks, CA: Sage.

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development, 9*(8), 28-35.

Gokhale, R., & Sharma, S. K. (n.d.). Challenges in Security Testing of Web Applications. Retrieved 20 November, 2017, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.2353&rep=rep1&type=pdf.

Gotterbarn, D., Miller, K., & Rogerson, S. (1999). Computer Society and ACM Approve Software Engineering Code of Ethics. *Computer, 32*(10), 84-88.

Gottipalla, A. K., & Yalla, P. (2014). A Survey Report on Security for Testing Phase of Software Development Process. *International Journal of Research and Applications, 1*(1), 6-11.

Graff, M., & Van Wyk, K. R. (2003). *Secure Coding: Principles and Practices.* Sebastopol, CA: O'Reilly Media, Inc.

Graham, D., Van Veenendaal, E., & Evans, I. (2008). *Foundations of Software Testing: ISTQB Certification.* London: Cengage Learning EMEA.

Greasley, P. (2007). *Quantitative Data Analysis Using SPSS: An Introduction for Health and Social Science.* Maidenhead: McGraw-Hill Education.

Guest, G., & MacQueen, K. M. (2008). *Handbook for Team-Based Qualitative Research.* Lanham, MD: AltaMira Press.

Gupta, V., & Singh, S. (2013). Security Threats in Software Development Life Cycle. *International Journal of Computer Engineering & Applications, 1*(1), 1-7.

Herzog, P. (2010). *OSSTMM 3: The Open Source Security Testing Methodology Manual.* Barcelona: ISECOM.

Hope, P., & Walther, B. (2008). *Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast*. Sebastopol, CA: O'Reilly Media, Inc.

Horlick, J. (2005). *Information Technology Security Testing: Common Criteria.* NIST Handbook Publication Series, *150-20*.

Howard, M., & Lipner, S. (2006). *The Security Development Life Cycle* (Vol. 8). Washington: Microsoft Press Redmond.

Huang, Y.-W., Tsai, C.-H., Lin, T.-P., Huang, S.-K., Lee, D. T., & Kuo, S.-Y. (2005). A Testing Framework for Web Application Security Assessment. *Computer Networks, 48*(5), 739-761.

İnci, B. (2017). Consumer Attitudes towards Private Shopping Sites in Turkey. *Chinese Business Review, 16*(1), 1-18.

Payment card Industry, PCI (2014). *Information Supplement: Best Practices for Implementing a Security Awareness Programme (Vol. 1)*: PCI Security Standards Council.

Irefin, I., Abdul-Azeez, I., & Tijani, A. (2012). An Investigative Study of the Factors Affecting the Adoption of Information and Communication Technology in Small and Medium Scale Enterprises in Nigeria. *Australian Journal of Business and Management Research, 2*(2), 1.

Jaiswal, A., Raj, G., & Singh, D. (2014). Security Testing of Web Applications: Issues and Challenges. *International Journal of Computer Applications, 88*(3), 26-32.

Janes, A., Lenarduzzi, V., & Stan, A. C. (2017 April). A Continuous Software Quality Monitoring Approach for Small and Medium Enterprises. *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion* (pp. 97-100). L'Aquila: ACM.

Johnson, A. M. (2005). The Technology Acceptance Model and the Decision to Invest in Information Security. *Proceedings of the Southern Association of Information Systems Conference* (pp. 114-118). USA: AISeL.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher*, *33*(7), 14-26.

Jokonya, O. (2016, May). The Significance of Mixed Methods Research in Information Systems Research. *Proceedings of the 11th Annual Midwest United States Association for Information Systems (MWAIS).* Retrieved 10th November, 2018, from https://aisel.aisnet.org/mwais2016/26.

Jovanovic, N., Kirda, E., & Kruegel, C. (2006). Preventing Cross Site Request Forgery Attacks. Paper Presented at the 2006 Securecomm and Workshops. https://doi.org/10.1109/SECCOMW.2006.359531.

Jovanovic, N., Kruegel, C., & Kirda, E. (2006). Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities. Paper Presented at the 2006 IEEE Symposium on Security and Privacy (S&P'06). https://doi.org/10.1109 /SP.2006.29.

Kang, Y.S., Cho, H.H., Shin, Y., & Kim, J.B. (2015). Comparative Study of Penetration Test Methods. *Advanced Science and Technology Letters*, 87(34-37).

Kaushik, S., & Mohan, T. (2013). Security in the Software Development Life Cycle. *International Journal of Engineering & Science Research, 4*(12), 1053-1056.

Khan, I. A., & Singh, R. (2012). Quality Assurance and Integration Testing Aspects in Web Based Applications. *International Journal of Computer Science, Engineering and Applications (IJCSEA), 2*(3), 109-116.

Khari, M., Sangwan, P., & Vaishali. (2016). Web-application Attacks: A Survey. *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development* (INDIACom), (pp. 2187-2191). New Delhi: IEEE.

Kelley, K., Clark, B., Brown, V., & Sitzia, J. (2003). Good Practice in the Conduct and Reporting of Survey Research. *International Journal for Quality in Healthcare, 15*(3), 261-266.

Krejcie, R. V., & Morgan, D. W. (1970). Determining Sample Size for Research Activities. *Educational and Psychological Measurement, 30*(3), 607-610.

Kripanont, N. (2007). Examining a Technology Acceptance Model of Internet Usage by Academics Within Thai Business Schools. (Unpublished Doctoral Dissertation). Victoria University, Melbourne, Australia.

Krishnaveni, S., Prabakaran, D., & Sivamohan, S. (2015). Analysis of Software Security Testing Techniques in Cloud Computing. *International Journal of Modern Trends in Engineering and Research, 2*(1), 417-424.

Kumar, R. (2005). *Research Methodology: A Step-by-Step Guide for Beginners*. Thousand Oaks, CA: Sage.

Landoll, D. J. (2005). *The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments*. Boca Raton, FL: CRC Press.

Lebanidze, E. (2006). *Securing Enterprise Web Applications at the Source: An Application Security Perspective.* OWASP: The Open Web Application Security Project.

Lee, Y., Kozar, K. A., & Larsen, K. R. (2003). The Technology Acceptance Model: Past, Present, and Future. *Communications of the Association for Information Systems, 12*(1), 50.

Leedy, P. D., & Ormrod, J. E. (2005). *Practical Research: Planning and Design*. New Jersey, NJ: Prentice Hall.

Lewis, W. E. (2016). *Software Testing and Continuous Quality Improvement.* Boca Raton, FL: Auerbach Publications.

Li, Y.-F., Das, P. K., & Dowe, D. L. (2014). Two decades of Web Application Testing: A Survey of Recent Advances. *Information Systems, 43*, 20-54.

Lim, W. M., & Ting, D. H. (2012). E-shopping: An Analysis of the Technology Acceptance Model. *Modern Applied Science, 6*(4), 49.

Mahadevan, L., Kettinger, W. J., & Meservy, T. O. (2015). Running on Hybrid: Control Changes when Introducing an Agile Methodology in a Traditional “Waterfall”

System Development Environment. *Communications of the Association for Information Systems, 36*,(5).

Mahendra, N., & Khan, S. A. (2016). A Categorised Review on Software Security Testing. *International Journal of Computer Applications, 154* (1), 21-25.

Malviya, V. K., Saurav, S., & Gupta, A. (2013 December). On Security Issues in Web Applications through Cross Site Scripting (XSS). Paper Presented at the 20th Asia-Pacific Software Engineering Conference (APSEC). https://doi.org/10.1109/APSEC.2013.85.

Manjula, R., Bagchi, K., Ramesh, S., & Baskaran, A. (2016). Policy Compliance in Information Security. *International Journal of Pharmacy & Technology, 8*(4), 22330-22340.

Mao, C. (2009, November). Experiences in Security Testing for Web-Based Applications. *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (pp. 326-330). Seoul: ACM.

Massey V., & Satao, K.J. (2012). Evolving a New Software Development Life Cycle (SDLC) Incorporated with Release Management. *International Journal of Engineering and Advanced Technology (IJEAT*), *1*(4), 25-31.

McDermott, J & Fox, C. Using Abuse Case Models for Security Requirements Analysis. *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99).* (pp. 55-64). Arizona: IEEE. doi: 10.1109/CSAC.1999.816013.

McGraw, G. (2006). *Software Security: Building Security In*. Boston, MA: Pearson Education, Inc.

McMahon, D., Seaman, S., & Buckingham, J. (2011). Non-profit Adoption of Websites and Website Types. *Journal of Marketing Development and Competitiveness*, *5*(6), 43-50.

Meucci, M., & Muller, A. (2014). The OWASP Testing Guide 4.0. The OWASP Foundation. Retrieved 12 January, 2016, from https://www.owasp.org/images/1/19/OTGv4.pdf.

Moon, J.W., & Kim, Y. G. (2001). Extending the TAM for a World-wide-web-Context. *Information and Management, 38*(4), 217-230.

Morse, J. M. (1994). Emerging from the data: The Cognitive Processes of Analysis in Qualitative Inquiry. In J. M. Morse (Ed.), *Critical Issues in Qualitative Research Methods* (pp. 23-43). Thousand Oaks, CA: Sage.

Mouratidis, H., & Giorgini, P. (2007). Security Attack Testing (SAT): Testing the Security of Information Systems at Design Time. *Information Systems, 32*(8), 1166-1183.

Munassar, N. M. A., & Govardhan, A. (2010). A Comparison Between Five Models of Software Engineering. *International Journal of Computer Science Issues, 5*, 95-101.

Nieles, M., Dempsey, K., & Pillitteri, V. Y. (2017). *An Introduction to Information Security.* NIST Special Publication, 800, 12.

The Ministry of Communications Technology. (2012). National Information and Communication Technology (ICT) Policy. Retrieved June 10, 2017, from https://www.researchictafrica.net/countries/nigeria/Nigeria_National_ICT_Policy_ (draft)_2012.pdf.

Nixon, R. (2012). *Learning PHP. JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites.* (4th Ed.). Sebastopol, CA: O'Reilly Media, Inc.

Obasemo, O. (2015). Why Lagos in Nigeria is Africa's Silicon Valley. Retrieved 12 January, 2017, from https://www.startupgrind.com/blog/why-lagos-in-nigeria-is-africas-silicon-valley/.

Odufuwa, F. (2012). Understanding what is happening in ICT in Nigeria. Research ICT Africa. Retrieved 21 May 2017, from

https://researchictafrica.net/publications/Evidence_for_ICT_Policy_Action/Policy_Paper_6_-_Understanding_what_is_happening_in_ICT_in_Nigeria.pdf

Ogheneovo, E. E. (2014). Software Dysfunction: Why Do Software Fail? *Journal of Computer and Communications, 2*(6), 25-35.

Ogunsola, E. (2016). The Nigerian Tech Ecosystem: Everything You Need to Know. Retrieved 23 November, 2016, from https://techpoint.ng/2016/09/30/ nigerian-tech-ecosystem-everything-you-need-to-know/.

Oladejo, B. F., & Ogunbiyi, D. T. (2014). Customised Software Development & Testing in Nigeria: Challenges and Way Forward. *International Journal of Science and Technology, 2*(7), 90-93.

Olivier, M. S. (2009). *Information Technology Research: A Practical Guide for Computer Science and Informatics*. Cape Town: Van Schaik.

Osho, L. O., Misra, S., & Osho, O. (2013). A Metric in Global Software Development Environment. *Pacific Journal of Science and Technology, 14*(2), 213-219.

Open Web Application Security Project. (2016). Information Systems Security Assessment Framework (ISSAF) Retrieved 02 February, 2017, from https://www.owasp.org/index.php/Penetration_testing_methodologies.

Oyetoyan, T. D., Cruzes, D. S., & Jaatun, M. G. (2016 December). An Empirical Study on the Relationship between Software Security Skills, Usage and Training Needs in Agile Settings. Paper Presented at the 11th International Conference on Availability, Reliability and Security (ARES). https://doi.org/10.1109/ARES.2016.103.

Oyewole, O. S., El-Maude, J. G., Abba, M., & Onuh, M. E. (2013). Electronic Payment System and Economic Growth: A Review of Transition to a Cashless Economy in Nigeria. *International Journal of Scientific Engineering and Technology, 2*(9), 913-918.

Salini, P., & Kanmani, S. (2012). Security Requirements Engineering Process for Web Applications. *Procedia Engineering, 38*, 2799-2807.

Nixon, R. (2012). *Learning PHP. JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites.* (4th Ed.). Sebastopol, CA: O'Reilly Media, Inc.

Pallant, J. (2016). *SPSS survival manual*. (6th Ed.) Maidenhead: McGraw-Hill Education.

Paquet, K. G. (2013). Consumer Security Perceptions and the Perceived Influence on Adopting Cloud Computing: A Quantitative Study Using the Technology Acceptance Model (Unpublished doctoral dissertation). Capella University, Minneapolis, MN.

Patil, D., & Phil, M. (2014). Challenges and Problems in Security Testing of Web-based Applications: A Study of Software Companies in Pune city. Paper Presented at the National Conference on Innovations in IT and Management, Chennai India. Retrieved 16 February, 2017, http://nci2tm.sinhgad.edu /NCIT2M2014_P/ data/NCI2TM_14.pdf.

Paul, M. (2016). *Official (ISC) 2 Guide to the CSSLP*: CRC Press. Florida, USA.

Pavlou, P. A., & Fygenson, M. (2006). Understanding and Predicting Electronic Commerce Adoption: An Extension of the Theory of Planned Behaviour. *MIS Quarterly, 30*(1), 115-143.

Pelizzi, R. (2016). Securing Web Applications. (Unpublished doctoral dissertation). Stony Brook University, New York, NY.

Prandini, M., & Ramilli, M. (2010). Towards a Practical and Effective Security Testing Methodology. In *Computers and Communications (ISCC), 2010 IEEE Symposium on* (pp. 320-325). IEEE.

Qian, L., Wan, J., Chen, L., & Chen, X. (2013). Complete Web Security Testing Methods and Recommendations. *Proceedings of the 2013 International Conference on Computer Sciences and Applications* (CSA-13) (pp. 86-89). Wuhan, China: IEEE.

Saldaña, J. (2015). *The Coding Manual for Qualitative Researchers*. Thousand Oaks, CA: Sage.

Saripalli, P., & Walters, B. (2010, July). Quirc: A Quantitative Impact and Risk Assessment Framework for Cloud Security. *Proceedings of the 2010 IEEE Third International Conference* on *Cloud Computing* (CLOUD 2010) (pp. 280-288). Miami, Florida: IEEE.

Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). *Technical Guide to Information Security Testing and Assessment.* NIST Special Publication, 800, 115.

Scarpino, J. J. (2010). Web Application Security Testing: An Industry Perspective on How Its Education is Perceived. *Issues in Information Systems, 11*(1), 142-153.

Schieferdecker, I., Grossmann, J., & Schneider, M. (2012 March). Model-based Security Testing. *Proceedings of the 7th Workshop on Model-based Testing*. Retrieved 16 February, 2017, https://arxiv.org/pdf/1202.6118v1.pdf.

Sekaran, U., & Bougie, R. (2016). *Research Methods for Business: A Skill Building Approach*. New Jersey, NJ: John Wiley & Sons.

Shanley, A., & Johnstone, M. N. (2015 December). Selection of Penetration Testing Methodologies: A Comparison and Evaluation. Paper Presented at the 13th Australian Information Security Management Conference (pp. 65-72), SRI Security Research Institute, Edith Cowan University, Perth Australia. Retrieved 16 February, 2017, from https://ro.ecu.edu.au/cgi/viewcontent .cgi?article=1181 &context=ism.

Shema, M. (2011). *Web Application Security for Dummies*. New Jersey, NJ: John Wiley and Sons.

Singh, A. & Kaur, P.J. (2019 July). Analysis of software development life cycle Models. *Proceedings of* the *second International Conference on Microelectronics, computing & Communication systems (MCCS 2017), (*pp. 689-699). Singapore: Springer.

Siponen, M. T. (2000). A Conceptual Foundation for Organisational Information Security Awareness. *Information Management & Computer Security, 8*(1), 31-41.

Soriyan, A., & Heeks, R. (2004). *A Profile of Nigeria's Software Industry*. Development Informatics Working Paper (21). Retrieved 16 February, 2017 from http://hummedia.manchester.ac.uk/institutes/gdi/publications/workingpapers/di/di_wp21.pdf.

Soriyan, H. A., Mursu, A. S., Akinde, A. D., & Korpela, M. J. (2001). Information Systems Development in Nigerian Software Companies: Research Methodology and Assessment from the Healthcare Sector Perspective. *Electronic Journal of Information Systems in Developing Countries, 5*(1-18).

Sowunmi, O. Y., & Misra, S. (2015). An Empirical Evaluation of Software Quality Assurance Practices and Challenges in a Developing Country. Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.129.

Sowunmi, O. Y., Misra, S., Fernandez-Sanz, L., Crawford, B., & Soto, R. (2016). An Empirical Evaluation of Software Quality Assurance Practices and Challenges in a Developing Country: A Comparison of Nigeria and Turkey. *SpringerPlus*, *5*(1), 1921.

Srilatha, R., & Someshwar, G. (2011). Security Testing for Web Applications in SDLC. (Unpublished Master's Dissertation). Blekinge Institute of Technology, Sweden.

Srinivasan, S. M., & Sangwan, R. S. (2017). *Web App Security: A Comparison and Categorisation of Testing Frameworks. IEEE Software, 34*(1), 99-102.

Stewart, L. (2013). Technology Acceptance in Organisations. (Unpublished doctoral dissertation). Kansas State University, Manhattan, KS.

Stuttard, D., & Pinto, M. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws.* New Jersey, NJ: John Wiley & Sons.

Symantec. (2016). Internet Security Threat Report. 21, 4. Retrieved 16 February, 2017, from https://www.symantec.com/content/dam/symantec/docs/ reports/istr-21-2016-en.pdf.

Szajna, B. (1996). Empirical Evaluation of the Revised Technology Acceptance Model. *Management Science, 42*(1), 85-92.

Talebi, N., & Ayaburi, E. W. (2016). Secure Software Development: A Developer Level Analysis. *Proceedings of the 11th Annual Midwest United States Association for Information Systems (MWAIS).* Retrieved 12 February, 2018, from https://aisel.aisnet.org/mwais2016/26.

Thomas, J., & Harden, A. (2008). Methods for the Thematic Synthesis of Qualitative Research in Systematic Reviews. *BMC Medical Research Methodology, 8*(1), 45.

Tian-yang, G., Yin-sheng, S., & You-yuan, F. (2010). Research on Software Security Testing. *World Academy of Science, Engineering and Technology, 70*, 647-651.

Tittle, E., Stewart, J. M., & Chapple, M. (2006). *CISSP: Certified Information Systems Security Professional Study Guide*. John Wiley & Sons.

Türpe, S. (2008). Security Testing: Turning Practice into Theory. *Proceedings of the 2008 IEEE International Conference on Software Testing Verification and Validation Workshop, ICSTW'08*. https://doi.org/10.1109/ICSTW.2008.38.

Ume, A., & Chukwurah, J. (2012). Underscoring Software Engineering Ethics in Nigeria's Fast-Growing Information and Communications Technology. *Asian Trans Comput,* 2, 21-30.

Vaismoradi, M., Turunen, H., & Bondas, T. (2013). Content Analysis and Thematic Analysis: Implications for Conducting a Qualitative Descriptive Study. *Nursing & Health Sciences, 15*(3), 398-405.

Vala, R., & Jasek, R. (2011). Security Testing of Web Applications. *Proceedings of the 22nd International DAAAM Symposium* (Vol 22, pp. 1533-1535). Vienna: DAAAM.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the Qualitative-Quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Systems. *MIS Quarterly, 37*(1).

Venkatesh, V., & Davis, F. D. (1996). A Model of the Antecedents of Perceived Ease of Use: Development and Test. *Decision Sciences, 27*(3), 451-481.

Venkatesh, V., & Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science, 46*(2), 186-204.

Venkatesh, V., Morris, M. G., & Ackerman, P. L. (2000). A Longitudinal Field Investigation of Gender Differences in Individual Technology Adoption Decision-Making Processes. *Organisational Behaviour and Human Decision Processes, 83*(1), 33-60.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 425-478.

Vernersson, S. (2010). Penetration Testing in a Web Application Environment. (Unpublished Bachelor's Dissertation). Linnaeus University, Sweden.

Williams, J. (2006). *OWASP Testing Guide*. The OWASP Foundation.

Wotawa, F. (2016 August). On the Automation of Security Testing. *Proceedings of* the *2016 International Conference on Software Security and Assurance (ICSSA)*, (pp. 11-16). Austria: IEEE.

Zia, Q. (2015). Metrics for Quality Assurance of Web based Applications. *Global Journal of Computer Science and Technology: E-Network, Web & Security*, 15(1).

APPENDIX A

INTERVIEW QUESTIONS

Security Testing Challenges of Web Developers in Lagos, Nigeria IT Industry

A Masters Research Study by

Ajayi Moyinoluwa I.

Discipline of Information Systems and Technology

College of Law and Management Studies

University of KwaZulu-Natal, Durban South Africa

Interview Questions (For managers, and leads of Development teams)

Section A: Background Information ***(External variables)***

1. Gender

Male Female

2. Years of Experience

☐ Less than 1 year ☐ 1-3 years ☐ 3-5 years

☐ More than 5 years

3. Job role/Description

☐ Software Development lead ☐ Software Tester/QA lead

☐ Team/Project lead

4. Application domain

☐ Payments/ Fintech/Banking/Insurance ☐ Health/Pharmaceuticals

☐ Retail/Ecommerce ☐ Travel/Hospitality

☐ Service providers/Consulting/Education

Section B. Technology Acceptance Parameters

a. PERCEIVED USEFULNESS – *How security testing is perceived and done by web development teams in Lagos, Nigeria*

(Cue: Awareness, Perceived value to application domain)

1. How is security testing carried out within your team? (OWASP, Penetration testing, reliance on Code technologies, Outsourcing to security experts)
2. Is the criticality of the security testing process dependent on the nature of the project?
3. Is technical know-how (expertise) a crucial requirement for implementing security testing strategies within your team

b. PERCIEVED EASE OF USE – *What factors influence the choice of the security testing approaches among Web developers in Lagos, Nigeria?*

(Cue: Ease of Adoption, Integration into development process)

1. What software development Approach do you adopt in building applications? (Devops, Agile, Waterfall, V-model)

2. What stage in the Software development life cycle is security mostly emphasised on in your projects?

3. What are the known factors affecting the adoption of security testing practises in each stage of the development process?

c. **ATTITUDE TOWARDS USING** – *How these factors affect the perception of developers in Lagos about adopting security testing in application development.*

(Cue: Disposition to use, Willingness to use)

1. How do the named factors affect/ impact on security testing in the development process?

2. Considering the nature of projects, if the factors were eliminated, would security testing still be adopted by members of the team?

3. Does Complexity affect the attitude of team members in implementing new strategies in the development process?

d. **BEHAVIOURAL INTENTION TO USE** – *This defines how the readiness of the leads and heads of teams to use security testing practises in the development process*

(Cue: Behavioural Intent, Readiness to Use)

1. What is your teams' behavioural disposition towards implementing and appropriate security testing practises for web applications?

2. How well does your team prepare and plan for implementation of security testing strategies in web application development?

3. Does your team have adequate technical support to implement security testing for each web application project?

e. ACTUAL USAGE- *How Security testing approaches and practises in applications development can be made more effective to improve its actual usage in the Lagos IT industry*

(Cue: Compliance/regulation, improved attitude towards using)

1. Considering your position, would you pioneer/champion a secure development practice in your team with adequate support (management or technical) or rather outsource considering its challenges?

2. What Compliance and regulatory policies currently exist to support security testing practice for easy adoption among software development teams?

3. What Approaches could be put in place to ensure compliance to standardised ST practises among teams in the Lagos IT Industry?

APPENDIX C

QUESTIONNAIRE

Questionnaire on Security Testing Challenges of Web Developers in Lagos, Nigeria IT Industry

A Masters Research Study by

Moyinoluwa Ajayi (216076371)

Discipline of Information Systems and Technology,

College of Law and Management Studies

University of KwaZulu-Natal, Durban South Africa

SECTION A: BACKGROUND INFORMATION (External Variables)

1. Gender (***Please Tick)***

Male	Female

2. Years of Experience

Less than 1 year	Less than 3 years	Less than 5 years	5 years and above

3. Job Role ***(Please select only one)***

Software Developer/Engineer	Software Tester/ Quality Analyst	Project Management/ BA

4. Application domain ***(Please select one domain in which you build applications most often)***

Payments/ Fintech/ Banking/Insurance	
Retail/E-commerce	
Service providers/Consulting/ Education	
Health/Pharmaceuticals	
Travel/Hospitality	

5. Which one of the security testing approaches do you apply most often when developing applications for your team? (***Select One Option Only)***

Developing security requirements and planning for it in the requirements gathering phase (Planning)	
Building security into the Design/Model of the application before actual development (Design)	
Reliance on Inbuilt code technologies and applying security frameworks during development (Development	

Penetration testing and Vulnerability assessments tests with tools (Metasploit, Wireshark, Nmap, Acunetix, etc.) during the testing phase after development	
Post deployment scans or Outsourcing to external security consultants after development.	

SECTION B: TECHNOLOGY ACCEPTANCE MODEL CONSTRUCTS

PLEASE NOTE: The Responses to the Statements below Should Pertain/Apply to the work you have done in the application domain you selected in Question (4) above.

Indicate your agreement with the following Statements:

	Statements	**Strongly disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly agree**
	Perceived Usefulness					
1	Security testing is important in the Development process (SDLC) of Web applications					
2	Using security testing practices prevents security vulnerabilities in the SDLC of Web applications					
3	Using security testing is useful in discovering application defects early in the SDLC					
4	Fixing application defects is faster and easier by using security testing approaches					
	Perceived ease of use					

5	Security testing techniques in the SDLC are simple and easy to learn					
6	Security testing frameworks are easy to integrate into the SDLC of applications					
7	It is easy for me to become skilful at using security testing techniques in the SDLC					
8	Security testing practices are easy to adopt as a part of my responsibilities in the SDLC of applications					
	Attitude towards using					
9	I like to use security testing in the SDLC because it helps to understand the application design better					
10	I adopt security practices because it helps my role in the SDLC and will help improve my team's processes and work output					
11	I prefer not to use security testing in the SDLC because it can delay my work deadlines					

12	I prefer not to not adopt security testing practices because it can make my work complex and cumbersome					
	Behavioural intention to use					
13	I would likely apply security testing in building applications to adhere to ethical and good coding practices					
14	I plan to use security testing in future in all developments because it is useful to my career					
15	I would use security testing in the SDLC as it is critical to the nature of the application					
16	With the necessary training and support, I intend to use adequate security testing approaches in the next required stages of the SDLC					
	Actual usage					
17	I adopt security testing in all stages of application development					

18	I engage regularly in activities to encourage security testing awareness and learning in my organisation					
19	I apply and use Regulatory policies that exist to support security testing practices in the SDLC frequently					

20. Indicate the software development process (SDLC) phases that you use security testing approaches and the percentage.

0%	1-25%	26-50%	51-75%	26-100%

APPENDIX B

ETHICAL CLEARANCE LETTER

UNIVERSITY OF KWAZULU-NATAL
INYUVESI YAKWAZULU-NATALI

06 October 2017

Ms Moyinoluwa Ajayi (216076371)
School of Management, IT & Governance
Westville Campus

Dear Ms Ajayi,

Protocol reference number: HSS/1666/017M
Project title: Security testing challenges of Web developers in Lagos, Nigerian IT industry

Approval Notification – Expedited Approval

In response to your application received on 08 September 2017, the Humanities & Social Sciences Research Ethics Committee has considered the abovementioned application and the protocol has been granted **FULL APPROVAL.**

Any alteration/s to the approved research protocol i.e. Questionnaire/Interview Schedule, Informed Consent Form, Title of the Project, Location of the Study, Research Approach and Methods must be reviewed and approved through the amendment/modification prior to its implementation. In case you have further queries, please quote the above reference number.

PLEASE NOTE: Research data should be securely stored in the discipline/department for a period of 5 years.

The ethical clearance certificate is only valid for a period of 3 years from the date of issue. Thereafter Recertification must be applied for on an annual basis.

I take this opportunity of wishing you everything of the best with your study.

Yours faithfully

.....
Dr Shamila Naidoo (Deputy Chair)

/ms

Cc Supervisor: Deepak Kumar and Professor Irene Govender
Cc Academic Leader Research: Professor Isabel Martins
Cc School Administrator: Ms Angela Pearce

Humanities & Social Sciences Research Ethics Committee
Dr Shenuka Singh (Chair)
Westville Campus, Govan Mbeki Building
Postal Address: Private Bag X54001, Durban 4000
Telephone: +27 (0) 31 260 3587/8350/4557 **Facsimile:** +27 (0) 31 260 4609 **Email:** ximbap@ukzn.ac.za / snymsnm@ukzn.ac.za / mohunp@ukzn.ac.za
Website: www.ukzn.ac.za

APPENDIX D

INFORMATION TO PARTICIPANTS

Security testing of Web developers, in Lagos Nigerian It Industry.

Informed Consent Document

This study aims to identify the challenges faced by web developers in applying appropriate security testing practises in the Lagos IT industry. Your participation in this study is voluntary and by participating, you are granting the researcher permission to use your responses. You may refuse to participate or withdraw from the study at any time with no negative consequence. There will be no monetary gain from participating in the study. Your anonymity will be maintained by the researcher and the School of Management, I.T. & Governance and your responses will not be used for any purposes outside of this study.

All data, both electronic and hard copy, will be securely stored during the study and archived for 5 years. After this time, all data will be destroyed.

If you have any questions or concerns about participating in the study, please contact me or my research supervisor at the numbers listed above.

Sincerely

Moyinoluwa Ajayi

Student No. 216076371@stu.ukzn.ac.za

Email: Moyin.ajayi@gmail.com

APPENDIX E

INFORMED CONSENT

CONSENT TO PARTICIPATE - Participants

I ……. have been informed about the study entitled (provide details) by Moyinoluwa Ajayi with student number 216076371

I understand the purpose and procedures of the study is to identify the challenges of security testing among web developers in Lagos, Nigeria IT industry

I have been given an opportunity to ask questions about the study and have had answers to my satisfaction.

I declare that my participation in this study is entirely voluntary and that I may withdraw at any time without affecting any of the benefits that I usually am entitled to.

I have been informed about any available compensation or medical treatment if injury occurs to me as a result of study-related procedures.

If I have any further questions/concerns or queries related to the study I understand that I may contact the researcher at 216076371@stu.ukzn.ac.za

If I have any questions or concerns about my rights as a study participant, or if I am concerned about an aspect of the study or the researchers then I may contact:

HUMANITIES & SOCIAL SCIENCES RESEARCH ETHICS ADMINISTRATION

Research Office, Westville Campus

Govan Mbeki Building

Private Bag X 54001

Durban

4000

KwaZulu-Natal, SOUTH AFRICA

Tel: 27 31 2604557 - Fax: 27 31 2604609

Email: HSSREC@ukzn.ac.za

Additional consent, where applicable:

I hereby provide consent to:

Audio-record my interview / focus group discussion YES / NO

Video-record my interview / focus group discussion YES / NO

Use of my photographs for research purposes YES / NO

_____ _____

Signature of Participant Date

_____ _____

Signature of Witness Date

(Where applicable)

APPENDIX G

LANGUAGE EDITING CERTIFICATION

We, the undersigned, do solemnly declare that we have abided by the University of KwaZulu-Natal policy on language editing. The dissertation was professionally edited for proper English language, grammar, punctuation, spelling, and overall academic style. All original electronic forms of the text have been retained should they be required.

GARY STUART DAVID LEONARD

UKZN Higher Degrees Certified Language Editor

Commissioner of Oaths V3358

07 May 2019

MOYINOLUWA IBUKUNOLUWA AJAYI

Student No. 216076371

07 May 2019