UNIVERSITY OF GHANA

SOCIAL CONSTRUCTION OF BESPOKE SOFTWARE FOR HIGHER EDUCATION MANAGEMENT IN A SUB-SAHARAN AFRICAN COUNTRY

BY

BIBI DZIFA TOMIWA

(10442379)

THIS THESIS IS SUBMITTED TO THE UNIVERSITY OF GHANA BUSINESS SCHOOL, UNIVERSITY OF GHANA, LEGON IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF AN MPHIL IN MANAGEMENT INFORMATION SYSTEMS DEGREE

JUNE 2015

DECLARATION

I do hereby declare that this thesis is the result of my own research and has not been presented by anyone for any academic award in this or any other university. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

…….. ……..

Bibi Dzifa Tomiwa Date
(Student)

CERTIFICATION

I hereby certify that this thesis was supervised according with the procedures spelt out by the university.

_____ _____

Dr. John Effah Date
(Supervisor)

_____ _____

Dr. Richard Boateng Date
(Co-Supervisor)

DEDICATION

I dedicate this work to the Almighty God who granted me life, strength and wisdom to enable me carry out this study and to my wonderful parents for their endless love, support, encouragement and prayer.

ACKNOWLEDGEMENT

I would never have been able to finish thesis without the guidance of my supervisor and faculty members, the help from friends and support and encouragement from my family.

I want to express my profound gratitude to my supervisors, Drs. John Effah, and Richard Boateng for their immense support, guidance, care and patience which provided me with the right kind of atmosphere to conduct this research. Also to Dr. Erasmus Addae, I say thank you for your support and advice.

I am also grateful to all my friends for their words of encouragement and support, especially Nurudeen Rufai, who was always willing to help and give his best suggestions.

I am deeply indebted to my parents and siblings for always cheering me on, encouraging me through their best wishes and most importantly praying for me. They were my backbone throughout this research.

Finally to the most High God who has seen me through to this point in my academic pursuit, I am forever grateful. The journey has just begun and I know and believe that He'll always be there to see me through.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

CITO	Chief Information and Technology Officer
COTS	Commercial off-the-shelf Software
CSD	Computer Science Department
ECIS	Enterprise Computing and Information Systems
ERP	Enterprise Resource Planning
FMIS	Farm Management Information Systems
GIS	Geographic Information System
HEIS	Higher Educational Information Systems
ICT	Information and Communication Technologies
ISD	Information Systems Development
ISTD	Information Services and Technology Directorate
IT	Information Technology
I&E	Infrastructural and Engineering
OSIS	Online Students' Information System
OSS	Open Source Software
PSS	Policy Strategy and Support
RSGs	Relevant Social Groups
SCOT	Social Construction of Technology
SDLC	System Development Life Cycle
SST	Social Shaping of Technology
UPSA	University of Professional Studies Accra
USIS	University Information System
VVU	Valley View University

ABSTRACT

This research set out to understand why and how higher education institutions in sub-Saharan Africa countries opt to develop bespoke IS instead of packaged proprietary software. The literature on Higher Education Information Systems (HEIS) stresses the selection, adoption and implementation of Enterprise Resource Planning (ERP) packages in higher education institutions paying touching briefly on the development of bespoke software for higher education management.

This study extends the existing knowledge on HEIS and Information Systems Development (ISD) research by drawing on the IS interpretive case study approach and the theoretical lens of the Social Construction of Technology (SCOT) to trace the choice and development of in-house bespoke software in two higher education institutions in Ghana, a sub-Saharan African country. Findings of the study revealed that higher education institutions find bespoke software development more cost effective than packaged software implementation and are motivated to develop bespoke solutions in-house because of the need to develop organizational capabilities and skills that would meet the information needs of the institutions. More also, the study revealed that rigid organizational cultures and practices of public higher education institutions can influence the choice of technological platforms adopted for bespoke development. Again, in terms of having access to software source codes, there are no major differences between the use of open source and proprietary tools for in-house bespoke software development.

This study also extends the limited scope of HEIS from the dominant focus of ERP adoption and implementation to the development of bespoke software for higher education management. Finally, the study suggests that further research be extended into

the decommissioning process of bespoke software and the security of bespoke information systems as it is beyond the scope of the SCOT theory used in this study.

CHAPTER ONE

INTRODUCTION

1.1 Research Background

Education is an instrument for transformation of the various sectors in any nation because of its vital roles in economic, social, political and technological development (Adewole & Fakorede, 2013). For any growth-oriented industry, change is inevitable and the education sector is no exception. Rapid growth in the academic field has made administrative task in the education sector complex (Krishnaveni & Meenakumari, 2010; Ternai, 2003; Wijaya & Manongga, 2012). Every nation require an educational system that will reduce the complexity and enhance the overall administration of higher education (Adewole & Fakorede, 2013; Krishnaveni & Meenakumari, 2010; Negash et al., 2012). The huge role of higher education in a society's economy and prosperity has added new dimensions to the measures of quality in higher education (Anaper et al., 2013).

In recent years, student enrolment in tertiary institutions have increased, making the work of administrative officers in charge of processing students' result a very tiresome exercise to deal with (Afolayan & Absalom, 2010). higher education institutions admits more students on a yearly basis, in addition to those already enrolled, thus processing students' academic record represents very significant challenges as it tends to require a great deal of human involvement thereby increasing the cost and causing delays. Given the profound economic challenges, that leads to a high need for cost reduction, companies and higher education institutions are exploring new and innovative ways to reduce their cost of operations, thus implementing IS seem to be the pertinent solution (Bumbac &

Orindaru, 2014). The rapidly reducing cost, increased advancement and wide adoption of Information Technology (IT) over the past decade, also continues to change higher education such that instructors, students and other stakeholders of higher education institutions operate in the vortex of this transformation (Kim et al., 2012). Implementing an IS in a tertiary institution is now a fundamental option since it supports all business functions and offer accurate, clean and stable data that is available in real time to users across multiple departments and business functions (Lupu et al., 2008).

HEIS is an organizational and technical system where ITs are realized and hardware and software are used for collection, processing, acquisition, storage, search and dissemination of information (Sagitova, 2012). A number of factors distinguishes HEIS from that of other institutions, organizations and enterprises. HEIS are based on geographically distributed computer systems, located in separate buildings, on different floors that are connected with each other and aim to increase the productivity of daily activities, eliminate duplicates and improve management efficiency. Considering the many internal variability of higher education institutions and the fact that they operate differently from other sectors, the decision to acquire a packaged proprietary software or to develop bespoke software becomes a dilemma to managers.

Packaged software, also known as off-the shelf software is developed with a high level of standardization, thus, core components are identical across all user organizations and is regarded as a solution to the difficulties with custom designed systems (Lucas Jr et al., 1988). Often times little or no configuration is needed for the software to fit the requirements of an individual organization. According to Howcraft and Light (2010) organizations that adopt packaged software are unable to feed in their requirements

before the development of the system occurs, rather they are only faced with a proven set of applications to choose from to suit their organizational needs. Microsoft, Oracle and Corel are some of the largest firms in the packaged software industry that build products for sale to a market, not to specific customers (Carmel & Sawyer, 1998). ERP is an off-the-shelf commercial integrated IS (Stamelos et al., 2003) that integrates functions across an organization into a single system to support an institution's financial, human resources and students' services transactions and processes (King et al., 2002). The implementation of ERP systems mainly involve the adaption of the system to the business needs of the enterprise through system customization and re-engineering of the business processes to match the system specifications. It is also critical to acquire external expertise, including vendor support, in order to facilitate successful off-the-shelf package implementation.

Bespoke software, on the other hand, is one that is exclusively developed as a unique system for an organization and usually based on very few standardizations. Therefore a software built for a company A within a particular industry may not meet the IS needs of company B even within the same industry (Rangarajan & Tiwari, 2014).

Existing studies on HEIS touched mainly on the selection, adoption, implementation and use of packaged software for higher education management (Dutta & Burgess, 2003; Lupu et al., 2008; Pollock & Cornford, 2004, 2005; Ternai, 2003; Zornada & Velkavrh, 2005). The focus of this study is therefore to extend the body of IS knowledge to the development of bespoke software for higher education management.

1.2 Research Problem

Countless studies have been conducted in the area of HEIS (Afolayan & Absalom, 2010; Courant & Griffiths, 2006; Dutta & Burgess, 2003; Effah & Abbeyquaye, 2013; Esteves & Pastor, 2005; Fowler & Gilfillan, 2003; Haab & Cramer, 2010; King et al., 2002; Lupu et al., 2008; Pollock & Cornford, 2004, 2005; Saadé, 2003; Sagitova, 2012; Zornada & Velkavrh, 2005). For instance, Pollock and Cornford (2004) evaluated the implications of implementing ERP solutions in Universities and noted that for higher education institutions to make the most of standardized software packages, they must learn to effectively manage the problems and difficulties that accompany translating generic software into the context of higher education institutions. Also in that regard, Pollock and Cornford (2005) analyzed the reasons why higher education institutions choose ERP solutions and established that, its use cannot be generalized easily across all higher education institutions due to the difference in specific context, requirements and practices, as a result, existing processes and practices within higher education institutions are drastically reshaped. Although other authors have argued that the gap between ERP systems and organizational idiosyncrasies can be bridged through customization, Pollock and Cornford (2005) noted that customizing generic ERP systems to uniquely fit the requirements of higher education institutions is notoriously difficult and therefore proposed that suppliers of packaged software should rather consider tailoring ERP packages to specifically cater for the needs of higher education institutions. Concerned about the way Universities struggle with the complex implementation of IS, Fowler and Gilfillan (2003) proposed an IS project management framework to provide a general guidance and to allow for stronger co-operation among the different stakeholder groups involved in the implementation. Their research was aimed at providing a basis for improving the likelihood of success in the implementation of ERP systems in

Universities. In an attempt to contribute to the understanding of ERP implementation in higher education institutions, Esteves and Pastor (2005) analyzed the major factors that affect it and concluded that problems associated with ERP implementations in higher education institutions are not technological, hence special attention be given to organizational factors and contextual influences. In another dimension, Courant and Griffiths (2006) investigated how Open Source Software (OSS) can be adopted for higher education management, since managers of higher education institutions are largely not happy about the cost and performance of packaged software. The authors suggested that higher education institutions can collectively develop open source applications that better suits the needs of partnering institutions. Higher education institutions are information intensive organizations, as such decisions about which IS project to prioritize is an important consideration for managers, Dutta and Burgess (2003) therefore proposed a technique named "simple multi-attribute rating technique" (SMART) to support the prioritization of IS projects.

In the context of sub-Saharan African Countries, IS development and implementation have been fairly researched into (Afolayan & Absalom, 2010; Effah & Abbeyquaye, 2013). For instance, Effah and Abbeyquaye (2013) drew on SCOT as a theoretical lens to understand how the Relevant Social Groups (RSGs) within an organization shaped the development and use of an OSS for students records administration in a HEI. Their findings demonstrated how interpretive flexibilities of RSGs within higher education environment can shape the development and use of a software and how the software can shape them in return during use. Afolayan and Absalom (2010) in their research identified the reasons why there are delays in the processing and release of students examination results in a HEI in Nigeria and how the implementation of an IS caused an

increase in the efficient delivery of service and benefits to both administration and students. A study by Chinyemba and Ngulube (2005) aimed at investigating how records are managed in higher education institutions also made reference to the fact that records are poorly managed as a result of inadequate IS and therefore suggested that functional analysis of business functions and requirement that can be incorporated into an IS should be carried out.

Different theories such as Diffusion of Innovation, Participatory Action Research, General Selection Theory, Social Shaping of Technology (SST) and SCOT have also been used extensively to study how technology evolve; the adoption and implementation of IS; and how they are shaped by the stakeholders involved (Dafoe, 2005; Dutton et al., 2004; Effah & Abbeyquaye, 2013; Howcroft & Light, 2010; Humphreys, 2005; Jakku et al., 2004; Kleinman & Klein, 2002).

All these researchers have contributed significantly to IS research, however there are some research gaps such as:

i. Issue Gap: Bulk of the previous studies in HEIS focused more on the benefits, selection, adoption, implementation and use of packaged software (King et al., 2002; Pollock & Cornford, 2004; Sagitova, 2012; Seeman & O'Hara, 2006; Zornada & Velkavrh, 2005) with minimal focus on development. Although the research by Phahlane and Kekwaletswe did not relate the scope of the study to the use of packaged software in HEI, it certainly did not focus on the development of IS. Whilst Effah and Abbeyquaye (2013), sought to understand the development and use of OSS software for higher education management, this

study seeks to understand the development of bespoke software for higher education management.

More so, existing literature on HEIS have failed to compare IS implementation in Public and Private Universities (Chinyemba & Ngulube, 2005; Dutta & Burgess, 2003; Effah & Abbeyquaye, 2013; Esteves & Pastor, 2005; King et al., 2002; Sagitova, 2012; Zornada & Velkavrh, 2005). Hence this research will engage in a comparative study of bespoke software development and use in private and public higher education institutions.

ii. Context Gap: While several studies have been conducted in the area of HEIS in sub-Saharan Africa, the bulk of it touched on its adoption, implementation and benefits to the institution (Afolayan & Absalom, 2010; Chinyemba & Ngulube, 2005; Krishnaveni & Meenakumari, 2010; Saadé, 2003). Accordingly, this research traces bespoke software development in higher education institutions in sub-Saharan Africa.

iii. Theoretical Gap: Various relevant theories have been proposed to explain social influence on technology. SCOT has been applied widely in IS literature to understand how technologies are influenced by social factors, however it centered mainly on selection and implementation of a technology in organisations without paying much attention to its relevance in bespoke software development (Howcroft & Light, 2010; Kleinman & Klein, 2002; Prell, 2009; Sahay & Robey, 1996; Ternai, 2003). Since bespoke software are developed to meet the specific needs of an organisation, it is important to understand how social factors shape

and influence their development, therefore the researcher devises the use of the SCOT theory.

1.3 Research Purpose

Following the research problem stated and the relevant research gaps outlined in the previous sections, the purpose of this research is to understand why higher education institutions in sub-Saharan African opt for bespoke software; how they carry out the development process; and how they are affected by their use.

1.4 Research Questions

Based on the research purpose, the study addresses the following questions:

1. Why do higher education institutions in sub-Saharan Africa opt for bespoke software?
2. How do higher education institutions in sub-Saharan Africa develop bespoke software?
3. What are the consequences of using bespoke software in higher education institutions in sub-Saharan Africa?

1.5 Research Methodology

This study adopts an interpretive approach as it helps to gain more insight into subjective interpretations and how practices and meanings are formed in order to understand social reality (Howcraft & Light, 2010; Orlikowski & Baroudi, 2001), thus the need of this stance to construct the explanations that accounted for the choice of bespoke applications and its development over packaged software. A qualitative research is considered the most suitable methodology for this study, as it is directed at providing an in-depth

interpreted understanding of the social groups that exists within the case institutions selected for this study by learning about their social and material circumstances, experiences, perspectives and history that consequently shapes the development of the in-house information systems (Ritchie et al., 2013). Given the interpretive stance of this study, the interpretive case study involving two higher education institutions in Ghana as cases, is used to trace the choice and development of in-house bespoke software.

1.6 Justification of Study

This study set out to understand the choice and development of bespoke software over packaged proprietary software by higher education institutions in sub-Saharan Africa. This study is considered important as it extends the current limited scope of IS research from the over-concentrated area of packaged software adoption and implementation to the development of bespoke solutions. Again, it contributes to knowledge by drawing on the SCOT theory to understand the influence of Relevant Social Groups (RSGs) of an organization on the development of bespoke software. In addition, it paves way for subsequent studies relating to bespoke information system development to be conducted.

This study will supply decision makers, software development team, managers of higher education institutions and other institutions adequate information to enable them make informed decisions about the choice and development of bespoke information systems.

1.7 Organisation of the Study

Following the above introduction, the study consists of seven more chapters. Chapter two focuses on detailed review of relevant literature on IS, HEIS, ISD, bespoke software,

open source software and packaged software. This chapter also helps to identify the various gaps that exist in ISD body of knowledge. Chapter three explores the theoretical foundation of this study which begins by describing the SCOT theory as the theoretical lens through which this study is viewed. This includes the discussion of the background of the theory and the analyses of its basic concept. The chapter also presents researchers who adopted this theory and how they used it in their research. The chapter then goes on to describe how this theory is relevant to finding the right answers to the research questions. Chapter four discusses the methodology adopted in conducting this research in order to find appropriate answers to the research questions posed. This chapter also discusses the interpretive research paradigm and why it is selected for the study. Chapter four also details how the study was conducted on the field, the research method used in the study, as well as data collection and analysis techniques used. Chapter five presents the findings of the study from two case Universities in Ghana that have developed and use bespoke software, with respect to the research purpose and questions. It discusses the rationale behind the choice of a bespoke solution and describes the development process of the solution after which the consequences of using such systems are also presented. Chapter six performs comparative analysis on the findings of the cases using SCOT which served as the theoretical lens through which this research is viewed. This is done to identify salient issues regarding bespoke development of HEIS. Chapter seven discusses the analysis of the findings of the research in order to answer the research questions posed in this chapter. The research questions are addressed in light of the reviewed literature in chapter two and the findings and analysis of the cases presented in chapters five and six respectively. Chapter eight provides a summary and conclusions for this study. It also provide recommendations for further research into bespoke systems development in higher education institutions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The previous chapter gave a background into IS, higher education institutions, and HEIS. It also discussed the research problem leading to the development of a research purpose which is to understand why and how higher education institutions in sub-Saharan Africa opt for bespoke software development instead of packaged proprietary software. This chapter tends to discuss and review relevant literature pertaining to IS, HEIS and ISD so as to unearth knowledge in this field, identify various research gaps and justify the need for this research.

2.2 Information Systems

Information has become a very valuable asset, as most organizations use it to gain competitive advantage over rival organizations in effect making information systems pervasive in all kinds of business organizations today (Panigyrakis & Chatzipanagiotou, 2011; Yeo, 2002). Information systems are software and hardware that support data-intensive applications (Salami & Ahmadi, 2010) and they offer potential for greater control of business processes that involves the monitoring and storage of large amounts of information (Lyons, 2005). The following section presents an overview of IS concepts relevant to this study.

2.2.1 Definition and Conceptualization of IS

Hardcastle (2008) defined IS as "a group of interrelated components that work collectively to carry out input, processing, output, storage and control actions in order to

convert data into information products that can be used to support forecasting, planning, control, coordination, decision making and operational activities in an organization" (p. 8). According to Yeo (2002) IS denote a wide combination of hardware, communication technology and software to manage information and it is considered as product of the coalition of stakeholders. IS serves to coordinate the work of many different organizational functions. An information system uses IT, procedures models, knowledge bases and databases to provide information that will support strategy, operations, management analysis and decision making in an organization.

2.2.2 Benefits of IS

IS implementation and use is of tremendous importance to organizations and users as it offers a wide range of benefits both tangible and intangible, quantifiable and non-quantifiable. This section discusses the benefits of IS to organizations and businesses. Irani et al. (2001) identified three levels IS benefits to organization, namely; strategic benefits, tactical benefits and operational benefits. IS provide information to management of an organization to enable them make decisions that ensure that the organization is controlled. Apart from IS been predominantly used to support managers in their decision making (Nasir, 2005), they also allow information to be readily shared among functional units within an organization and eliminates the need for local data storage and duplication thus offering great transparency and reduce the occurrence of data inconsistencies respectively. Brynjolfsson and Hitt (1996) noted that the primary reason for which managers invest in IS and IT was for enhanced customer service followed by the need to save cost. Organizations also consider IS as important tool to help understand the needs of their customers, to deliver greater value to them in order to retain them and gain their loyalty (Nasir, 2005). IS implementation and use provide

organizations with competitive advantage (Irani et al., 2001). The continuously changing, highly competitive environment, rapid development of economic globalization and IT have made many organizations to take advantage of IS to support key business processes (Abdallah et al., 2013). IS implementation have disseminated into different domains of business such as the maritime industry, industrial market, banking sector, health sector, agricultural sector and even in the sustainability of the environment because they offer new ways of improving efficiency and to survive in highly competitive markets (Gunasekaran & Ngai, 2004; Harsh, 1998; Jangra & Dua, 2012a, 2012b; Melville, 2010; Nasir, 2005). For example, a paper by Salami and Ahmadi (2010) show how the implementation and use of a Farm Management Information Systems (FMIS) played an important role in how well farmers are able have access to a wide range of information about many aspects of their farming deal with the increasing demands. The next sections presents a discussion on themes in IS research that are relevant to this study, since the field of IS extends over a broad range of issues.

2.3 Higher Education Information Systems

IS use in higher education institutions is intended to support the smooth relay of information resources to users, so as to offer them the opportunity to perform their various functions efficiently (Sagitova, 2012). It also allow for integration between the social players and participating entities of a university environment, by providing tools that enable online information exchange (Oliveira et al., 2011).

According to Krishnaveni and Meenakumari (2010) there are three main functional areas of information administration that are important to higher education institutions, and they are, student administration, staff administration and general administration. IS facilitates

these areas by allowing information to be transferred, stored, retrieved and processed by almost all who work, study or interact with a given institution. It is however important to note that HEIS are uniquely different from other enterprise IS because they intended to manage a variety of data that changes periodically based on the academic process (Lupu et al., 2008; Sagitova, 2012; Supriana & Permatasari, 2007). The structure of higher education institutions is complex and consists of a lot of diverse groups with overlapping responsibilities, making the management of information very tough, with the use of enterprise systems.

Sufficient research exist to point out that higher education institutions have resorted to the implementation of ISs to satisfy the arising demand for information both internally and externally (Dutta & Burgess, 2003; Pollock et al., 2003; Saadé, 2003; Zornada & Velkavrh, 2005). In a study to assess how IS are implemented and deployed in Romanian Universities, Lupu et al. (2008) remarked that large Universities tend to use commercial software packages whereas smaller Universities prefer self-developed solutions because of the cost of commercial software packages. Fowler and Gilfillan (2003) explored the consequences of implementing ERP solutions in Universities concentrating mainly on its project management with the hope to promote implementation success by offering suitable guidelines. The goal of their study was to necessitate successful implementation of ERP systems in higher education institutions by bridging the gap between the limited knowledge of the senior managers of higher education institutions in ISD issues and the lack of experience of IS professionals in the development of systems that uniquely fit higher education institutions. Findings of the study revealed that the concept of IS strategy alignment is vaguely understood within the University context, hence the relationship between IS strategy and project management strategy was not evident.

Esteves and Pastor (2005) pointed out that higher education institutions are usually limited in terms of budget and IT/IS resources and therefore expect to largely benefit from the implementation of packaged software, however, the study by Courant and Griffiths (2006), suggests that managers of higher education institutions are not satisfied with the performance and cost of software packages. With respect to cost, higher education institutions spend huge amount of money to implement and customize packaged systems, with significant cost incurred each time a new version of the software is released. Regarding performance, software packages are not well tailored to suit the needs of higher education institutions and because they are proprietary, making desired modifications is almost impossible.

Existing literature in this field has covered issues relating to the adoption, implementation, use and benefits of packaged IS for higher educational management (Dutta & Burgess, 2003; Lupu et al., 2008; Pollock & Cornford, 2004, 2005; Ternai, 2003; Zornada & Velkavrh, 2005) with case study being the popular and effective method used to investigate the issues surrounding their implementation (Chinyemba & Ngulube, 2005; Negash et al., 2012). Others have also discussed the use of open source tools for higher education management (Courant & Griffiths, 2006; Effah & Abbeyquaye, 2013), this paper is therefore focused on the in-house bespoke development of an IS for higher education management while still employing the case study method.

2.4 Thematic Research Areas on IS

IS literature is scattered across various themes. There is substantial effort in the field of IS that has gone into the factors that influence its development, implementation, use and

maintenance (Caldeira & Ward, 2003; Ćirić & Raković, 2010; Guinan et al., 1998; Irani et al., 2001; Lassila & Brancheau, 1999; Light, 2001, 2005; Light & Papazafeiropoulou, 2004; Sarker, 2000; Thong, 1999; Von Alan et al., 2004; Yeo, 2002). The following sub-sections presents a discussion of literature on themes that are relevant to this research including adoption implementation, use and maintenance.

2.4.1 Information Systems Development

ISD involve risks that realize dramatic consequences, hence, considerable skill is required to design and maintain IS as it changes the way people work; the processes of an organization; and more often the way customers relate to the organization (Lyytinen & Robey, 1999; Negash et al., 2012). Although ISD involves a wide range of obdurate social problems, organizations still invest in it for reasons such as business survival, improved efficiency and gaining competitive advantage (Ćirić & Raković, 2010; Williams & Edge, 1996). It has been reported in literature that higher education institutions are increasingly turning away from bespoke solutions to ERP systems (Fowler & Gilfillan, 2003; Pollock & Cornford, 2004, 2005), however Courant and Griffiths (2006), argue that it is not rare for higher education institutions, especially the large ones to consider finding their own solutions to enterprise problems, by way of adopting open source tools to develop bespoke software. Advancement in technology over the last decade has made it possible for the IS development to be moved away from the traditional cycle method to a more flexible and dynamic one that enables standardization and code reusability (Welke, 1994). Lyytinen and Robey (1999), on the other hand, claim that the advancement in development technologies is not sufficient to improve the rate of successful implementation of IS, thus, Burstein and Gregor (1999) proposed that if a new system or method is expected to be successful, then it must provide

better solutions to IS problems than existing systems or methods. That is, it should be faster, more efficient, more elegant, or have some other features that makes it a superior solution. In order for IS implementation to be successful, its development must not be centered only on technical factors but also on social and organizational factors (Gibson et al., 1999; Irani et al., 2001).

The System Development Life Cycle (SDLC) has been acknowledged as one of the earliest attempts made to collate activities involved in software development, however it has been argued not to be taken to represent the de-facto approach by which ISD is undertaken (Boahene & Bowles, 1999). SDLC models allow for IS to be built in a very deliberate, structured and methodical way, with each stage of the life cycle reiterated (Langat & Kamuren, 2013).

Table 1 - Summary of the basic stages in the SDLC

Stage	Activities Involved
Project planning, feasibility study	A high level view of intended project is established and goals are determined
System analysis, requirement definition	Project goals are developed into defined functions and operations. End user information needs are also identified and analyzed.
Systems design	Desired features and operations, including screen layouts, business rules, process diagrams, pseudo code and other documentation are described in great detail.
Implementation	The real code is written
Integration and testing	All the pieces of codes are brought together into a special testing environment to check for errors, bugs and interoperability
Acceptance, implementation and deployment	Software is put into production to run actual business
Maintenance	This stage is what happens during the rest of the software's life and it goes on seemingly forever. It involves changes, corrections, additions, moving to a different technological platform and more.

Adapted from Langat and Kamuren (2013)

Requirement engineering is crucial to the success of an ISD project as it plays an important role in fufilling the purpose for which the IS was intended (Engu et al., 2012; Saiedian & Dale, 2000). Schneidewind (1987) posited that requirements are more likely to originate from experience in the usage of a system than from an abstract specification in the design of a system at its early stages, therefore requirement gathering is as important in the maintenance of an IS as in its development.

2.4.1.1 The Development Process

Authors like Effah and Abbeyquaye (2013) and Irani et al. (2001) have contributed to the area of information systems development. The bespoke IS development presented in the case organization of Irani et al. (2001) required the support of a consultant to help facilitate its design, development and implementation. Similarly, the development of an open source application for higher education administration studied by Effah and Abbeyquaye (2013) pointed out that developers sought the help of consultants to bring it to a reality (Effah & Abbeyquaye, 2013). In the study by Irani et al. (2001) the case organization revised its business plan and processes in order to draw a project strategy for the development of an IS before commencing with it. This re-engineering allowed for the bespoke software to be aligned with best practices of similar software. Members of the organization were educated on the importance and benefits of integrated IS as well as the impact it would have on their job functions. Developing an IS may require the acquisition of resources such as skilled and knowlegeable people in the field of IS, hence the need for sending existing IS staff for external training to develop new skills or to recruit skilled personnels and students with software engineering, IS or computer science backgrounds on industrial attachment (Irani et al., 2001). Irani et al. (2001) noted that recruiting students with background in software engineering, IS and computer science to

work on an ISD is of great benefit to the organization as it would help sustain a constant flow of innovation, motivation and inspiration. The authors also noted that ISD may require the acquisition of appropriate hardware and software to maintain competitive advantage.

Large and complex IS are usually developed in modules. From the findings of Effah and Abbeyquaye (2013) study, it can be inferred that the OSS application developed for higher education administration was developed and deployed in modules with each module properly tested before live implementation.

2.4.1.2 Bespoke Software Development Versus Packaged Software Development

IS can either be developed for a specific user or organization or for a wider market. The development of an IS that is tailored to suit the specific requirement of a client is termed bespoke software development, whereas packaged software development targets a wider market as a tradable product licensed for use. Sawyer (2000) summarized the differences between packaged and custom IS development in a case study involving three development teams, out of which two are packaged software development teams. He then concluded that despite the many commonalities between packaged and bespoke IS development, there exists enough differences between the two domains that could profoundly change the way stakeholder groups view software. Some differences between bespoke software development and packaged software development identified by Engu et al. (2012) are presented in Table 2.

Table 2 - Differences between Bespoke Development and Packaged Software Development

Differences	Bespoke Development	Packaged Software Development
Initiation of Project	Initiated by customer and terminated based on a mutual agreement by customer and management of the software development organization when the final product is developed and released.	Initiation of a software package development is by product management and it is a continous process that allows the modification of the software based on the changing requirements of the its wide user base.
Objective	To satisfy the needs of a customer	To market the product and achieve a wider customer satisfaction. Releasing product onto the market at the right time and place
Success Criteria	Depends on the acceptance of the software by the customer	Depends on product value, product review, market share, time to release and the satisfaction of a large market
Requirement gathering	Gathering of requirements are easy and the customer is the source of the requirement	There is a wide range of requirements and gathering requirements are not easy. Requirements are usually gathered by teams through the release of an earlier version of the product
Analysis and Negotiation	Cost and future modifications can be negotiated with the software developer	Negotiation is not possibly, however new versions of a product are released periodically to satisfy new customer needs
Validation	Validation of product is done before release	Validation is done on a beta version of the product, flaws are corrected and original version of the product is released
Financial Risk	Fiancial risk is shared among the customer and the development company	The software developer is solely responsible for the expenses of developing a

		software product in order to gain the customers attention
Tpyes of customers and customer satisfaction	Product is developed based on the requirements of a known customer Customer satisfaction can be easily achieved through the design and development of a product that will satisfy customer needs	Customer is unknown, the focus is on the mass market where product consists of common featuresthat fufill the common needs of all customers. Different versions of the product are released to attain customer satisfaction
Relationship	Software developers are always closer to customers in a bespoke project.	There is a huge gap between software development company and the customer. Customers and software developers interact through intermediaries
Constraint Based Delivery	Release of software product is dependent on agreement between the customer and the software developers	Release of a product is dependent on the market based on the expectancy of such a product in the market
Invention of Requirements	No need of inventing requirements as there is a fixed set of customers	Since there are no fixed set of customers, and therefore no fixed set of requirements, new requirements are always invented

Constructed from (Engu et al., 2012)

User involvement is central to bespoke software development, whereas it isn't common in packaged software development (Sawyer, 2000). Based on Irani et al. (2001) study, organizations that opt for bespoke software development do so in the need for a system that would acknowledge their idiosyncracies, the need to avoid the increasing inflexibility of a packaged system and the fact that it provides new opportunity to gain operational support that is essential for success. In essence the authors suggest that bespoke solutions tend to address both human-organisational and technological issues in

software development project which is critical to its success. Courant and Griffiths (2006) study revealed that most managers of higher education institutions are dissatisfied about the performance of packaged software products, but they are also skeptical about the successful development of a reliable bespoke software that would meet their needs. In the same vein, a study by Irani et al. (2001) revealed that the manager of the case organization was initially not enthusiastic about bespoke IS development because it was more expensive than acquiring a packaged system, whereas many others within the organization saw it as a turnaround for the organization, hence supported it.

Findings from Irani et al. (2001) infers that the initiative to develop bespoke software stems up from the technical team of an organization. This was evident in the case study of the author's research that stated that the software selection and implementation team took the initiative and were responsible for the development of a business case for a bespoke IS. The organization involved in the study believed that they would gain more satisfaction with the results of their own system better than that of a rigid vendor software package.

A qualitative study by Alves et al. (2007) investigated the RE practices and challenges faced in packaged software development. Findings from the study show that the limited proximity between software developers and customers facilitates the invention of a software product requirement. Furthermore, the fact that time to market a software product is a strategic objective of the development company, requirements of a product are generally invented by the development company. The findings from this study also revealed that requirements are rarely documented, which is also the case of bespoke software development mainly because documentation does not act as a formal agreement

between the supplier and the customer. The authors also showed that market-driven software development companies conduct an ad-hoc gathering of requirements, which can also be attributed to limited time and budget constraints.

2.4.1.3 Proprietary versus Open Source Tools

A bespoke software may be built from scratch, with its source code files derived from other open source software (Gu et al., 2010). OSS is software whose program code can be accessed and modified without any constraint, whereas proprietary software is developed by a business enterprise to generate profit from licensing or selling the software (Khelifi et al., 2009; Von Krogh & Von Hippel, 2003). OSS development is characterized by an existing internet-based communities of software developers that collaborate to develop software that either they or their organizations need and are not likely to be successful without the existence of the OSS communities (Von Krogh & Von Hippel, 2003). Anyone with appropriate programming knowledge and enthusiasm can use, study, understand and amend any open source program written by anyone.

OSS development is usually preferred where the users need the software to solve their own or shared technical problems (Hippel & Krogh, 2003). Open source tools have upper hand over proprietary tools when it comes to custom user-specific applications; afford public institutions the opportunity to modify the way a functionality works to suit their needs; and provides them with the option of freely distributing software without having to bear the cost of licensing (Gupta, 2012; Steiniger & Hay, 2009; Von Krogh & Von Hippel, 2003). The use of OSS tools in software development eliminates license costs, which offers a solution to vendor lock-in in the future and also rids development of unecessary complexities and affords the freedom to make modifications to the tool (Aziz

et al., 2005). Proprietary software licenses on the other hand impose restrictions on the use of the software such that it does not allow the distribution and re-distribution of the software and prohibits modification since source codes are not made available to the users, hence facilitating vendor lock-ins (Steiniger & Hay, 2009). This implies that an organization that adopts open source tools can concentrate on its work rather than on vendor lock-in to rapidly migrate onto the new application which in turn allows the organization to take advantage of its prevailing investments.

Open source tools are also suitable for collaborative developments, because interoperability issues are eliminated (Gu et al., 2010). On the other hand, proprietary tools are ineffective in terms of interoperability, software transparency and transfer of data, so therefore not suitable for collaborative developments (Gupta, 2012). According to Gu et al. (2010), open source tools impressively hasten the software development process especially as users are in charge of all the source files and are able to adapt, configure and construct their own executables to be deployed in their own environment. Because the extent to which modifications are made to the open source software distribution by the different users who download them, cannot be predicted, and hence attested, Gu et al. (2010) introduced a source code based attestation scheme called SCOBA to enable proper attestation on custom software that has unpredictable versions and settings. There is a lack of systematic method for discovering and classifying OSS requirements in development projects (Vlas & Robinson, 2012). This makes it difficult to better understand the requirements, life cycles, scope, goals and the overall direction of the projects. In view of this, the authors developed tools that are capable of discovering requirements and classifications from the natural forms found in OSS developments.

They also contributed to practice by creating tools that can aid in the automated analysis of requirements.

2.4.2 IS Implementation

IS implementation as defined by Sarker (2000) refers to "all that must be done by a specific organization for it to be able to harness the capabilities of a particular information technology as envisioned" (p. 1). Implementing an IS involves the design, delivery and use of the system in an organization (Yeo, 2002). An information system is considered successful if it is able to perform reliably and dependably as promised and is able to offer prompt service (Landrum et al., 2009). According to Panigyrakis and Chatzipanagiotou (2011), an IS that is made up of state-of-the-art technology and software and has high level of accuracy, process speed, easily accessible, user friendly is the basis of an effective functional IS.

Numerous research have demonstrated that quite a number of IS implementation projects result in failure (Guinan et al., 1998; Irani et al., 2001; Kim & Kankanhalli, 2009; Yeo, 2002). Lassila and Brancheau (1999) noted that the impacts of IS implementation are often mediated either with avoidance, resistance or adaptation by users. According to Kim and Kankanhalli (2009) user resistance is ranked as the major challenge in IS implementation and this is as a result of diverse changes in technological and social systems. People aptly blame an IS when things go wrong, even when the problem is not related at all to it (Irani et al., 2001). Markus and Pfeffer (1983) shed light on why the implementation of IS often meet with some resistances. They asserted that there is greater potential for conflicts, resistance and failure if an IS substantially alter the existing power structure of an organization. In addition, where the goals or preferences of an IS do not

correspond with that of an organization, the system will encounter resistance and risk failure. More so the incongruence of the language and symbols of an IS to the organizational culture instigate resistance to a system and likely cause system failure.

IS implementations have become inevitable in contemporary business processes, and can therefore not be neglected, rather certain changes must take place in order for an implementation to be successful. Changes are vital and unavoidable component of IS implementation, therefore it is important that they are identified in advance and managed in an appropriate manner (Ćirić & Raković, 2010). In response to the changes caused by IS implementation users may resist the new IS and cause underutilization or non-use of the system (Kim & Kankanhalli, 2009). Top management support and the provision of training, guidance, time and resources to users to learn the new system is necessary to enhance organizational support of IS implementation changes (Effah & Abbeyquaye, 2013; Kim & Kankanhalli, 2009). The problem with this assertion is that whilst top management support reduces or eliminates user resistances to a new system, it can however not be ascertained whether users are actually satisfied with the performance of the system. Hence the support of top management for IS only coerces users to use the system leaving them with no option to resist the system even if it does not meet their needs. Managing changes require that special attention be given to resistance to change, organisational culture and stakeholders not neglecting the participating roles of organizational cultures and stakeholder during change management as they are crucial to the success of IS implementation (Ćirić & Raković, 2010).

2.4.3 IS in Use

Using IS result in the build of organizational capabilities (Caldeira & Ward, 2003; Effah & Abbeyquaye, 2013). IS/IT staff gain more experience and capabilities during IS use as they tend to resolve basic and complex issues regarding the system at source code level without the assistance of consultants (Effah & Abbeyquaye, 2013). The need to resolve such issues also lead to IS/IT staff participation in online forums that broaden their knowledge and understanding of the system's functionalities and intricacies, however this is more feasible for applications developed with open source tools. The knowledge gained by IT/IS staff from the use of the OSS application eliminated the need to wait for software package vendor to resolve technical issues regarding package issue, which also implies that there is minimal downtime as a result of technical error.

Abandoning and underutilizing an IS results in failure and frustration on the part of senior managers even when the system is functioning as expected (Lassila & Brancheau, 1999; Yeo, 2002). The authors then suggested that a positive initial experience should be created for users of a new IS and organizational change handled tactically as these can lead to more or less effective use of IS.

Ćirić and Raković (2010) identified two kinds of users that may cause the greatest problem in IS implementation, namely; latent opponents and potential promoters. Latent promoters generally resist change in a passive manner, they may agree to the functionalities of a system, but then insist that the system is not functioning as they expect it to. Potential promoters on the other hand, readily agree to the need for a change, but usually need to be more convinced about the benefits of the changes from the IS implementation. Irani et al. (2001) asserted that successful users measure and control the

effectiveness of IS continuously whilst properly addressing the contentious human and organisational factors.

2.4.4 IS Maintenance

User requirements for a system are dynamic and so is technological advancement which causes existing systems to be less useful and therefore creating the need for enhancing and making modifications to them or adopting new forms of technological solutions to be developed (Alves et al., 2007; Engu et al., 2012; Schneidewind, 1987; Williams & Edge, 1996). Successful IS inspires user-generated requests for change and enhancements, therefore IS maintenance is an inevitable part of its entire implementation process (Bennett & Rajlich, 2000). Software maintenance is defined as modifications made to a software after it has been delivered so that any faults would be corrected in order to improve its performance or other functionalities or to adjust it to suit the dynamic work environment (Bennett & Rajlich, 2000; Schneidewind, 1987). Software maintenance comes with its own challenges such as huge cost since it consumes a large part of the overall lifecycle cost (Bennett & Rajlich, 2000; Dekleva, 1992). More so the implementation speed of software maintenance is slow due to the fact that maintainers have to work with program codes that has little or no documentation at all (Biggerstaff, 1989; Schneidewind, 1987). According to Bennett and Rajlich (2000), organizations tend to lose a lot of business opportunities as a result of not being able to quickly and reliably change software.

2.5 Software

The concept of software has been identified earlier on in this chapter as a component of IS. Organizations have the option to develop their own IS solutions using their regularly

trained experts in order to keep up with the dynamic pace of technology or to liaise with a software vendor who will provide them with the IS solutions needed (Langat & Kamuren, 2013; Nelson et al., 1996; Nelson et al., 1998). The following sub-sections present a discussion and critical appraisal on the options available to users and organizations in the implementation of IS.

2.5.1 Packaged Software

Software packages, also known as Commercial off-the-shelf Software (COTS) are developed for sale to the general public (Sawyer, 2000) and according to Lassila and Brancheau (1999), they "provide relatively complete support for standardized business processes and come either ready to use or with a varying degree of integration" (p. 10). Software packages are standardized products intended to appeal to a large audience of users based on some assumptions about their nature of work and organizational processes (Howcroft & Light, 2010). They are usually distributed with a user guide that details its in-built functionalities. Software vendors offer extra training courses to provide guided introduction into the software or to learn how to use the software for particular tasks (Steiniger & Hay, 2009).

Using packaged software requires an organization to identify its information requirements, evaluate alternative packages and selecting the package that best suits the information requirement needs of the organization. Lassila and Brancheau (1999) summarized the benefits expected from packaged software as including lower cost of implementation; shorter implementation timeframes; and faster means of attaining project objectives. The authors however also noted that implementing software packages could disrupt the relationship among existing organizational structures and processes,

thus resulting in changes in the organizational process, as users fine-tune their work processes to suit the standardized processes supported by the software. Many organisations that implement packaged software are therefore faced with the dilema of whether to customize the software to suit the organisations' need, which is usually expensive, or to change the practices of the organization to adapt to the work flow of the software (Sawyer, 2000). In order to minimize the extent to which organizational structures and processes are disrupted, organizations tend to customize software packages, however, it is commonly argued that software packages do not allow for cost-effective customization (Khelifi et al., 2009; Stamelos et al., 2003). Apart from the cost involved in customizing packaged software, there are usually implications for organisations in terms of future maintainance and update (Brehm & Markus, 2000; Gibson et al., 1999; Sawyer, 2000). Light (2001) recommended that organisations carefully consider the sustainability of customizations made to packaged systems, as they may require maintainance in light of upgrades which can be quite cumbersome.

The complex nature of organizations in terms of professional and organizational interests and the changing patterns of relationship between them complicates the decision-making on the procurement of packaged systems. Hence Pollock and Hyysalo (2014) were motivated to conduct a study on the role of a reference actor within the development and procurement of enterprise packages. They defined a reference actor as an individual user or a network of users within and across an organization who offer their organizations' experiences as a standard for others. Using a qualitative approach and the theoretical foundation of SST, the concept of reference actor, was used to understand how the activities of users and software vendors are conjointly intertwined in the production, selling and marketing of packaged software. Findings of their study show that the

selection of a new technology is not limited to managers or certain groups within an organization, but involves intense interactions among a wide range of RSGs. Howcraft & Light, (2010) also contributed to knowledge in this area by drawing on SCOT for theoretical support together with knowledge drawn from literature to develop a framework for packaged software. The study by Howcraft and Light (2010) is very different from that of Pollock and Hyysalo (2014) in the sense that the former dwelt more on the role of a reference actor in the selection and acquisition of a packaged application within an organisation whilst the latter seek to understand the role of the reference actor in shaping the kind of product that a packaged application vendor offers for sale.

Lucas Jr et al. (1988) argued that the measurement of the performance of packaged software by users is strongly linked to their former use of computers in an organization, thus users with prior use of computers usually have a certain level of expectation from new IS that tend to influence the measure of its performance. The authors therefore noted that in order for the implementation of software packages to be successful, managers should rate the skills of users; vendors should also rate the customer capabilities and should actively participate in the installation of the software and provide necessary support. Lassila and Brancheau (1999) also emphasized that for the implementation and assimilation of packaged software to be successful, there is the need for an understanding and effective management of the association between IS and the organizational context. In the same vein, Magni et al. (2012) expressed that users typically use only 20% of the features found in technologies, thus Ke et al. (2012) in their study, drew on the Self Determination Theory to examine factors that motivate users to deeply explore the features of enterprise systems. They further went on to suggest how an organization can

harness it's organizational lever to cultivate intrinsic motivation among users that could ultimately lead to the successful implementations of IS.

ERP systems are generic off-the-shelf software packages that are intended to cover the fullest range of organizational activities and processes (Fowler & Gilfillan, 2003). According to Salimi and Dankbaar (2008), ERP software packages demand came about as a result of IS department's inability to cope with systems integration. The authors noted that most of the literarture on ERP implementation assumes that top management drives the implementation process, but they argued that the ITD, software developers and users from other functional departments play a crucial role in the implementation process. Organizations that adopt ERP systems, do so with the hope of gaining economic benefits from commodified solutions; benefiting from the interoperability of standard platforms; and aligning with best practices in their line of work. However because of the too-generic nature of the ERP packages, a wide gap is created between the standardised generic solutions and the specific contexts, practices and requirements of the organizations which tend to make ERP systems rather more costly and risky (Pollock & Hyysalo, 2014; Pollock et al., 2003; Wagner et al., 2006). Software vendors tend to make money from the periodic upgrades and updates they make to their products, thus making software packages expensive, thus vendor controls the cost of packaged software by determining what organizations have to pay and when (Khelifi et al., 2009). Temizkan et al. (2012) contributed to policy in IS, by studying the patch release behaviour of software vendors and suggesting the establishment of legislations that will influence software vendors to release patches that benefit clients in an optimal manner. The authors established this by developing a model drawn on a cost-based theory to study the different types

vulnerablities in software and their impact on the vendor's integrity, which according to them is governed by legislation.

This study is intended to understand the development of bespoke software in higher education institutions, however it is important to understand the choices available in the implementation of IS. Below is a tabulated appraisal of packaged systems.

Table 3 - Critical Appraisal of Packaged Systems

Advantages	Disadvantages
1. Organizations can take advantage of sophisticated and thoroughly tested software based on best practices	**1.** Cost of implementation and maintenance could be substantially higher than that of bespoke software
2. Implementation of packaged software takes significantly less time than bespoke software.	**2.** Organizations incur a certain level of dependency on the technological direction of the vendor company
3. Packaged software are usually configurable which creates an environment for future exploitation without the need to build new functionalities.	**3.** Packaged software usually requires major customization to accommodate processes that give the organization a competitive edge
4. Greater degree of compatibility and integration with other software	**4.** Organizations have to rely on vendor for technical issues that are encountered.
5. User support and training are usually readily available through professional support groups and well documented manuals.	**5.** Software vendor retains exclusive rights to source codes, making modifications difficult
6. Upgrade and updates to the software is guaranteed	**6.** Usually requires the purchase of vendor recommended hardware

2.5.2 Bespoke Software

Bespoke software, also referred to as tailor-made software or custom software is one that is specifically developed for a user or organization based on its own requirements (Gu et

al., 2010). Bespoke applications are either developed by an organisation's internal IS staff or outsourced to a software development company (Sawyer, 2000; Wang et al., 1997). The drive to develop a bespoke solution usually stems up from an organization's need for an IS that acknowledges their idosyncracies and recognizes and addressess human issues that arise in relation to the software (Irani et al., 2001).

Table 4 - Critical Appraisal of Bespoke Systems

Advantages	Disadvantages
1. Designed to fit exactly to a business requirement of an organisation.	**1.** Development process for bespoke systems take a long time
2. Organisation has full ownership of the software and its source code as well as the knowledge acquired in the development process.	**2.** Greater chance of hidden bugs since software have not been widely tested and have no proven track record
3. Making enhancements to the software is easy as the development team are closer to or within the organization	**3.** It usually cost more to develop a bespoke system, because all costs of development are borne by a single organization
4. The relationship that exists between the development team and the users facilitates communication and expectation delivery.	**4.** There is the lack of professional support
5. Can differentiate an organization from its competitors and provide it with a competitive advantage	**5.** Updates to the software are made erratically
6. There is a much higher quality of support for software, since organization deals directly with the development team.	**6.** Limited Best Practices
7. More Intuitive and user friendly	
8. In most cases, less training is required, because the software is designed to work exactly how the organisation wants it work	

Previous studies concerning bespoke software have covered a wide range of issues including requirement engineering (RE) process; development cost; and its comparison with packaged software (Engu et al., 2012; Jebreen et al., 2014; Stamelos et al., 2003).

Kirkup and Kirkwood (2005) indicated that higher education institutions usually fall on bespoke software to appropriate the use of Information and Communication Technologies (ICT) to facilitate teaching and learning activities among lecturers and students. Bespoke software is also used in higher education institutions to find solutions to the inefficiencies of packaged systems. For instance, data from an electronic attendance monitoring system, Uni-Nanny piloted in two schools at the university of Glamorgan, had to be downloaded at the end of each day, into a bespoke software before the data can be queried to produce reports (Bowen et al., 2005).

Extant research have been conducted in the area of bespoke software, however little study have been made on bespoke software development in higher education environments. Based on the established research gap, this paper is aimed at understanding the rationale behind the development of bespoke software by higher education institutions, how the social composition of these institutions shape the development and the consequences of implementing bespoke systems in the institutions.

2.5.3 Open Source Software (OSS)

OSS refers to software systems that are free to use and whose source code is freely made available to the public to be used as it is and/or modified without the need to obtain a license (Khelifi et al., 2009; Kumar et al., 2011; Von Hippel, 2001). Most OSS systems emerged from an attempt by a developer to provide solutions to problems that he/she faced, thereafter making the solution available to others for free (Von Hippel, 2001). Since the source codes of OSS are freely accessible, the practice of software development from OSS has evolved, thus organizations that hope to acknowledge the idiosyncrasies of their companies' IS needs, settle on this (Irani et al., 2001; Von Hippel, 2001).

According to Khelifi et al. (2009), IS literature claims that IT decision makers value how OSS reduce their dependency on software vendors more than the cost savings, but the authors argue that in many developing countries OSS products are considered as feasible alternative to packaged solutions due to the limited budget that is normally allocated to IT solutions.

Several OSS products provide significant service to higher education institutions. The likes of the Linux operating system, the Apache web server and MySQL database form the basis for which software solutions have been built not only in the higher education institutions but also in other industries. Courant and Griffiths (2006) pointed out that if universities and colleges pool resources to build an open source software and their collective efforts attract an extensive base of contributors, the resulting application would meet their needs and potentially be beneficial to the broader community.

Many open source programs have achieved a level of maturity making them an option that can compete effectively in the marketplace with their proprietary equivalent so that users are able to decide among equivalent alternatives (Khelifi et al., 2009). The internet and rapid technological advances in computer hardware and software and networking technologies have made it much easier to sustain a communal development of software (Von Krogh & Von Hippel, 2003).

2.6 IS and Organizational Context

It is commonly reported in literature that in order for an IS project to be successful, it is important for IS and the organizational context to be adapted during implementation (Alenljung & Persson, 2008; Alter, 2003; Boahene & Bowles, 1999; Gibson et al., 1999;

Irani et al., 2001; Lassila & Brancheau, 1999). According to Irani et al. (2001) organizations experience failure with IS because user expectations are not met. This implies that consideration of interpersonal, cultural and organizational aspects of work environments boil down to the requirement gathering stage of IS implementation as Saiedian and Dale (2000) noted that the success of the requirement engineering process, which form the basis of successful IS implementation, depend not only on technical issues but also upon how well people communicate and work together. This consequently led to the need for involving and enfranchising organizations and their staff in IS development and implementation.

IS and the organization they support are complex (Von Alan et al., 2004), thus Alter (2003) noted that a mismatch between an IS and organizational structure will result in resistance during implementation and cause continuing conflict when the system is operational, although some IS are implemented to deliberately cause a change in the organizational culture. Mengesha (2010) argued that the IS development and implementation is harnessed by interpretations, behaviors and actions of the stakeholder groups, management style and the political processes of an organization. Nasir (2005) noted that IS literature in the category of organizational structure have discussed issues such as the advantages and disadvantages of structural adjustments caused by IS implementation in organizations and the problems of information access due to organizational structure. For instance, Markus and Pfeffer (1983) revealed several ways in which IS are related to organizational power. They stated that since information system is used in decision-making, it implies that those who have access to and control the flow of information have power within the organization. Again, the authors noted that the use of IS to confer legitimacy and restrictions on staff, usually indicate the level of power or

authority of staff in an organization. Also, because IS are used to change the performance of individuals and ultimately the outcome of organizational procedures, the use of these systems entails the exercise of power. The authors also indicated that the use of IS to make decisions, change organizational performance and grant access can affect how power is distributed within an organization and this can at times lead to interpersonal conflict because those who are affected are likely to fight against the effected changes. Thus, a study by McGrath (2003) that sought to critically analyze the mutual effect of organizational culture and IS implementation on each other, established that, where IS implementation must be accomplished in the face of cultural change in an organization, a balance must be negotiated between the improvisation of working methods and the enforcement of the use of the IS.

2.7 Chapter Summary

This chapter aimed at finding out the most explored issues and the most used theories in the area of bespoke and packaged software development to know the areas in information system that require more research. The findings from the review suggested the need for more studies to be conducted away from the adoption and implementation of packaged proprietary applications in higher education institutions, which makes up the bulk of researches in this area, to the development and implementation of bespoke HEIS. Based on the discussion of this literature review, this study identified and chose to understand why higher education institutions within the context of sub-Saharan Africa opt for bespoke developed information systems instead of packaged proprietary software and since higher education institutions differ greatly in their social constituents and information needs, like any other type of organisations, their information system functionalities and consequently the requirements will be informed greatly by the social

groups within individual institutions, thus the choice of the SCOT approach as the theoretical foundation of this study. The next chapter presents an overview of the SCOT theory and how it applies to this study.

CHAPTER THREE

THEORETICAL FOUNDATION

3.1 Introduction

The preceding chapter delved into relevant literature to identify gaps which this study is going to fill. This research is premised on SCOT, hence this chapter gives a brief history of the SCOT approach; its basic tenets, the reason behind its use and its application to find answers to the research questions of this study.

3.2 Brief History of SCOT

The SCOT approach was first described by British Physicist and Sociologist Trevor Pinch and Dutch Engineer Wiebe Bijker and was later developed by Bijker (1997). It argues that both social and technical features are necessary to describe the development of a technology, hence, technology is socially shaped or constructed, that is, its properties are determined largely by the interpretive frameworks and negotiations of RSGs (Brey, 1997). This is supported by the claim of Williams and Edge (1996) that technology is a social product shaped by the conditions of its creation and use, thus at every stage of the creation and implementation of a new technology, a number of social factors that affect which technical option is selected for the technology is considered. SCOT is a theory for technological development, and its basic premise is that technologies emerge from social interactions among social groups and actors (Prell, 2009) , and technical artefacts are described through the eyes of the members of these groups (Bijker, 2001). It is important in understanding technologies and change in organizations, as well as implementation and use of IS.

3.3 Basic Concepts of SCOT

Technologies and the social context in which they operate cannot be treated as separate phenomena which implies that in defining technology, the social setting within which it began and becomes rooted must be incorporated (Williams & Edge, 1996). In view of this, the basic concepts SCOT is discussed in order to enrich understanding on bespoke software development in higher education institutions in sub-Saharan Africa.

3.3.1 Relevant Social Groups

The concept of RSGs is the first step in SCOT analysis. RSGs and actors involved in the development of the technology in question are identified. A relevant social group is a collective entity sharing the same set of meanings about a specific artefact (Kline & Pinch, 1996) and may be "used to denote institutions and organizations as well as organized and unorganized groups of individuals" (Pinch & Bijker, 1984). RSGs are emergent but are not always easy to identify, therefore it is possible for the views of a smaller group not to be embodied or determined, also powerful group could take actions that excludes the interest of other groups (McGrath, 2003). A multi-group design process can face many problems as a result of different interpretations that lead to conflicting images of the technological artefact, because these groups may have different motivation for use, power in the organizational structure, educational backgrounds, occupational culture and political and social interests. This study is interested in identifying the RSGs within higher education institutions that are capable of influencing the development, implementation and use of an information system (Sahay & Robey, 1996). The social groups represented may include system analysts, programmers, policy makers, system administrators, technician and users (Dutton et al., 2004; Sahay & Robey, 1996) and they all differ from one another in various ways, thus leading them to develop different

interpretations of the system. With the continual interactions of the interdependent social groups through using the IS, they tend to shape its use and outcome.

3.3.2 Technological Frame

Technological frame constitute the basic concepts of SCOT and it is the set of meanings, interpretations, assumptions, expectations and knowledge people use to understand technology (Kyriakidou & Venters, 2007) . Orlikowski and Gash (1994) used the definition borrowed from cognitive biology and coined by Gioia and Sims (1986) as " a built up repertoire of tacit knowledge that is used to impose structure upon and impact meaning to otherwise ambiguous social and situational information to facilitate understanding". Bijker (1993) also referred to it as a repository of knowledge, cultural values, goals, practices, and exemplary artefacts shared by a social group, and structures their attributions of meaning to objects and processes in technical innovation, and their subsequent actions. Technological frames provide a valuable lens through which interpretive flexibilities of the RSGs can be explained and understood. Stakeholders draw on specific technological frame to make sense of systems which ultimately shape their perceptions and subsequent actions (Mengesha, 2010). Orlikowski and Gash (1994) argued that in order for people to interact with a technology, they have to make sense of it as a result they tend to develop their own expectations, assumptions and knowledge of the technology which influence their interactions with it. The authors identified three domains that social groups interpret in order to understand a technology; they are the nature of the technology which refers to people's image of it and understanding of what it has to offer and what it can do, the technology strategy, which refers to the reason behind the choice and implementation of that technology as well as the value it has to offer the organization and the third is, the technology in use, which also refers to people's

understanding of how the technology will be used and the likely or actual conditions and consequences associated with its use. Technological frame does not only include understanding the nature and role of a technology, but also its specific conditions, use and implications in a given setting, therefore people within the same technological frame can have different understandings of a particular system when placed in different context with different conditions applied. Difference in the technological frames of RSGs of an organization, such as management, technical team, and users always give rise to difficulties and conflicts in the development and use of the technology or even a change in technology (Orlikowski & Gash, 1994).

3.3.3 Interpretive Flexibility

Another concept of SCOT is interpretive flexibility which establishes that "the same artefact can mean different things for different social groups of users" (Kline & Pinch, 1996). Sahay and Robey (1996) referred to interpretive flexibility as the capacity of the information system in question to sustain the different interpretations of the multiple relevant groups. The interpretive flexibility concept of SCOT shows that neither an artefact's identity, nor its technical "success" or "failure" are intrinsic properties of the artefact but are contingent on the meanings given to it by competing social groups (Dafoe, 2005). Depending on a technology's physical properties, the users' knowledge, perception and skills about it and the context in which users interact with the technology, the technology can be deemed to be changeable, thus the degree to which users' perceive a technology as changeable is referred to as interpretive flexibility (Pozzebon, 2001). The different meanings and perception held by the different social groups may also lead to disputes about who is right and wrong, however Meyer and Schulz-Schaeffer (2006) argue that this stance holds true for any object or artefact whose meaning has already not

be established, not necessarily technological artefact. The authors therefore advocated for interpretive flexibility of any technological artefact to be concerned with its usefulness, which was defined as "the ability to produce desired effects sufficiently, reliably and in a repeatable way which would not be possible or would require more efforts without the artefact". Meyer and Schulz-Schaeffer (2006) also noted that interpretive flexibility plays a minor role in the development of a technological artefact that is to replace or enhance an already existing artefact, because it is already well known who the users of the artefact were going to be, how and for which purpose they will use it and the context which they will use it. This implies that the development and implementation of a technological artefact is not based solely on the technical issues, as interpretive flexibility allows for the consideration of a range of social factors (Jakku et al., 2004; Williams & Edge, 1996). Since HEIS consist of various modules that are subject to different interpretations by the many social groups within the institution, the SCOT approach is best suited to understand the social construction of IS.

3.3.4 Stabilization

Stabilization refers to the degree to which the conflicts of understanding of a technology are resolved among RSGs and extent of acceptance of the technology by the groups (Effah & Abbeyquaye, 2013; Prell, 2009). This definition of stabilization in other words mean that the problems that RSGs have about a technology may not necessarily be solved, but a relative stability can be achieved based on some mutual understanding among them. Bloomfield and Danieli (1995) used their research on a case of a Hospital Information System, named MANEX to improve our understanding on this point. Doctors were not able to obtain instant laboratory reports in real time, as such considered MANEX not useful, however when persuaded of future plans to include their

expectations of the system into MANEX, they gave up their demands in return of future promised capabilities. Stabilization also means the convergence of the different interpretations into shared view of the technology (Pinch & Bijker, 1984). It is a process which occurs at different times for different RSGs. Humphreys (2005) criticized the definitions of closure and stabilization by Pinch & Bijker as too limited to show the true picture of how technologies relate to each other and how they contribute to changes in technologies over time, however he affirms that the definition is useful in understanding the innovation of a particular artefact. He also identified three components within stabilization, that is, the structure of the technological artefact, its use and its language and then threw more light on the difference between stabilization and closure which is; closure is about the RSG whilst stabilization is all about the technological artefact.

3.3.5 Closure

The interpretative flexibility of a higher education institutions diminishes as RSGs interact and this is termed as closure (Prell, 2009). Closure is a basic concept of the SCOT approach that occurs between RSGs and can be attained either by the groups reaching a consensus of an artefact's meaning or positioning the technology to be the solution of a problem. It is the stage at which interpretive flexibility vanishes, that is, where various RSGs accept that their problems with the technology in question will be resolved (Pinch & Bijker, 1984). In the closure process the different RSGs seek to establish their own interpretations of a technological artefact as the most convincing and therefore attempt to influence the interpretations of other social groups, however a certain interpretation becomes more and broadly accepted by the other social groups which ultimately leads to the technology artefact to be seen as the appropriate solution to their common problem (Meyer & Schulz-Schaeffer, 2006). A technological artefact is said to have reached

closure if it is no longer open to changes and modifications. SCOT has however been criticized for its inability to easily account for closure because the possibilities of interpretive flexibilities seems unending (Williams & Edge, 1996).

3.4 Examples of Studies that used the SCOT approach as a theoretical foundation

The studies that have adopted the SCOT have taken a variety of forms. For example, Howcraft and Light (2010) augmented the SCOT approach with a political perspective to account for how the social actors within organisations can shape the selection process of packaged software. The researchers using an in-depth longitudinal study expanded their analysis beyond the organizational level and contact with end user to the wider market forces and relevant social actors such as the software supplier and vendors, IT consultants and industry analysis to understand a technology and its adoption. Sahay and Robey (1996) used the interpretive approcah with SCOT as the theoretical stand to demonstrate the importance of social interpretation to understand the implementation and consequences of a Geographic Information System (GIS) that was implemented in two local goverments organizations in the United States. Prell (2009) based her analysis of Connected Kids – an online IS intented for young people - within the SCOT context to interprete the structural issues relating to effect of the role of actors and access to resources on the development of a technology. Kleinman and Klein (2002) also summarized the SCOT conceptual framework and presented concepts from organizational sociology and political economy to illustrate the influence of social structures in shaping the technology.

3.5 Application of SCOT in this Study

The SCOT has gone through several critiques, reformulations and subsequent evolution, but it is still considered useful for investigating the development of bespoke information system, because of its capability to study complex technological innovations and multiple stakeholders' views and how they shape or are shaped during the process (Thoreau, 2010; Winner, 1993). Since SCOT, is based on the premise that social groups interpret artefacts differently and seek to shape them according to their different systems of meaning (Kleinman & Klein, 2002; Pinch & Bijker, 1984), it provides an appropriate theoretical lens to view this study, not only that, "the foundation of social construction provides a useful platform for interpretive studies of information technology" (Sahay & Robey, 1996). Again, SCOT is used in this study to explain how the identified different social groups and technological innovations within the case institutions selected for this study, shape the development, implementation and use of bespoke software (Effah & Abbeyquaye, 2013; Prell, 2009). More so since this study seeks to find answers to why higher education institutions in sub-Saharan Africa opt for bespoke student information system software, SCOT offers a more helpful approach for understanding the innovation of the systems (Humphreys, 2005). Again because this study is aimed at exploring the consequences with the use of an in-house bespoke developed information system in higher education institutions, SCOT remains a useful approach to gaining knowledge in this regard, because it provides a detailed insight into the ways in which different social groups make sense of and relate with new technologies.

3.6 Chapter Summary

This chapter discussed the SCOT approach as the analytical lens for this study. It began by giving a brief history of the SCOT approach and further discussed the basic tenets of

the theory. The chapter ended by discussing the application of the SCOT theoretical foundation to the study. The next chapter discusses the method used to collect data that answers the research questions of this study.

CHAPTER FOUR

RESEARCH METHODOLOGY

4.1 Introduction

The previous chapter discussed SCOT and its applicability to this study. This chapter focuses on the methodology used in conducting the study. It starts by explaining the dominant paradigms in qualitative IS research and then goes on to choose the interpretive paradigm and justifies why it is suitable for this study. Furthermore, the research methodology, research method, case selection and the field work of this study are discussed. The chapter then discusses the methods of data collection and analysis used.

4.2 Research Paradigms in Information Systems

Paradigms are "integrated set of assumptions, beliefs, models of doing good and techniques for gathering and analyzing data" (Neuman & Robson, 2004). Mertens (2010) acknowledged that paradigms guide the researcher in identifying and clarifying his/her beliefs with regard to reality, methodology, knowledge and ethics. Paradigms are constituted by three main perspectives; Ontology, Epistemology and Methodology. The ontological stance of paradigms focuses on the objectivity or subjectivity of reality and its characteristics (Creswell, 2012). The epistemological perspective looks at the nature of knowledge and how it can be applied to determine whether it is objective or subjective (Ritchie et al., 2013) . The methodological stance of paradigms is associated more with the methods and tools used in data collection, analysis and drawing conclusions when conducting a research. This section gives an overview of the three main paradigms that exist (positivist, critical & interpretive) in the general disciplines of IS research, following the Orlikowski and Baroudi (1991) classification.

4.2.1 The Positivist Paradigm

The positivistic researchers' aim is to explain and ultimately enable the prediction and control of a phenomena. Research is conducted in a deductive manner to discover causal relationships that form the basis of generalizing the knowledge acquired (Paré, 2001). The ontological perspective of positivism assumes reality exists and is objective due to some form of a general cause-and-effect law (Guba & Lincoln, 1994). This is to say that, irrespective of the researcher, and the object or phenomenon being researched, there is an objective truth of the reality that researchers can precisely measure and use statistics to test (Neuman & Robson, 2004). The epistemological stance of the positivist is that knowledge can be replicated, therefore when different studies are conducted on the same phenomena by different researchers at different times, the same results will be recorded (Guba & Lincoln, 1994). This stance of the positivist also argues that researchers play passive neutral roles in the phenomenon of interest, in order not to pose a problem to the validity of the findings of a study (Orlikowski & Baroudi, 1991; Sahay & Robey, 1996). Positivists usually adopt the use of laboratory experiments, field experiments and surveys. In order to identify underlying regularities, they collect large data which is statistically analyzed, so that ultimately research questions and hypotheses can be validated or falsified (Weber, 2004).

4.2.2 The Interpretive Paradigm

The Interpretive paradigm posits that truth about a phenomenon depends on the context in which the phenomenon is being observed. Researchers in this field try to understand the way others interpret, view and understand events, partly because they assume that these events or concepts influence the behaviors of the individuals (Kaplan & Duchon, 1988). The ontological underpinnings of the interpretive paradigm embraces multiple

realities and explains the social world from the viewpoint of organizational actors that are directly involved in the construction of reality (Creswell, 2012; Goldkuhl, 2012; Hirschheim & Klein, 1989). Interpretivism supports that the researcher cannot be detached from the phenomena since it involves the researcher subjectively understanding a phenomenon. Perception people have about the real world is dependent on their experience as such the real world and its observer cannot be detached. The epistemological stance of the interpretive paradigm posits that knowledge of reality is based on social construction by human factors and that people assign meanings and values to their unique context and attempt to explore the phenomenon of interest in its natural settings, without imposing any previous understanding on it (Andrade, 2009; Kroeze, 2011; Orlikowski & Baroudi, 1991; Walsham, 1995a). Goldkuhl (2012), notes that interpretivism seeks to work with the subjective meanings that already exists in the social world in order to reconstruct them, understand them, avoiding distorting them, and use them as a building block for theorizing. The interpretations of realities in interpretive research are likely to shift over as its constituents and circumstances change. This is as a result of the transformation, transfer and usage of the meanings. Interpretive researchers employ qualitative methods in finding answers to their research questions. They tend to use case studies, ethnographic studies, phenomenology as preferred research methods (Weber, 2004).

4.2.3 The Critical Paradigm

The critical paradigm is proposed as means by which researchers identify oppositions, contradictions and conflicts that exist in societies and organizations as a result of structural problems, so that the causes of these variations can be accounted for and possibly eliminated (Orlikowski & Baroudi, 1991; Wynn Jr & Williams, 2012).

Researches conducted from the critical perspective is used to develop in-depth explanations and descriptions for the outcomes of specific social phenomena by taking into account factors such as the breadth of information technology, social constituents, and the organization which may have played a causal role in their occurrence (Wynn Jr & Williams, 2012). Researchers in this field do not only conduct studies and propose theories but they actively effect changes in the phenomenon under study. Ontologically, the critical paradigm acknowledges the existence of an independent reality. According to Wynn Jr and Williams (2012), critical research assumes that social reality is constituted and constructed by people and that despite the deliberate effort to change their social and economic circumstances, they are normally limited by social, cultural and political powers. Epistemological stance of the critical paradigm posits that knowledge is grounded in social and historical practices, thus a whole phenomenon cannot be studied in isolation of its society and context within which it operates, because of the possibilities of influences of historical, economical, social and political conditions on the phenomenon (Orlikowski & Baroudi, 2001). Critical researchers in this field tend to adopt long-term historical and ethnographic studies of organizational processes and structures (Orlikowski & Baroudi, 2001). The methodologies associated with this research offers the researcher new opportunities to investigate complex organizational phenomenon in a holistic manner by providing more detailed causal explanations of the phenomenon (Wynn Jr & Williams, 2012).

4.3 Choice of interpretive Paradigm

Interpretive research is useful when the research aims to gain deep insight into IS phenomenon, including its development and management (Klein & Myers, 1999). Since this study aims to understand why and how higher education institutions in sub-Saharan

Africa opt for bespoke software development and the consequences of their choices, and Howcraft and Light (2010) assert that interpretive research helps to gain more insight into the subjective interpretations of members of the RSGs within institutions, this study is based on an interpretivist perspective. Interpretive research requires the understanding of how practices and meanings are formed in order to understand social reality, thus the need of this stance to construct the explanations that accounted for the choice of bespoke applications and its development over packaged software (Orlikowski & Baroudi, 2001).

4.4 Research Methodology

Research can be conducted using any one of the following methodologies namely, quantitative, qualitative and the mixed-method approach (Creswell, 2012, 2013). The quantitative approach is a highly structured research methodology that is used for testing objective theories by examining how variables of the phenomenon relate (Creswell, 2013). In order to ensure reliability, generalizability and replication of research findings, quantitative researchers follow strictly, from start to finish, the deductive procedures of the methodology (Chen, 2011). It is characterized by positivism, objectivity, scientific methods and experiments and data is collected in the form of numbers using predetermined instruments (Creswell, 2012). According to Johnson and Onwuegbuzie (2004), quantitative research is focused mainly on deducing, confirming, testing of hypothesis and theories, explaining, predicting and analyzing statistics. Researchers using this approach employ inquiry strategies such as experiments and surveys, and are involved in what is being researched, or bring their own personal interests and values into the research. The Quantitative approach is suitable for phenomena where variables are clearly defined and the phenomena, although very complex, can be broken down and quantified (Chen, 2011).

Corbin and Strauss (2008) define qualitative research as one that finds answers to research questions without employing any statistical procedures. The Qualitative approach is appropriate for exploring and understanding the meanings that individuals or groups give to a phenomena as well as obtain details such as feelings, thoughts, processes and emotions that participants of a research ascribe to a phenomena which otherwise may be difficult to obtain by the other research methodologies (Corbin & Strauss, 2008; Creswell, 2013). The researcher conducts his research in a natural setting and collects data in the form of words through interviews, observation, and document analysis using strategies of inquiry such as action research, grounded theory, ethnography and case study (Creswell, 2012). Qualitative research is also characterized by subjectivity, that is, the data collected for the purpose of the research directly involves the researcher and the findings of the research is also influenced by the interpretations of the researcher. Johnson and Onwuegbuzie (2004) identified the major characteristics of this approach as induction, discovery, exploration, hypothesis or theory generation, the researcher being the primary tool for data collection and the analysis of textual data.

Mixed method research, as defined by Johnson & Onwuegbuzie (2004), is a class of research where the researcher combines the qualitative and quantitative research techniques, methods, approaches and strategies into a single study. According to them, when researchers gain a deep understanding of the strengths and weaknesses of the qualitative and quantitative approaches, they can combine these approaches effectively to result in complementary strengths and no overlapping weaknesses. Cronholm and Hjalmarsson (2011) note that this approach is aimed at obtaining findings that are more trustworthy and (Creswell, 2013) complements this fact by also acknowledging that the combination of qualitative and quantitative approaches is to provide a more wholistic

understanding of a phenomena than either of the approaches. The inquiry assumption behind this mixed method approach is based on the fact that collecting both numeric and textual data, either simultaneously or sequentially will best provide understanding to the research problem (Creswell, 2013; Kaplan & Duchon, 1988).

The choice of research methodology is guided by the paradigm adopted for this study and since qualitative strategies emphasize an interpretive approach to find answers to research question (Kaplan & Duchon, 1988), the qualitative research is therefore considered the most suitable methodology for this study. Since this study, viewed through the SCOT theoretical foundation is aimed at understanding the why and how of the choice and development of bespoke software instead of packaged proprietary software by higher education institutions in sub-Saharan Africa and the consequences of their choices, the qualitative approach is especially helpful as it is directed at providing an in-depth interpreted understanding of the social groups that exist within the case institutions selected for this study by learning about their social and material circumstances, experiences, perspectives and history that consequently shapes the development of the in-house information systems (Ritchie et al., 2013).

4.5 Research Method

Since this study is based on a qualitative approach, it is just natural for the study to adopt a qualitative research design. Basic designs in qualitative research include phenomenology, ethnography, grounded theory and case study. Phenomenology is a method used to uncover and understand “the constructs that people use in everyday life to make sense of their world” (Ritchie et al., 2013). It is based on the assumption that there is an essence to shared experience and therefore seeks to understand the

phenomenon of a lived experience. Ethnography seeks to understand the social world of the people being studied by allowing the researcher to live in their community in order to produce a detailed description of the people, their culture and beliefs (Ritchie et al., 2013). This method is used to generate insights that other forms of research cannot because, in this case the researcher observes the participants' behavior in their natural context. Ethnographers normally use methods such as participant observation, unstructured interviews and document analysis to collect data. Grounded theory, through the correct application of a systemic process, provides the researcher with the opportunity to theorize from evidence that exists in data (Andrade, 2009). According to Myers and Avison (2002), grounded theory suggests that the data collection and analysis stages of a study should be conducted simultaneously as it is useful for developing context-based, process oriented explanations of the phenomena. Qualitative case study is used to study complex phenomena within their context using a variety of data gathered from different sources to illuminate the case (Baxter & Jack, 2008). The case study enables the researcher to study socio-cultural phenomena, capture reality in greater detail and to understand people and the context in which they live (Kyriakidou & Venters, 2007). Ritchie et al. (2013) noted that case study is particularly helpful in gaining holistic, contextualized and comprehensive understanding where a full account of the phenomena under study cannot be gained through a single perspective. According to Yin (2003), case study is a suitable method when the study seeks to answer 'why' and 'how' questions that the researcher has little or no control over.

Case study can be taken from a positivist, critical or an interpretive paradigm and may involve single or multiple cases within a single study. Interpretive case studies allow the researcher to be directly involved with the data collection and analysis of the research

and thus makes it possible for the researcher's own construction as well as the participants of the research to be presented (Guba & Lincoln, 1994; Klein & Myers, 1999; Neuman & Robson, 2004). Andrade (2009) also notes that the interpretive case study is used to create an integral and persuasive explanation about a phenomenon that takes into account each participant's different perspectives. The critical case study research involves critical reflections on current practices, assumptions and questions taken for granted and it critiques the status quo based on the theories of one or more critical theorists. Words, such as 'validity' and 'reliability' which imply an objective reality independent of social reality are not normally used in critical studies. Case study research within the positivist perspective is designed and evaluated based on the natural science model of research, that is, controlled observations, controlled deductions, replicability and generalizability and used to generate and/or test theory (Darke et al., 1998; Lee, 1989).

A single case study is suitable when the researcher wants to understand a unique case, whereas the multiple case study allows the researcher to understand the differences and similarities between cases. Multiple case study is aimed at replicating findings across cases and to investigate a particular phenomenon in diverse settings, thus the multiple cases are used to predict similar or contrasting results (Darke et al., 1998; Walsham, 1995b; Yin, 2003). Carlsson (2005) also notes that a researcher can accumulate supporting evidence through multiple case-studies so as to obtain 'theoretical saturation'.

Given the interpretive stance of this study and its goal to understand the reason behind the choice of the development of a bespoke software by higher education institutions and

how they are implemented, the interpretive case study involving multiple cases is considered more appropriate for this study.

4.6 Fieldwork and Case Selection

All fieldworks concerning this study were conducted in Ghana. The selection of cases and gathering of data occurred within the period of September 2014 to March 2015. Since organizations within the same industry have common characteristics as well as distinct ones (Hekkala & Urquhart, 2013), two higher education institutions in the country were selected for the purpose of this study through theoretical sampling that was based on the research purpose and research questions. The Universities selected as cases for study were the University of Professional Studies, Accra (UPSA) and Valley View University (VVU). In order to be selected, these institutions were identified to be higher education institutions in sub-Saharan Africa, using in-house bespoke IS. The selection process began with a discussion between the researcher and the researcher's lead supervisor to identify higher education institutions that have implemented bespoke IS. The researcher, in order to trace their choice and development of bespoke systems, then decided to select the two case Universities based on easy access to them, close proximity and the fact that supporting evidences that can be used to identify similar or contrasting results and to replicate findings can be accumulated from this cases. Formal letters from the researcher's department from the University of Ghana signed by the Head of Department introducing her as a student researcher was sent to the authorities of these Universities to enable her gain access to the information needed to answer the research questions of the study. Furthermore, the researcher's lead supervisor introduced her to two of his students, one of whom was a formal staff in one of the case institutions selected and had spear-headed the bespoke software development, the other, a staff of the same institution. The

researcher contacted this fellows through mobile phone and consequently arranged face to face meetings to obtain more assistance on gathering data. Through snowballing, the researcher also came in contact with other people who ultimately assisted her in gaining access to the case institutions after knowing the purpose of the research. The following section describes the sources of data collection for this study.

4.7 Data Collection

Data collection is not simply gathering data for later analysis, but tied to the research questions and the epistemological stance of the study (Chism et al., 2010). Data for the research was collected from multiple sources including informal discussions, observations and semi-structured interviews. Face-to-face interviews were conducted with participants of the cases who were significantly involved in the bespoke software implementation. Participants of the interview included individuals with diverse roles and background within the institutions (Liu et al., 2011). A semi-structured set of guidelines, designed based on the research purpose and questions of this study, were used to conduct the interview in order to identify the various concepts and themes related to this study. Each interview lasted between 45 and 90 minutes and was recorded on tape after gaining the consent of the participants. Relevant data was also gathered through observation and informal discussions with staff of the Universities. Observation was used to provide a view of the research settings that was unfiltered by those being observed (Chism et al., 2010).

4.8 Data Analysis

Data collected were analyzed at different levels. Themes, such as events, issues and resolution of issues were identified and data was organized according to these themes (Elbana, 2013). Interviews, field notes and observations recorded during the data collection stage, were read thoroughly, over and over again, in order to develop a clear connection between and among concepts and themes (Orlikowski & Baroudi, 1991; Walsham, 1995c). More data were gathered from the field as themes emerged and concepts were identified to help shape consecutive data analysis (Baxter & Jack, 2008).

4.9 Chapter Summary

This chapter discussed the methodology used for conducting this research. The chapter started by discussing the interpretive research paradigm and went on to explain why it is an ideal choice for this study. The chapter also discussed the choice of the interpretive case study as the appropriate method for carrying out this study and further went on to discuss the criteria used for selecting the cases for this study. The data collection methods and how data will be analyzed in this study were also discussed. The next chapter presents the findings of this study using the research questions posed.

CHAPTER FIVE

CASE FINDINGS

5.1 Introduction

The previous chapter discussed interpretivism as the research paradigm for this study and further discussed the interpretive case study as its method. The selection of cases and how the fieldwork was conducted were also discussed. This chapter presents the case study findings as it provides a description and analysis of each case. For each case description, the chapter presents background on the existing information system and what led to the choice of a new IS. It also presents how the development of the new system was done for each case and the consequences of using the newly bespoke software. The chapter then concludes by giving a summary on the findings and introducing the reader to the next chapter that presents analysis of the case findings.

5.2 UPSA Case Description

This section describes the findings from the UPSA case study. It begins by providing an overview of the University which includes when the University was founded, its vision and missions, its transition from a private institution to a full-fledged government institution. This section then goes on to describe the challenges faced with Online Students' Information System (OSIS), the existing IS which was in place and how management of the University arrived at the decision to develop a bespoke IS. Furthermore, a comprehensive description of the bespoke software development process is given. The section concludes by providing details on the consequences of developing a bespoke system in the University.

5.2.1 Background of UPSA

UPSA was founded in 1965 as a private professional business education provider. The then Institute of Professional Studies was later taken over by the Government of Ghana in 1978 and subsequently established as a tertiary institution with the mandate to provide both tertiary and professional education. In 2008, the University received a presidential Charter to operate as a full fledge public University.

The University is committed to providing and promoting higher professional education and training by leveraging a structured mix of scholarship with professionalism in Ghana and beyond. It has a vision of being a world class education provider in both academic and professional discipline that is nationally entrenched, regionally recognized and globally relevant. UPSA as at the year 2014 had a student population of over 11,000.

UPSA like any other organization places value on data and information as such, right from its inception, data, records and information has been managed through various means, with each newer method adopted being a better way of managing records than the former. In order for the University to run smoothly, timely and accurate flow of information is necessary as students, management, accreditation board, potential students, investors and faculty staff need it. At the beginning, UPSA manually managed information by writing records in notebooks labelled appropriately and storing them on shelves and in cabinets. This method was quite manageable because the student population as at then was not up to a 100, but then there was the challenge of data inconsistencies. In 1997, the University started to use computer applications to manage information, where the records were kept in Microsoft Excel files and stored on a computer. Because records were saved on a desktop computer, which was then used by

the IT officer who doubled also as the administrator and without network of any sort among the three computers which were in use then, data and information was passed around using floppy disks which also posed the challenge of inconsistencies in data. Later on in 2008, when the University gained the mandate to operate as a full-fledged public university, its management felt the need to look for a better and safer way to manage data and information of the institution because, by this time, not only had the student population increased tremendously, the employee base, as well as assets had also increased greatly, and using the Microsoft excel workbook to manage the information of all these developments will lead to further challenges.

In response to their need for an optimized way of managing information, management of the University in 2008, went in for a software package named Online Students' Information System (OSIS) which was developed by a company called IT Consortium. OSIS had since been in operation, though with the increasing challenges that users encountered on a daily basis. In 2013, many of the stakeholders, including students, faculty members, management (the vice chancellor, the pro-vice chancellor, the registrar and the finance director), administrative staff, IT staff had grown tired of using the OSIS and their heightened complaints had finally settled on the ears of management also coupled with some other major challenges that could not be resolved out-rightly and so it became necessary for the University to opt for another system which led to bespoke development of the University Information System (USIS).

It is also worthy of note that in latter parts of 2012, a major structural change was effected in the University and one of the notable ones relevant to this research was the transformation of the IT Department to Information Services and Technology Directorate

(ISTD). The department existed to give basic IT support to staff and to ensure that the internet was working at all times, whereas the formation of ISTD led to the establishment of three sub-units; the Enterprise Computing and Information Systems (ECIS), Policy Strategy and Support (PSS) and Infrastructural and Engineering (I&E) Units. The ECIS unit was to ensure that the information needs of every other departments were met. I&E was responsible for developing systems and applications and extending network infrastructure to new buildings on the Campus and PSS, had the responsibility of formulating policies and ensuring these policies were implemented, whilst providing IT support to users. These sub-units were headed by newly employed deputy directors who all reported to the Chief Information and Technology Officer (CITO), the director of ISTD, who was also a new staff. I&E was to be made up of a development team and a networking team, however, there were no staff members to form a development team, as this was a new initiative brought on by the CITO, but there was an existing networking team. Details of how the development team came to be is discussed later on in this chapter because their contribution to this case cannot be ignored, however, for the purpose of this research, the terms ISTD staff and development team would be used interchangeably. Following the description of how the USIS emerged, the rest of this section traces its development.

5.2.2 Decision for USIS

As mentioned in the previous section, OSIS had various challenges with its operations, as a result, a number of stakeholder groups were not satisfied with it, because it caused a strain in relationships among and between members of different stakeholder groups. For instance, the OSIS did not have a module for graduation, therefore a Microsoft Access application built by one of the IT staff was used to process data and in order to determine

which students were eligible to graduate and to generate broadsheets. The Access application was always fed with data from the OSIS database to do its tasks, but because the IT staff did not have access to the OSIS database, common tasks such as generating data for reporting had to be referred to the vendor which usually took a long time, thereby slowing down the entire graduation process. When management of the University queried the administrative staff responsible for the graduation process, they in turn blamed the problems on the incompetence of the ISTD. Administrators expected that requests and complaint made about OSIS to the ISTD should be resolved immediately, however that was not possible as these demands were always referred back to the Vendor of the system. In an interview with one of the ISTD staff, he said that, "it usually felt like we were making too many demands and they begin to threaten us with more charges and bills" and so he felt that taking ownership and management of any information system used within the University will ease up a lot of tension between his unit and the other units and make his work easier. Since the unit can have access to source codes and work with it to resolve problems that would otherwise have taken a long while to be resolved by the vendor, it will ultimately erase the notion that the ISTD lacked competency as it had always been perceived by the other units and departments.

There were times when inaccurate data was used to generate broadsheet with the Access application, as such, some students who should have qualified for graduation were not on the graduands list, and so resort to suing the University in the court of law which created a lot of problems for the University.

There was also the issue of trust with the OSIS as interview with certain members of the organization revealed that although no solid evidence was available to prove their points,

they perceived that the OSIS had become porous, so that student examination grades and other sensitive data were easily manipulated. Thus management felt that changing the system would help restore credence in the business processes of the University which was phasing out as a result of the many compromises that existed in the OSIS in terms of security and policy that led to things been done inappropriately, thus the words of one of the senior managers during interview,

> *"we needed to streamline these things by putting in place appropriate controls and standard operating procedures".*

This also contributed strongly to why a new Information System must be implemented.

One major factor that led to the abandonment of the OSIS was about finances, although not upfront with it, but the rapid nature with which the decision was made and some of the comments that were made by interviewees revealed this fact. For instance, comments like *"we saw the development of USIS as a means to cut down on cost"* coming from one of the senior managers in an interview confirmed this point. UPSA as at 2013 spent a total of more than $160,000 that year on OSIS which was definitely going to increase annually based on the size of the database as per the contractual agreement. Several appeals made by the directors of the ISTD to the management of ITC to review the terms of their contract with regards to OSIS so that the University can renew its contract with them and continue to use their services amounted to nothing. More so because the University was then growing really fast which also indicated that their bill to the OSIS service providers was also going to increase, management of the University decided that they were no longer going to employ the services of ITC, hence the need to urgently change systems.

Now as it became clearly understood and unanimously agreed upon by the RSGs identified for this institution that there was definitely a need for a change of systems, several options were presented and decisions as to whether to purchase an already packaged system or to build one in-house was made. This decision wasn't a straight-forward one as the end could not be predicted from the beginning. Management of the University, prior to the popular demand for a change had already done some assessment on a system named i-Campus, which was developed by a Ghanaian company registered as Telligent (GH) Systems Limited and were satisfied with the performance. However because, the system was built on a VB.NET platform using the ASP.NET, the development team raised an objection to its adoption, despite the willingness of the vendor to sell the package together with its source code, because it was thought to create a major learning curve since the software teams members had their basis in the PHP web programming language, thus will need more time to learn and familiarize themselves with the asp.net language which may take at least a year. The university would still have to go back and forth to the vendor of i-Campus to solve basic problems within the development team's learning period, which also implied that the problems with OSIS would reoccur in addition to the monetary charges that will be incurred in seeking the help of the vendor every time.

UPSA is a public university and is therefore regulated by government laws. The University as at this period was restricted by an embargo the government had laid on all public institutions on the employment of new staff, therefore people with expertise in the area of the technology used for i-Campus development could not be employed. This made the i-Campus system less attractive to the RSGs than it originally was, however this did not deter senior managers from still considering the possibility of acquiring it,

since their intention was to go ahead and employ persons with the necessary expertise so they can be paid through internally generated fund until the embargo is lifted.

Despite all the arguments and concerns raised by the development team, management started the procurement process of the i-Campus, following a series of presentations and demonstrations which was made to the stakeholders for assessment, comments and criticisms. Like other public institutions, UPSA is also bound by the procurement law of Ghana which has laid down rules and regulations as to how items should be procured. The University was also in this case obliged to adhere strictly to this policy which could prolong the process for about a year or more. Management of the ISTD therefore took advantage of the opportunity in the length of time it could take for the procurement process to be completed, and believing that the period could afford them enough time to build a system similar to i-Campus, if not better, brought about the inception of the USIS. Thus the thought of developing USIS was originally the initiative of the development team of ISTD, because they knew that the vendor package that the university was going to purchase was too generic for it, considering how frequently academic structures change in the environment. They also believed that they had the capabilities and skills to do it.

After one year, the university was still on the procurement process to acquire the i-Campus, by which time, the development team had gone far into the development of USIS by themselves. It was at that point, that management of the University realised that they did not have enough time to continue to wait for acquisition of the i-Campus, because at that time it was in late December 2013 and another semester was to begin in January 2014 which is expected to have a new system in place. Management then gave

the USIS project a second thought and decided that it should be thoroughly tested, demonstrated and presented to the stakeholders so that feedback will be gathered and used to better enhance the system. The numerous presentations and demonstrations conducted brought awareness to the USIS such that by then, about 50% of the University staff and stakeholders had at least heard about it and ready to offer any form of assistance, in terms of information gathering and feedback to ensure that the whole project was a success.

5.2.3 Development of USIS

The nature with which management of UPSA arrived at the decision to implement the USIS project, resulted in most of the development issues and decisions to be left to the discretion of the development team. Although the project at its inception was not formally known to the University, it still went through the SDLC, though not in the purest form. The University as at then did not have permanent staff who were developers, they were mainly National service persons and students who were on industrial attachment, therefore, the director of the ISTD decided to put up a development team which consisted of designers, systems analysts, programmers and business analysts. These people were brought together from various places who had passed interviews and numerous tests meted out to them to justify their inclusion in the team. The team was formed solely for the purpose of developing the USIS and subsequently become part of the University through employment. However since the University was still restricted by the embargo and could not employ the team members, funds were generated by the directors of the ISTD to pay them so that they can be motivated to remain with the team until they are finally employed as staff of the university.

The development team evaluated the OSIS and gathered requirements from stakeholders through formal and informal means. Interview sessions were held with various stakeholders to know how the existing system has worked for them and how they would want it to work for them. The team also sought for other IS relevant to higher education institutions that could serve as benchmarks in addition to the i-Campus system. The benchmark system was needed for the team to understand the general business processes underlying a students' IS. An Indian software, known as EBISFRAME ERP was used as a yardstick for the development of the USIS. EBISFRAME ERP was selected because it had all the modules that UPSA needed at that time, such as admissions, course registration, examination and grading, graduation, fleet management, hostel management, procurement system, clinic management and inventory management.

The requirements and feedback gathered by the development team from the interviews with stakeholders and their personal observation of the existing system and benchmark systems were then developed into a new system requirements specification document. Proposals were prepared by the directors of the ISTD which stated the specific problems that the USIS sought to address. The proposal also gave a brief overview of the estimated timelines and budget of the project. A comprehensive project management plan was developed which detailed the project schedule, cost plan, quality management plan, communication plan, risk management plan and change management plan. Procedures for how the project will be executed, monitored, controlled and closed were also clearly defined so that the team could stay focused throughout the project. Moreover, tasks, activities and timelines were assigned to individual team members.

Development of the USIS was done in modules, in order of relevance and urgency so as to adequately address the issue of limited time which was a concern of management of the University. Since all team members had been assigned tasks, that is, a module each or an aspect of a module and deadlines were assigned to each task, the team members worked tirelessly to achieve the goals that had been set before them. In order to ensure monitoring and control of the software, a development committee was set up along the way to regulate the activities of the team.

The development team commenced this project in February 2013 and by March the same year, all necessary documents and requirements needed for the codes of program to be written was made available to every member of the development team who commenced work immediately. Since the system was built in modules, by December 2013, management of the University issued a directive that the USIS should be prepared for use in the following semester, that is, January 2014, by then the system had the major modules built and running. The following section presents a discussion on how USIS was implemented.

5.2.4 Implementation of USIS

The course registration module was the first to be rolled out. This module was heavily dependent on other modules such as the course, department, faculty and setups modules. Several tests were carried out on the course registration module and presentations and demonstrations were done by the development team to the rest of the stakeholders so that issues and concerns regarding the USIS would be adequately addressed and ironed out before the system could be deployed. Also trainings and user manuals were developed for users to help them use USIS effectively. At this stage of implementation, almost all

the stakeholder groups had become fully part of the project, and because they perceived the system to be capable of meeting their needs, they were ready to support the cause of it in whatever way they can to make it successful.

The course registration module was first used by the level 400 weekend school of the University, whose students population formed only about 5% of the entire student population and the system worked perfectly, but for a few hitches which resulted from limited hardware capacity, because the application was piloted on a local server, which was not built for that kind of activity. These hitches were resolved by simply hosting the application on a dedicated server, when the rest of the student population had to use the module for course registration. By the end of the August 2014, the admission module, the examination and grading module completed and had gone through rigorous tests and debugging. These modules were then integrated with the course registration module to form the USIS.

There were however some challenges with rolling out the course registration module of USIS, which was the first of the modules to be deployed. The system had gone through several tests on the local environment which proved it to be successful and ready for implementation, as a result was not piloted before it was implemented live. The USIS application in itself wasn't the problem, however the server on which it was hosted on the live environment couldn't handle a lot of request at a time and therefore kept crashing. This occurrence raised a lot of concerns as some stakeholders expressed fears of sustainability of the system. Some more skeptical RSGs argued that the earlier the system was abandoned, the better it was, so that it wouldn't cause the University greater harm in the future. The heads of the technical team however explained what led to those

occurences to management, who in turn understood the issue and agreed to purchase servers with high specifications for the purpose of hosting USIS.

5.2.5 Consequences of using USIS

USIS implementation was successful and it run fairly well, as a result the development committee who were not members and staff of the ISTD gave it a score of 98%. The score was given based on a number of criteria including its ability to save the university a minimum amount of $160,000 annually.

Having completed and thoroughly tested the relevant and urgent modules that the UPSA needed at that time, the development team was fully confident that the USIS was capable of meeting the requirements of the university better than OSIS and it also had additional functionalities that could make work easier for various types of users. For instance, during the use of OSIS, the VC would always make a request to administration for reports and statistical information on say, applications and admissions for a particular year. The request would then be routed to ISTD, who in turn ran queries directly from the database to retrieve the information and hand it over in a hard copy form before the VC can lay hold on the reports, whereas in USIS, an interface had been provided for the VC to log onto so that he can by himself generate any kind of report he wanted within minutes without having to suffer any delays. The team was very impressed at their ability to develop a system that gained the admiration and support of other stakeholders. Although at the initial stages of the development, their attempts and efforts to make USIS a reality was not esteemed, the CITO of the ISTD, encouraged and motivated the development team to carry on with the development as it would sharpen their programming skills and open doors of opportunities for them in other organizations.

More so, the friendly user interface of the system, the ease of use, consistency and accuracy in data and processed data after migration from the existing system to the new system contributed to the excellent rating of the system. However, as staff of the university began to use the USIS, the extent of its use and its expectations at the initial stages, later on became different especially for the faculty members and the administrative staff. The contrasts in expectations for these social groups was traced back to the different technological frames within which they operated. Although the system had received lots of complementary remarks from users and stakeholders, the system could not be regarded as stabilized. This was due to the dynamic nature of the needs of users and stakeholders in the UPSA environment, and so the system was constantly reviewed by the development committee who took feedback from the RSGs and submitted them to the development team who in turn worked in their capacity to ensure that the concerns were addressed:

> *"One of the immediate benefits that can be appreciated now is that when there's a problem, the solution is at our doorstep, because the development team is here. But also because of that proximity, sometimes, the stakeholders make all kinds of requests at short notice and it gets done for them, which they couldn't have gotten from the third party."*

Developing USIS had both positive and negative consequences on the development team. A positive consequence was that in order to properly maintain and upgrade it, the development team attended several training sessions that was organized to help improve upon their development skills and keep them up to speed with issues relating to the

technology platform adopted for development. These training sessions were fully funded by the University and they were very beneficial to the team.

Furthermore, the development team gained more knowledge and expertise through the development of the USIS, such that they were sought after by other small-to-medium sized enterprises (SMEs) to help develop in-house solutions. Therefore the experience gained from the project afforded the team a certain level of competence, confidence and publicity.

On the other hand, the development team, in order to maintain and upgrade USIS had to sometimes work odd hours just so to ensure that functionalities requested for by stakeholder groups are developed and running smoothly by a stipulated date. For example, the team had to work throughout the night of one of the Christmas holidays to migrate students' examination grades from the existing system to USIS. This activity was met with some challenges because the database structure of the results table of the existing system was very different from the structure of the new one which slowed down the whole process.

The team also had to constantly deal with issues that did not relate directly with the new system. A typical example is when users had to call on the assistance of the development team to resolve trivial issues such as connecting to the internet in order to be able to have access to the USIS. Some of these users even go to the extent of writing to management that the system was not good enough and so should be abandoned for a new one. Management in response to these complaints had issued several performance queries to the members of the development team, which they otherwise would never have received

if not for the development of the system. Members of the development team at some point became unhappy and discouraged by the many query letters they received from management and consequently felt that their efforts in developing an in-house solution for the University was not really prized, considering that they did it voluntarily and for that matter for free.

The development team recognized the capability of USIS to reduce response time, offer a 24/7 support due to the availability and proximity of the developers, save cost, allow ownership of the product, eliminate surprises from third party vendor and also build institutional capabilities, although it had its own consequences. This is evident in the response of the CITO of ISTD when asked about the disadvantage of developing and implementing USIS:

> *"I would have also wished that we did things at a reasonable pace instead of responding to the demands of stakeholders. Because sometimes the demands came at very short notice so it puts the engineering team under stress to come up with a solution at short notices. It had consequences on the systems, because sometimes the systems were not thoroughly tested and it went into production. So, that I would have insisted that we follow the work plan, so that all these short notice request, even though sometimes were critical, if I had my own way, would have being done differently."*

Team members were also sometimes subjected to harsh scrutiny by administrative staff, in the event of any suspected changes of students' academic record. Most of the members of the team were new in the University environment implying that they were not well known, and because of the experience with the existing system which was alleged to have

been manipulated to change students' academic records, stakeholders could not afford to give anyone the benefit of doubt and so handled suspicious issues aggressively, even when it wasn't so.

Lecturers were happy with the system as it allowed them to upload students' examination results from anywhere provided there was internet access, unlike the previous system where results had to be entered only on the University's network, which implied that lecturers must be on campus to be able to upload results. Another benefit of the system was that lecturers could download an Excel file that lists the students who had registered for their respective courses, enter the results onto the excel sheet offline, and afterwards upload the excel file onto the system to record the examination marks and save. The existing system did not have that functionality, as a result lecturers always had to be online to enter results directly onto the system. The problem with the old method was that, if in the course of entering results, the internet connection gets lost, the lecturer would have to stop and continue at a later time when the internet connection was back. Despite the upload functionality on USIS, the lecturers had issues with the fact that it did not have progress bar to indicate the progress of the upload. For Excel files of about 3000+ students, the upload could take as long as 8 minutes or more and lecturers could not tell most of the time whether it has completed or not. Most times, when the upload was taking very long, they complained that the system had frozen, because nothing indicated the upload progress. Also lecturers found it easy to identify, trace and resolve issues relating to Incomplete (IC) Grades of students result on USIS, unlike the existing system where IC grades of students were not easily identified and resolved, as result students' overall performance were affected and in some cases the students were denied the chances to graduate at the appropriate time.

Again the USIS could enable examination results of students to go through a series of checks before finally being approved and available for students to view. Once a lecturer submits a set of examination results, it gets escalated to the Head of Department (HOD) of that course, who either approves or disapproves the results. A set of disapproved results is sent back to the lecturer's dashboard, so that necessary actions could be taken on them, since only a lecturer had the privilege to enter and modify grades. If a lecturer submitted a set of results, he or she could only view them but could not make any modifications to them, a feature that won the heart of the academic staff, as it was not functional in the existing system, thereby causing a lot of inconsistencies in students' results. A set of results approved by the HOD gets escalated to the Dean of Faculty who also either approves or disapproves the results. An approved set of results get escalated to the academic director, who then publishes the results for the students to view. However this feature required that faculty members log on to the system daily and check on their dashboards regularly to know if something new had arrived for their immediate action, especially during examination grades entry period. This was particularly not practicable for the faculty members, as it was very new to them and so therefore caused a delay in the approval process of a set of examination results. An instance of this situation was when a HOD did not know that courses had been escalated to him and therefore did not act on it, it brought the approval process of all those courses to a halt. It was until students' complained that their results had not been released that, the problem was traced to the HOD in question and rectified. The shortcoming here with the system was that, there was no automated way anyone could tell that some courses had been stuck in the hierarchy if there were no complaints from affected students therefore lecturers requested that the system should be provided with functionalities that could serve the purpose of

notifying them of any activity that required their attention and that could tell at which point or stage any of their courses was on the hierarchy and who was to act on it.

Another concern about USIS that faculty members had was the escalation of students examination result to respective HODs, Faculty Deans and finally to the Academic Director who then publishes the result. Faculty members did not have any fore-knowledge about this process, because it was not a practice with the existing system and so they did not understand the logic behind it neither did they see its relevance and so they concluded that the development team have gained too much power as such were changing the ways things worked in the university. Some even thought that the members of the development team were being mischievous. However when the reason and logic was explained, it was found out that there was an error with the logic that was programmed into the system. The ideal thing was for examination grades of a course to be escalated to the HOD of that course and not to the HOD of the students who registered for the course, so this gave rise to confusion on the parts of the HOD and Deans because they saw strange courses (courses which were not offered in their departments) on their dashboard and they did not know what to do with them. A solution was offered during one of the meetings held to resolve this conflicts that the logic should be corrected before its next use and so the development team took on the task and effected the necessary corrections on the system.

Administrators from the beginning simply wanted a means to generate reports at a particular point in time, but as they began to use the system they realized that they needed reports on the history of such activities. For instance, an illustration was given on how report was generated for the Vice Chancellor whenever he requested for it which really

impressed the administrators because it did not require the intervention of ISTD. With time they realized that they needed the statistics of the number of applications received and admissions offered for the previous years, which was not a functionality of the system, however upon request made to the CITO of ISTD, this functionality was incorporated into the system.

5.3 VVU Case Description

This section presents findings from the case of VVU. It begins by delivering a general overview of the University and a brief introduction the existing system used for managing records. The section further presents reasons why the university decided to implement bespoke solutions in-house and describes how the system was developed. The section concludes by recalling the implications of the University's decision to develop a bespoke system for managing information.

5.3.1 Background of VVU

Valley View University (VVU) was established in the year 1979 by the then West African Union Mission of the Seventh-day Adventist. The University was originally named Adventist Missionary College and was situated at Bekwai-Ashanti. In 1983, the college was transferred to Adentan where it operated in rented facilities, until in 1989 it relocated to its permanent site on the Accra-Dodowa Road and then got renamed to Valley View University. The University in 1995 became affiliated to Griggs University in Silver Springs, Maryland, USA and was opportune to offer a four year bachelor's degree in Theology and Religious studies, but in the year 1997 the Ghana National Accreditation Board granted VVU an accreditation to award its own degrees, thus making it the first private institution in Ghana to be granted accreditation. VVU's vision

is to be a leading centre of excellence in Christian education and it sets out to prepare lives for service to God and humanity through academic, spiritual, vocational and technological excellence.

VVU had a manual system which was known as the Green Card system to manage students' records. In order to register for a semester's course, students had to go pick up a green card, and accurately write the course details for which they are registering for after which they had to obtain some signatures before the registration process could be completed. There were six signatures for the green card registration of the Adventist tuition. The green card process required all students to first obtain a financial clearance from the finance office to ensure that they have paid the minimum amount of fees required. The second clearance is obtained from the University's library to show that the student did not owe any book. The third clearance is obtained from the student's department to actually indicate approval for the student to take all the courses for the semester. The fourth clearance obtained had to do with advancement of accommodation, to prove that a student did not misbehave whilst resident in the University's hostel facilities. The fifth clearance was obtained at the cafeteria and the last one was at the registry, where students signed to indicate that they had completed the course registration process. Students had to stay in long queues at each stage of the green card process to register for courses. The courses are registered on a green card, hence the name, 'Green Card System' and these cards are kept at the records office of the University. At the end of every semester, lecturers enter grades of individual students on his/her green card, so the green cards served as result slips for each semester and are then piled up as transcripts for the student. This manual process became increasingly difficult to use as the student population of the University increased. At times it took a student about a week before

he/she could complete a registration process. Because of this manual process, examination grades were printed and pasted on notice boards. Students, who for some reasons, did not check their results before the sheets were taken off the notice boards did not know their examination grades. The implication of this was that students who did not know that they should have re-taken a course to qualify them for graduation and for that matter could not graduate, took legal actions against the university, which ended up in the media and tarnishing the University's image.

Management of the University, therefore decided to implement a technological solution to eliminate these problems and to enhance the speed of operations within the university. That decision ultimately led to the development of i-School.

5.3.2 Decision for i-School

In 2002, the University called on some vendors which they believed could provide technological solutions to information management problems. Out of five (5) vendors who were contacted, only one of them actually came to do a demonstration of a similar system they have already developed, the rest only talked about how they could do it. Management of the University then decided to acquire the system from the vendor who did the demonstration. The university as at that time, had an academic Computer Science Department (CSD) with lecturers who were quite knowledgeable in the field of Information System. The members of the CSD after seeing the demonstration of the system from the vendor, then told the management of the University that they could develop a better system for the University. Management, did not readily accept the offer made by the members of the CSD because of the many instances of the failure stories associated with bespoke information system that they had heard, however the head of the

CSD made a negotiation with management to give them two (2) weeks to develop a prototype of their proposed system and then invite the vendor back so that they could both demonstrate their respective applications to management of the University. Management agreed to this negotiation and in two weeks, members of the CSD were able to develop a prototype of the system which was made up of all the interfaces and the concepts that would make up the entire design to show management that they were capable of designing the system. Demonstrations from both sides were done to management, after which majority of the decision makers, favored the prototype that that was developed by members of the CSD because they perceived it to be able to meet the specific needs of the University and to solve the actual problems that the University was facing. This decision to develop i-School emerged out of the need for technological solutions to better manage the records of the increasing student population of the University.

5.3.3 Development of i-School

The student information system, which was known as i-School, was initiated by members of the Computer Science Department (CSD), also referred to in this section as, the technical team of the Valley View University (VVU). The team proposed the development of the system, presented it to management, and management accepted it. The technical team upon obtaining approval from management to develop i-School commenced work on it by designing interfaces. The team was made up of two lecturers and five students from the CSD, with the proposal stating that the students should be granted scholarship for their period of study and employed in the University when they had completed.

A version 1.1 of i-School was developed using an oracle database as a client server application for only course registration. Course registration at that time, was not online so students had to go to campus to join queues respective to their various departments that were manned by faculty staff from each department who did the course registration for the students. Each department had five manned computers that had being set up to exclusively do course registration for their students. The process required the faculty staff to use a student's ID to pull his/her detail from the system, thus if a student is not eligible for registration, it would be indicated on the system. For instance, if a student had not been cleared by the library as having no issues, the system would flag it so that the student would have to go to the library to resolve the issue, before he/she could come back to register courses. This version allowed for the registration of courses to be done only on the University's premises and it lasted for only one semester, that is, the 2004/2005 academic year, first semester.

In 2004/2005 academic year, second semester, i-School was upgraded to a web based application, however the registration process was no different from the previous version. Students still had to go to the campus and join a queue until it was their turn to register. The difference with this newer version which was referred to as 1.2 was that, lecturers and faculty staff who manned the registration system, did not have to use specific computers, that had the client-server application installed on it, they could access the application from any computer, so long as they were on the University's campus and network. More so lecturers could use the system to enter students' grade. The technologies used for the web application version of i-School was PHP, HTML, MYSQL, APACHE, JAVASCRIPT, and CSS.

The 2005/2006 academic year first semester saw another evolution of i-School that is the version 1.3 that included a biometric registration system for the cafeteria. This new module was piloted that semester, however it failed, because there were incompatibility issues with the biometric device that was acquired for the program.

VVU system was such that students did not necessarily have to pay their fees in full, they could pay the minimum requirement to qualify them to attend lecturers but the system would not permit any student who owed fees to sit for the end of semester examinations. Normally in i-School chits with specific codes were generated for students who did not owe any fees, so that they could be allowed into the examination halls to take the exams. These codes were designed in a way that it was difficult for people who had not been trained to work with it to decode. The 2005/2006 academic year second semester's version of i-School introduced a functionality that printed out a list of all defaulters in terms of financial settlement to the University, two weeks before the examination period so that affected students could clear themselves of all financial obligations.

In 2006/2007 academic year, the development team decided to expand i-School, by this time one of the lecturers who formed part of the development team had gone abroad for further studies and people within the University had started having issues with the lead programmer who was a student by then and so this posed a great challenge to the team. The issues levelled against the lead programmer, who was in his final year at that time, was that he had become arrogant and disrespectful and was alleged to have manipulated and changed exams grade for certain students, so he got suspended. This somehow slowed down the expansion of i-School, because the original plan was to develop the system into becoming a full blown ERP system with components such as the library

system, the cafeteria system, the hostel management system, the fleet management system, however with the absence of two of the team members, the rest of the team concentrated on other peripheral things such as fine tuning the transcript and reports and feeding the system with data.

In August 2007, the second lecturer also went abroad for further studies, by this time the students who had become staff of the University took over i-School' upgrade and maintenance. The success of the first version of i-School had earned the trust of most RSGs within the university and so the development team could afford to conduct system analysis and requirements gathering which translated into another version of the i-School. In September 2008, version two of i-School was rolled out with the following features: Grade module that could enable lecturers enter grades for students on the platform; students were able to log onto their i-School accounts to do course registration by themselves; a module for hostel management; a module for applications and admissions; and a timetable module. The new set of developers felt more comfortable with the ASP.NET programming language than that of PHP, therefore the version two of i-School was developed using the ASP.NET programming language. This version of i-School nearly failed when it was deployed because it had a whole lot of errors such as, students seeing courses they did not register for, showing on their portals. There were several complaints about the system, however the head of the development team as at then, pushed hard for the system not to be abandoned, because people had already started requesting for the restoration of the version one of i-School.

Version three of i-School enhanced the grade module that facilitated easy entry of examination results by lecturers. The version two could only allow lecturers to enter the

grades manually onto the system, that is, they had to deduce examination grades by themselves through the totaling of raw marks before manually feeding them into the system. However version three allowed the upload of raw marks from Excel files and the system did the computations and deduced the grades. The lecturers were quite pleased with this functionality because it took a load of work off them thereby made the whole process faster. A graduation module, where students can apply for graduation was also a functionality of version three. The graduation module worked by processing registered applicants to determine whether or not they qualified for graduation. Another remarkable feature of the version three of i-School was its availability online, such that users of the system did not only have to be on the University's network before having access.

The development of i-School did not follow the conventional SDLC, when asked during an interview whether system requirements were gathered from stakeholders before the prototype was designed, an informant who was part of the development team at the time, responded by saying:

> *"No, the reason being that, one, those of us who were involved were actual users of the existing system, so we knew the problems that we went through, so at that first instance, there was no systems analysis. We understood how basically the processes work, how registration was done, how the semesters were run and the way we wanted our results slip because we'd been doing it."*

Also documentation of the system was not done from the beginning of the system, because the developers claimed, there was no time to do that.

> *"At that time, which was in 2003 we were doing agile (agile development) without knowing that we were doing agile. Our approach was agile and in fact, we didn't know that there was something called agile. Our concept was to develop*

something for our clients to see, so that they can see our seriousness and trust us and so we didn't have that luxury of trust from the University that could afford us at least three months to do necessary documentation of the system, before implementation begins."

Security of the system was not very advanced, because the only forms of securities employed were installation of a proxy server to ensure that unauthorized persons did not get onto the servers and having the antivirus running. Another effective security check functionality of i-School was the replication of the examination grades table. What happened was that there were three different tables that students' grades were written into, if the normal processes were followed, however if somebody should change grades through unauthorized means, the fellow would only be able to change them on just one table which had a suggestive name. Any discrepancies in the grades on any of the table will cause the system to automatically flag out so that necessary actions would be taken.

5.3.4 Implementation of i-School

The discussion on i-School development engulfed its implementation, thus much has been covered in the previous section concerning the implementation of i-School, however it is important to note that i-School was implemented on a modular basis. VVU first rolled out course registration module which was the most needed in the University, after which several other modules were developed and integrated into the system after several tests were carried out on them. User manuals and trainings were also carried out for users of i-School to enable its effective use.

5.3.5 Consequences of using i-School

Using i-School had a lot of consequences on the developers, one of which was working odd hours, and even on holidays. When users experienced trivial issues, they'd always call upon the technical team to resolve them, even when the issues were not software problems and most of the time these issues did not require the attention of the technical team. It was as if the technical team was in charge of everything which even caused some people to feel that the team had become too powerful. This was evident in an interviewee's response:

> *"In fact there was a time the registrar told me that, right now I should come and be the registrar because I'm doing everything."*

According to the technical team, the issue of power play was their biggest problem they ever had, because there were people who felt that they were going to be too powerful, which they did not like and as a result opposed the implementation and usage of i-School.

Members of the CSD had to constantly deal with politics within the University to keep the system running, because anyone could just wake up any day and decide that the system was not good and as a result should be changed.

When asked how they had planned to keep the system running in the foreseeable future, a member of the technical team replied as follows:

> *"We setup a mini laboratory to train programmers so that it could serve as a teaching platform, so we recruited some first year students, including male and female to do all the trivial jobs in development, such as designing interfaces, typing help files etc. The idea was to bring all of them on board to understand the way the system works, so that by the time they got to third year, they could take over the*

maintenance of the system, because we were not expecting that whoever became the lead programmer was going to remain with the university forever. This plan worked because when the lead programmer was away, we had people who could play around the system and do certain things."

5.4 Chapter Summary

This chapter presented how a bespoke higher education management information system was constructed from the perspective of two cases of higher education institutions in Ghana. It described the rationale behind the choice of a bespoke software for each case, the development process of the bespoke software and the consequences of using the software. The next chapter presents a comparative analysis on the findings of the study to identify emergent themes and concepts that will be used for discussion in chapter seven.

CHAPTER SIX

ANALYSIS OF FINDINGS

6.1 Introduction

The previous chapter presented a description of the case in this study. It began by giving the background of the case institutions and then went on to present how the institutions arrived at the decision to develop an IS in house. The previous chapter also described the development process of the bespoke system in each institution and then highlighted the consequences of using the system. This chapter analyzes the findings of the case presented in the previous chapter in light of the SCOT theoretical framework adopted for this research. The cases presented in this study are analyzed comparatively by first identifying and describing their respective RSGs and the different interpretive flexibilities held by them. The chapter then goes on to compare the technological frames of reference held by the RSGs within each case institutions and how it influenced the bespoke development. The chapter then comparatively analyzed the stabilization and closure process of both cases with regard to the bespoke information systems, thereafter the chapter concluded with a summary of all that had been discussed and an introduction into the next chapter.

6.2 Comparative Analysis of RSGs and their Interpretive Flexibilities

Four groups that influenced the implementation and use of USIS were identified in the case of UPSA and they included management, administrative staff, faculty members and the development team, whereas in the VVU case five RSGs were identified to have influenced the development of i-School. They were management, registry staff, faculty

staff, the technical team and students. Each of the groups identified in each case had its own subjective meanings that it ascribed to IS in question.

6.2.1 Management

The opinions of management in the bespoke system development for both case institutions were pivotal as this group are highly placed in organizational structure. The management of UPSA comprised of the Vice Chancellor, Pro-Vice Chancellor, Registrar, Finance director and the University Council members. Management of UPSA was aware of the challenges that the OSIS posed to administrative staff and faculty members, however did little to intervene in their plight, until the sky-rocketing cost of continual use of it was realized. The circumstances surrounding the time at which management of UPSA decided to switch systems was a clear indication of how much they wanted to control the cost of managing students' records in the University, which is also evident in the response of an interview respondent:

> *"For me, from the point of use of a user, I felt there was a need for change, my problem with the way the change was done was that, I felt we should have run the systems (USIS and OSIS) alongside at least for one semester, but the ideal should have been for one year where we would have run the two systems concurrently, compared results and in the process, use the timeframe to fine tune the new system, before we finally jump onto it. But for me, the changeover wasn't the best and I'm sure it even presented problems for developers. I'll say that the major consideration on the part of management was cost and so it gave a certain drive to the need for change, but for us as end users, yes we knew there was the*

need for change, but we felt the change wasn't handled in the best way due to the overriding consideration of cost on the part of management."

Management therefore interpreted the development and implementation of USIS as a means to cut down on the cost of managing information, while benefitting from the tailored and enhanced solutions it had to offer the University. Management of the VVU, which was made up of the President, Vice President, Registrar and the Finance director, had however not had any experience with the implementation of a technological solution to manage records, but were eager to adopt one that would be a successor to the existing manual 'green card' system. What management of VVU expected from i-School was a technological solution that would eliminate common problems and mistakes encountered in the green card system, enhance and speed up their processes and accommodate the growing student population of the University. Snippet from interview comments support this fact:

"We realized that the 'Green card' system wasn't helping because it took a long time for student to register courses and at some points in time, people get frustrated because they are not able to register. Sometimes the cards may get missing…and we just decided to come up with an electronic system that will make registration easier"

For both cases, although the development and implementation of these systems looked promising to deliver expected results, management were still sceptical about it and therefore resolved from the initial stages of the development were not worth investing financially into. This resolution was due to the frames of reference which they had and it will be discussed later on in this chapter. It is also worthy of note from both cases that,

managers by virtue of their position in their respective institutions took decisions that led to the birth of the bespoke systems.

6.2.2 Administrative Staff

The administrative staffs of UPSA are similar in constituents to the Registry staff of the VVU. In UPSA, the administrative staff comprised of the Director of Academic affairs, all registry staff, the Examination officer and all staff of the examination unit and staff at the student services Unit, whereas in the case of VVU, the registry comprise all staff at the examination unit, hostel manager, head of cafeteria, head of transport, staff in charge of students' registration and records and the chaplaincy. These social groups are also identified as relevant to the construction of bespoke software for higher education management in their respective institutions. Administrative staff of UPSA interpreted USIS as a system that would make it easier to generate all kinds of reports without having to move data around from one legacy system to the other. USIS offered a platform that would make their work smooth and efficient without having to wait for vendors to solve basic technical issues for them. The case was no different for VVU as registry staff were thrilled at having a system that would make managing student records easier by means of storing, retrieving and modifying, however since the development of i-School was done in-house they were unconvinced about the system being able to effectively generate complex reports. UPSA staff, having already used a packaged system and therefore knew the challenges involved with generating certain kinds of report and so they believed that having a system developed in-house will make it easier for reports to be generated the way they wanted it. The administrative staff of UPSA also perceived USIS as a system that would enforce policy and restore credence to the business procedures of the University especially regarding handling of students' examination results. Also in VVU

the i-School was expected to automate the process of entering and managing examination results for students as the exiting manual process was becoming increasingly tedious and prone to mistakes, which sometimes go unnoticed. Again the administrative staff of UPSA also saw USIS as a means to control the activities of lecturers especially regarding the processing of students' examination results and forwarding to the academic affairs directorate for publishing. The existing system did not have functionalities that allowed activities to be done within a specific time and a typical example was with the entering and approval process of students' examination results. Although notices and memos are given to faculty members to complete this process by a specified date, there happened to be defaulters of this notice all the time, as a result it took a very long time for some examination results to be released with students blaming the administrative staff for it. Administrative staff thought that the strict adherence of USIS to time was necessary for things to be done appropriately in the University. The case of VVU, however had no expectation of this functionality in i-School because the University was going to use technological means to manage students' academic records for the first time and that alone was enough to eliminate the delays they encountered in the process using manual systems. i-School was not seen as a means of control but rather a means to enhance solely VVU's business procedures.

6.2.3 Faculty Staff

The faculty staff of both Universities is comprised of lecturers, HODs, Faculty Deans and faculty staff and their role is mainly to teach students, give out assignments and administer examination, assess the academic performance of students and award grades based on their performance. Faculty members of UPSA were appreciative of the fact that USIS made the entering of examination results much easier as compared to OSIS. USIS

offered the lecturers the platform to enter results from anywhere in the world provided there was internet access and made generating reports on a set of result effortless. In the same vein, the faculty staff of VVU also viewed i-School as and easier less tiresome way to manage students' academic records. Lecturers were intrigued with the fact that manual entry of examination grades on green cards for students was abolished. It became easy for faculty staff to handle re-sit situations for students as sometimes became difficult due to the fact that the cards that bore record of students' previous academic records got missing and thereby eliminating the proof that a student has to resit an exam. Therefore the implementation of i-School for faculty staff of VVU meant that the processing of examination results and the management of the academic records of students had gotten easier. Despite the benefits of USIS to faculty members of UPSA, they were not pleased with its rigidity with respect to time and as a result interpreted USIS as managements' way of putting control measures in their activities. Using USIS to process students' examination results implied that faculty staff had to complete the entire process at a specified date, after which the system cannot be easily manipulated to compromise its stance and since this feature was not implemented in the existing system, the faculty staff had some difficulty in accepting it and therefore fought for it to be discarded, however management stood their ground and insisted that it should be implemented. The faculty members, especially the teaching staff also saw USIS as system that lowered their status in the organizational structure of the University because they would have to dance to the tunes of the administrative staff, since the administrative staff were in charge of managing the system, performing activities such as, creating user accounts, assigning roles and privileges and setting up critical dates and periods such as application periods, course registration period and period for processing examination results. This

interpretation of USIS was evident in a lecturer's angry comment made on one of his visits to the ISTD to have an issue resolved:

> *"You IT people have given the administration too much power that they can now dictate to us when we should finish marking our scripts and entering examination results. It seems they've forgotten that some of us teach four (4) courses and each course with a minimum of 1000 students. You people have to either take that thing off or revise it, 'because it won't work here!"*

Again Faculty staff of the VVU did not have any need to interpret i-School as a system that would exert control on their activities because it was introduced to be a successor to the existing manual system.

6.2.4 Technical Team

The development team of UPSA included a CITO, the deputy directors of ISTD, three programmers, a system analyst, and a graphic designer. The members of this team had all been involved in the development activities such as selecting benchmarking systems, gathering requirement and formulating implementation plan. In order to appropriately and effectively manipulate and use a technology, it is essential that enough is known about it. The development team organized training and demonstration sessions for the different user groups at different times. These sessions provided education and support on just what the different user groups needed to know and work with. For example the administrative staffs were trained only on the modules that they work with on a daily basis and the lecturers were also trained on how to use the examination module to enter and process student grades. This was done so that the users can realize how very easy it is use the system and to ensure the security and confidentiality of the system.

VVU's technical team included two lecturers from and three of the best students from the computer science departments who initiated the concept of i-School and worked hard at bringing it into reality.

VVU technical team's perception of i-School was a system that could meet the specific needs of the University. The team saw that none of the systems that were demonstrated by vendors to management of the University was tailored to suit their specific needs and therefore foresaw the possibilities of huge amount of money going into the customizations of those systems to suit the needs of the Institution. Similarly the development team of UPSA also saw the USIS as a technological solution that would acknowledge the idiosyncrasies of the University. Moreover the system was seen a means to eliminate unnecessary delays in the business processes and to ultimately clear off the perception that the ISTD staff were not competent.

Again USIS was seen as a platform that could save the University a substantial amount of money yearly which could be used to fund other developmental projects. They also saw business opportunities emanating from the implementation of USIS, in the sense that income could be generated if the system is repackaged and sold to other educational institutions. VVU's technical team, unlike the development team of UPSA did not view i-School from a financial perspective, not that they did not regard it as cost-beneficial to the University, but their expectations and for that matter perception off it was mainly about its performance, that is the ability to successfully replace the existing manual system, enhance work processes and meet specific needs.

Furthermore, the UPSA development team saw the development of USIS as a stage to build upon capabilities and development skills and to leave a legacy for the University as something that can be owned and boasted of. This perception was captured in a remark made by the CITO of development team:

> *"Nothing good comes easy, but it's pleasing to see that these are ideas we've sowed and they've come to reality. It's pleasing to see that it helped young engineers build their competencies, experiences and exposure under your management."*

The same can be said of the technical team from VVU, apart from providing technological solutions that are tailored to meet the needs of the University, they saw the ability of producing skilled developers who could be used by the institution through the development and implementation of i-School. The team also saw the development and implementation of the system as an opportunity to prove that they were capable of successfully developing what others thought they would fail at. This was evident in the response of one of the students who formed part of the development team, to an interview question:

> *"…our motivation was intrinsic, to test the knowledge that we had gained…and for us as students, it was also an opportunity for us to showcase what we could do and what we have learnt over the years and to prove them wrong that basically there are some things that we could do right…and when we thought about how we built our own system from scratch, it tells us that we are being taught well".*

A respondent also supported this point by a comment he made:

> *"…It's also a sense of pride that you know that the software that you are using is developed by people you have trained, so you know that the graduates coming out are market ready…"*

It is evident from the two cases that the technical team are more interested in how efficiently and effectively the information systems can meet the specific needs of the universities and the fact that the working with and on the system was a good way to improve upon their technological skills.

6.2.5 Students

VVU at the initial stages of their development identified students as relevant contributors to the development of i-School. Students had to do course registration via the i-School, although from the onset, the system didn't offer the convenience of registering anywhere in the world by themselves, they were satisfied that the days of standing in several long queues for hours on different days just to complete course registration was over. To the students, i-School had made course registration much easier and faster, because it eliminated the long queues and delays they endured during registration with the manual system. The stress of having to wander from one office to the other to obtain all the six kinds of clearances before course registration process was completed had been greatly reduced. It took a student at least two days to complete course registration on the manual system and sometimes there were mistakes with the process, but with i-School, course registration was completed within hours and was accurate. Although students still had to join a queue to register, this time the queue was shorter and was only once as there was no need to join other queues because the many clearances to be obtained had already being taken care of by the system. Students were thrilled at this new system because it sped up their registration process. In the case of UPSA, students were not identified as part of the RSGs, even though the students had to use the USIS to perform online registration by themselves. Just like the case of VVU, UPSA students had always join long queues for hours in order to be able to do register courses, only that students did not

have to obtain numerous clearance, all that was needed was a financial clearance in order to register. It was evident that because developers had a fair idea of what course registration on an online information system entails from the knowledge gathered from the existing system and "best practices" from benchmarking systems, the online course registration feature was incorporated into the design of USIS without the considering the views and inputs from student. Later on in the use of the USIS, it became apparent that students also influenced the construction and shaping of USIS. The case of UPSA goes to buttress the fact that all the RSGs may not be easily identified from the beginning of a software development project, however, newer RSGs emerge as the project evolves.

6.3 Technological Frames

This section explains why the RSGs identified in each case interpreted the bespoke systems the way they do and how they relate to it.

6.3.1 Management

Management of UPSA expected USIS to eliminate the rising cost associated with information management in the institution. Snippets from the comments made by interviewees point to the fact that the main motivation for USIS was for financial reasons. Example of such comments include:

> *"USIS was developed out of need for financial reasons and was expected to save the University of a Minimum $160,000 annually."*
>
> *"The ISTD, because it's a directorate that advices management on what to do when it comes to technology, recommended that the system should be built in-house and hence the management of the university supported. The other motivation was also to reduce cost, increase productivity and efficiency."*

Since the need to change systems was of high necessity and a matter of urgency, management originally thought that there wasn't enough time for a new system to be developed and thoroughly tested. Although management knew that developing the application in-house was more cost effective, they didn't believe that there actually was a development team that could develop USIS. As at the time the development of USIS was initiated, the University did not have any staff on the development team. There were a couple of staff members who formed the then ITD which later became ISTD. These members were mainly network, hardware and support staff. The formation of ISTD largely contributed to the decision to develop USIS, because the directors' of ISTD, who were recruited to manage the newly formed ISTD, argued that the directorate should contribute more to the University, than when it was only a department. As a result, a development team was put together with each member of team qualifying to be one after passing a series of interviews and successfully executing assigned projects. Members of the team were by then not employed staff of the University and the University was also not ready to employ anybody because of the government embargo laid on employment, but the team members were willing to take up the challenge of developing the application.

Unlike, management of UPSA whose major expectations from USIS was it cost benefits, VVU's management were on the lookout for a system with good performance and so were ready to settle on a vendor package that had been previously assessed and approved. VVU were not faced with the challenge of cost and so would have acquired a technological solution at any cost so long as it was going to meet specific needs. Findings from their case also did not reveal that the University was pressed for time to switch systems. However when lecturers from the CSD offered to develop an information system in-house for the University, the proposal was not discarded immediately, unlike

the case of UPSA where the only reason for which in-house bespoke development was reconsidered was due to cost considerations. A major factor that enabled i-School to be a success and established at VVU, was the fact management made a decision to support the course. During the time that management was seeking to award a contract for the development of a new system to a third party company, the president of the University paid a visit to Soft Tribe, a software development company in Ghana. The CEO of the company then decided to introduce him to the programming team on his staff. The president was surprised, to know that about half of the programming team were students from his University. Interestingly, he found out that the students he met at Soft Tribe would not have qualified to join the technical team back on campus by the standard outlined by the leaders of the technical team (the two lecturers on the team). With that knowledge the level he had for the technical team to develop a technological solution in-house rose, which led to his unflinching support of the cause.

VVU also did not have staffs who were software developers, but management of the University took the risk to use very good students under the guidance of two lecturers from the CSD, who formed the technical team to carry out the development. It must also be noted that other RSGs of VVU were initially vehemently against in-house bespoke development because of the numerous failure stories of bespoke information systems they had heard about, but management eventually decided to give the technical team a chance, since it was going to cost the institution nothing except for the cost of acquiring a licensed ORACLE DBMS. The case of UPSA revealed that the cost of employing developers for the purpose of developing USIS contributed to its resistance, until a reasonable negotiation was made between heads of the development team and management. This implied that the development of USIS would not have met with much

resistance, had there been readily available human resources, because then management did not have to incur any cost on a system whose end could not be predicted.

Management also argued that the development team were not up to the task of developing such a system and therefore were sceptical about it. The major reason that management doubted the ability of the team to develop USIS was because, both the development team, and even the Directors of ISTD were new in the UPSA environment and had not gain wide recognition in the University, therefore it was only natural for management to hurriedly discard the idea of a whole University management system to be developed by a couple of guys who have just been put together. Similarly, management of VVU did not have confidence in the technical team to be able to develop a usable system, however because those who form part of the team were well known for their academic and professional successes, management decided to offer them the chance to prove themselves, then again, the University had nothing to lose since the experiment did not require huge financial investments nor was it pressed for time to switch systems. Bespoke development of a system implies that the system is tailored to suit the specific needs of the University, it is evident from both cases that how well the developers know the way things are done within an organizations goes a long way to determine the success of an information system implementation. Management of VVU was less sceptical about the competence of their technical team, as compared to the management of UPSA about the competence of their development team because of two reasons: i) their knowledge of the success stories of the technical team ii) the fact that they have been with the University for quite some time and they know the manner of operations, as a result can embed their knowledge and experience into the design of i-School. For the same reasons management of UPSA were more sceptical, in the sense that because of the duration of time that the

developers had spent in the University, they were limited in knowledge about the persisting challenges in the existing systems and the manner in which operations are carried out and even as they were also hoping to beat out time by developing USIS before i-Campus was purchased they resorted to the use of 'best-practices' functionalities adapted from benchmarking systems.

Management was also concerned that USIS could afford the ISTD staff, especially members of the development team, too much power and privilege. They feared that since the application was developed in-house, it was going to be easier for grades of students to be manipulated, because members of the development team were now closer to the students and could be acquainted to some of them, thereby leading to the porosity of the system. This was also the case for the management of VVU, therefore they advocated for the actual management of the i-School to be done by the administrative staff who would work with real records whereas the development team would only work with dummy data in the development environment. Management also laid down a policy that ensured that student developers were not allowed to have access to the system in the live environments, modifications and updates to the live system were to be done by developers who were staff. VVU management also made certain that adequate security and checks were placed in the system to flag out any such instance of students' record change.

6.3.2 Faculty Staff

At UPSA, there was an initial apprehension on the part of Faculty members towards the USIS because they felt they were being forced into the unknown and left with no options at all. One lecturer expressed his dissatisfaction as follows:

> *"In this environment, people are compelled by management directly and indirectly, subtly and cohesion. The cohesion wasn't so visible, but it was more subtle, reason was this, we had to work on results, the data that we can use to work on the result is on the USIS platform. If it were that the OSIS was available, we would have used it instead. But course registration was done on USIS, and so if you don't want to use the USIS, what can you do?"*

The lack of adequate formal communication about USIS resulted in faculty members having the perception that USIS was introduced mainly to cut down on cost, which is evident in the statements of a faculty member:

> *"Management had a challenge with the cost of keeping the old platform running and they were the ones calling the tunes, we just had to go along. As to when exactly the decision was made to build in-house, I'm not privy to that."*

Faculty members had limited knowledge about the USIS prior to its demonstration to them. The first time information about the USIS was communicated to them officially was when they received memos to make themselves available for training. Although during the requirement gathering process, the development team members sought their opinions on what they expect from USIS, the faculty members never anticipated the implementation of the new system to be so soon because they had not receive any official communication to prepare them for a change of systems:

> *"We did not know about the system. We only got to know when we were called to listen that 'something has been brought, this is how we intend to use it'. There was that detachment, because if it had been a university-wide thing and information had gone round, it would have allowed for that effective building."*

In contrast to OSIS, which was too liberal in terms of timelines and deadlines, USIS was designed to have a strict time-bound functionality that did not favour mostly the faculty members, hence making the USIS gain further condemnation. This feature was such that major periodic activities were set up in the system to be completed within a period specified by the administrative staff. The feature worked in a way that once the set up period had elapse, the system would automatically deny access to the affected modules so that further work cannot be done on it. For example, period for course registration is setup in the USIS by the administrative staff responsible, so that students can only do course registration within the setup time. This principle is also applied in application for admission by prospective students as well as entering of examination results at the end of every semester or trimester. Lecturers were then forced to complete everything relating to examination results within the specified time, so that results can be made available to students on time, unlike in OSIS, where student could only see the examination results of the previous semester almost the end of the current semester. Although faculty members saw the USIS as a tool to enhance and make easier their operations, they were however not in agreement with the administrative staffs' expectations of USIS to control the entire examination grading process, which resulted in conflicts and differences. This can be deduced from a comment made from one senior lecturer:

> *"You IT people have given the administration too much power that they can now dictate to us when we should finish marking our scripts and entering examination results. It seems they've forgotten that some of us take four (4) courses and each course with a minimum of 1000 students. You people have to either take that thing off or revise it, 'because it won't work here!"*

The faculty members saw that time-bound functionality of USIS as a platform for administration to control them and their activities. Other faculty members also expressed

concern over why the USIS was not in the custody of the ISTD, but rather with the administration, so that one faculty's dean voiced out his strong displeasure during his visit to the ISTD to have an issue resolved:

> *"Why should I go to [mentioned name of the administrative staff in charge of creating accounts and assigning privileges] to have my account details. Why can't I do it here (the ISTD)? Is that person better than me? This task should be the responsibility of the ISTD and not just anyone. I'm going to make this known to management!"*

The case of the VVU didn't reveal major opposition from the faculty members, except for the fact that the faculty members were somewhat sceptical about the ability of the technical team to provide a lasting technological solutions to the challenges they faced. On the other hand, their familiarity with the lecturers on the technical team gave them some level of assurance that the team would be able to deliver as expected and more also that, the president of the University had urged the team on to develop i-School was a booster of the faculty members' conviction that their expectations would be met through in-house technological solutions.

The appreciation that the faculty staff of VVU felt towards the provision of a technological solution to their plight cannot be ignored as the i-School was to make processing of students results less stressful. Lecturers had to manually enter grades onto a green card for each student, which is created at the beginning of each semester when a student registers. The challenge with this was that it always yielded mistakes, as lecturers had to compute grades from raw marks for each student before entering it on to the green card. Because of increasing size of the student population, lecturers also sometimes made

the mistake of entering one student's grade unto another student's green card. There were also instances where a student's green cards for a particular went missing which caused a lot of strains on the relationship between students and the management of the school such that affected student take legal actions against the University which eventually boils down to the lecturers or the administrative staff. Another challenge that was faced by lecturers was that, the process of entering examination results on the green cards encountered a lot of delays, thereby slowing down the whole process. This was because the different courses registered for by a student are taught by different lecturers, and all this courses are recorded on a single green card for one student per semester, so a lecturer had to wait for the green cards of a set of students, in order to enter examination grades, if they are in use by another lecturer. So having i-School implemented in VVU gave the faculty members a huge relief and caused them to really cherish it.

Back on the UPSA case, the anger and the feeling of disregard that faculty members had with the introduction of USIS was however pacified after their first use of it to enter and process students' examination results and grades. With the existing system, lecturers had to enter raw marks into the system and manually compute and determine grades for those marks before entering it into the system, which was always a tedious thing to do as it slows down their work and easily introduced mistakes which took extra efforts and time to correct. And for those who were not conversant with the use of Excel application formulas, it was hell for them, but USIS relieved them of all these stress, in that all that needed to be done by lecturers was to feed the system with the raw marks and the it would automatically compute and determine grades based on a selected grading scheme.

Moreover USIS offered a comprehensive 'at a glance' graphical report that provided information on the number of students that scored a particular grade. This feature made

it easy for lecturers and faculty staff to trace and resolved Incomplete (IC) results issues before examination results were escalated to the various hierarchy for approval and publishing.

6.3.3 Administrative Staff

To the administrative staff of UPSA, USIS was expected to put in place appropriate controls and standard operating procedures that would streamline the many compromises in terms of policy that plagued the existing system. This was mainly due to the fact that they had different opinions about the way and manner certain operations should be carried out.

The administrative staff of UPSA saw the potential of the system to drastically deal with the issue of system porosity and manipulation of grades which was rampant in the existing system if the controls were strictly adhered to. Administrative staff had often received hints of grade changes done illegally by some unidentified persons who somehow hacked into the existing system to modify students' grades. This was usually brought to the notice of the administrative staff by students who were aware of the acts and were concerned about the integrity of the whole system. This situation was heightened as some public media started probing into it and interviewing students at random about its verity. Even if that was true and it was actually happening, OSIS had no way of flagging out such cases whenever it occurred and asking for that functionality to be included by the vendor of OSIS was not only going to take forever, but also have some cost implications on the University. The birth of USIS gave the administrative staff an assurance that this problem would be taken care and credence would be restored to the entire University's system of operation. Although there were some concerns about

the possibility of the same situation repeating itself again especially since the developers can easily do that without leaving any trace, the directors of ISTD gave the rest of the RSGs the assurance that it wasn't going to be repeated because appropriate measure would be put in place to ensure the security of the system, both from outsiders and people within. More also the stakeholders agreed to set up an audit committee that would randomly audit the system to ensure that there was no foul play. VVU's case did not report any case of porosity in the green card system, although it was very possible to have occurred, but just as the administrative staffs of UPSA were concerned about the system being manipulated from within, the registry staffs of VVU also had similar concerns and especially as the programmers were also students. But they took comfort in the fact that the students were not going to have access to live data and were not going to have anything to do with the live version of the i-School. All modifications and updates, when completed locally by the students, would be pushed up to the live environments by the lecturers (Staff) who were on the technical team.

The administrative staff of UPSA also appreciated the ability of USIS to speed up their work, especially in relation to publishing of results, generating list of qualified applicants into the university and students who are eligible to graduate. The administrative staff had often been queried by management about the delayed publication of students' examination results, which usually results in misunderstandings and verbal confrontations because the administrative staffs apportioned the blame of delayed publication to faculty members. Whilst faculty members did not appreciate the time-bound controls imposed on them through USIS, the administrative staffs were glad that it existed and were always grateful to the technical team for designing the functionality into the system. The registry staffs of VVU were just happy with the fact that having a

technological solution will lessen the amount of law suits the University faced as a result of disgruntled students whose green cards oddly went missing. To the registry staff also, managing students record got easier as the manually system made it extremely difficult to retrieve students' records. Even though there was a chronological way of retrieving records, the registry staff still put in more efforts to manually flip through the hard copy files until they find the record they are looking. Having i-School made it possible to retrieve records at a button click within a split of a second, which raced up their work processes and made it more efficient.

Like the faculty members, UPSA's administrative staffs were as well not officially informed about USIS at its inception, even though requirements were sought from them to develop the application, however they embraced it when a demonstration was made to them and they realized that it had more to offer than the existing system. Prior to the decision to develop USIS and even way before management finally decided that they were not going to make any more financial commitment to the existing system, the administrative staff had always being bugging management for a change of system, and so they readily accepted USIS as solution to the problems they've being encountering with the existing system. They were even happier to know that, anytime there was a problem, the solution was at their doorstep, so they do not have to wait a long while to have their problems resolved by a third party. The registry staff of VVU, were not left out of the development of i-School and so they were well informed of the status of the project at every stage. This group did not have any cause to throw away the technological solutions that was offered them, as a matter of fact they needed the i-School, even more than, the technical team had the desire to design and implement it, although they

criticized some of the functionalities which they believed would not work well for them and advocated for changes and slight modifications.

6.3.4 Technical Team

For both institutions developing and implementing an in-house bespoke application for managing information in their respective institutions was an initiative taken by the development team or technical team, in the case of VVU.

At UPSA, USIS was expected to transform the way students' records were managed, eliminate the problems that the existing system posed and meet the specific needs of the University. The existing system was perceived to be too generic, hence could not acknowledge the particular and changing needs of the university. The CITO of the ISTD saw USIS not only as a tool to be designed, developed and deployed to transform information management at UPSA, but also as a platform to develop technological capabilities within the University. He explained it this way:

> *"Coming from an IT background and also coming from the school of thought that thinks that technology should be radiating from the universities to the industries, I thought that we should build some kind of institutional capabilities, technological capabilities as a university."*

Similar to the UPSA case, the technical team of VVU, was also largely motivated to take on the i-School project due to the mere fact that it was within their reach and to disprove the concept that things coming from within are not the best. In response to a question of what drove the team to offer to develop a system, the lecturer who was involved from the beginning said:

> *"You are a student, you are going through training, you are trying to understand systems before you go into developing systems. Now the guy outside is a practitioner, all he knows is to write codes but he doesn't try to understand systems. They pretend as if they try to understand systems, but they don't, and if they would do anything, they based it on the research that we performed in the institutions. So if we have the people in the institutions, why do we go outside? Those outside, we trained them!"*

USIS was expected to provide financial returns to the ISTD and UPSA as a whole, thus in the words of the CITO of ISTD:

> *"...then also I needed cash to put in place needed infrastructure and the existing system for me I thought was a drain on the purse of the directorate, for that matter the university, I considered it a capital plight because what we were paying for was not far outweighed the benefits we were getting.... USIS has saved the University of a Minimum Annual Amount of $150,000."*

The case of VVU however didn't explicitly report on any financial expectations from i-School, however the technical team knew that developing and implementing it placed the University at an advantage in terms of cost. This analysis was drawn from a response of a member of the development team when asked whether their decision to implement i-School was met with opposition:

> *"...those from the finance department were just against it...even though they were going to buy the other software for I think about $70,000 or something. They were just against ours that was going to be virtually free..."*

The two lecturers who formed part of the technical team at VVU were privileged to have sat on the committee that reviewed several vendor packages that the University had

wanted to acquire. Seeing the functionalities and the performances of the systems, the lecturers decided that the various software demonstrated would not be able to meet the peculiar needs of the University without having to go through major customizations which would require more financial commitments, which was how the concept of i-School came about. The lecturers had been working the institutions for a very long time and they knew the problems they faced and also knew that going for a generic application would only do more harm than good to the university, hence the formation of the technical team to develop a tailored technological solution for VVU. Having had a rough experience with the implementation of USIS, the UPSA development team thought that they have had enough from vendor packages. The implementation of OSIS, presented the ISTD as a bunch of incompetent staff who do not know why they existed. The team endured a lot of pressures from all angles and departments of the University. Whilst the administrative staff were on their neck to produce some reports, the faculty staff were also on their case about why some students examination grades have changed, and to make matters worse, management also kept issuing them with stern query letters, this made the team to strongly advocate for an in-house systems solution that could be effectively and efficiently managed by them.

6.3.5 Students

VVU students also had a cause to be grateful for the design and implementation of i-School. The students had endured a lot of stress as a results of standing in at least six different queues sometimes for two weeks just so to register the courses they had to take within a semester. Their plight was not as intense as that of the students at UPSA, as the UPSA students only needed one financial clearance, hence a maximum of two different queues to stand in, in order to register courses. In both institutions, the students were

appreciative of the fact the technological solutions offered them by their respective technical teams alleviated and sped up their course registration process.

In the UPSA case, students were not considered as part of the RSGs at the design and implementation stage of USIS, because the development team had made certain assumptions of how the system would be used and interacted with by them. But as they began to use the system, students had their own expectations of the system which was slightly different from what the development team had in mind for them. In order to use USIS, students had to log on with their index number as username and a specific format of their date of birth as password, and that was the provision made by the system, with time the students requested for the ability to be able to change their password to something more discrete. This request made by the student body qualified them as RSGs, because their expectations of USIS influenced the modification of USIS in the sense that providing the functionality that would enable password change for students, also required that another functionality that would handle password reset is added to the system. In order to deter the abuse of the password reset functionality by unauthorized students, this functionality was made available to only support staff who were responsible for resetting students' password where necessary.

Table 5 - Summary of UPSA RSG composition and their expectations from USSIS

RSGs	Composition	Expectations
Management	The Vice Chancellor, the registrar, the finance director and assistant registrars	To deploy a student information system that would cut down on cost.
Development Team	The, CITO, deputy directors of the ISTD and the developers	To deploy a system that would not require the intervention of a vendor company in order to perform any modification and enhancements.
Faculty Staff	Faculty deans, HODs, lecturers	A system that would make the processing of students' examination results easy.
Administrative staff	Director of academic affairs and all assistants, student services officers, examination officers and all assistants	A system that would make it easier to generate all kinds of reports without having to move data around from one legacy system to the other

Table 6 - Summary of VVU RSG composition and their expectations from i-School

RSGs	Composition	Expectations
Management	The President, the vice president, and the registrar	An integrated system to transform the existing manual solutions to a technological one and eliminate problems associated with it.
Technical Team	Members of the Information Technology Services (ITS)	A system that would meet the specific needs of the university at minimal cost
Faculty Staff	Faculty deans, HODs, lecturers	A system to process and collate students' examination results.
Administrative staff	Registry staff, chaplaincy	A system that would make it easier to generate all kinds of reports without having to move data around from one legacy system to the other
Students	All students of the University	A system that would ease and speed up the process of course registration.

6.4 Stabilization

Stabilization in USIS was not fully attained, because of the different meaning and expectations attributed to it by the different social groups. Although the collective efforts of the RSGs and their decision to own, contribute and communicate their expectations was aimed at the stabilization of USIS, it was however not attainable because, as the system remain in use, certain problems and opportunities are realized by users that usually calls for its modification. More also because higher education institutions are dynamic institutions with UPSA not an exception, the business processes constantly change, thereby making the information needs of the relevant social group to also change causing their expectations of USIS and consequently the meaning ascribe to it be changed as well. A typical example is the case where management of the University decided that final year students should be able to take re-sit courses exams between the period of June and August each year to give them a chance to be a part of graduation in September. The implication for the development team was that a functionality to take care of this request had to be embedded into the design of USIS. Prior to the design and implementation of USIS, there was no such provision for final year students, even in the existing system and so this functionality was not part of the original design of USIS, but as users and other RSGs thought that this idea could be made possible by the development team, they decided to embark upon it leaving the development team with no negotiation option. Another example is with the decision made by management to conduct January admissions of Diploma students. Application for admissions for all applicants was usually opened from January to May of each year, after which admissions are done between the periods of June to August so that admitted students can report to school in September. Also Graduation for final year students is conducted in September each year, which means that diploma students had to wait till January the following year before they

can apply for Degree programs and be enrolled in September. This had been the norm throughout the previous years and in the use of the existing system, however management decided to open applications for diploma students for the period just after their graduation in September till December, so admission and enrollment can be done in January the following year. This arrangement also implied that, the January in-takes had to remain in remain in school during the summer holidays to take their second semester course, so that they could catch up with the colleagues for the following academic year, since reporting to school in January meant that they were already behind by a whole semester. The development team had to work tirelessly to put all this functionalities in place in the USIS for it to serve its purpose, which is, meeting the specific needs of the university. These examples are to support the claims in literature that stabilization are usually not achieved in the development of an IS, because the information needs of its use are dynamic (Williams & Edge, 1996).

6.5 Closure

In the case of UPSA closure was usually attained either by the word of an authority, by the application of existing policies or by negotiation, however it cannot be said that the USIS has reached full closure due to the differing views and opinions that arises as users try to solve problems that are identified in the course of using it. Any closure that was reached was temporal as there were frequent enhancements, updates and revisions made to the system. A typical example of how an issue on the examination module was closed is presented next. Faculty members wanted every lecturer who taught a course to enter the grades himself or herself, whereas administration thought that, using a lead lecturer concept where one person was to claim responsibility for everything was the best. Going back and forth the ISTD always sided with the administration, to tailor USIS the way

they wanted it. When asked during an interview, why the ISTD always sided with administration, the deputy director in charge of I&E said that:

> *"It was mainly because I think they were more justified on many occasions because anytime such a conflict came up you realize ultimately, it is the administrators who make most use of the system. It is as if the faculties only use it periodically but administrators use it almost every time throughout the academic year. We thought responding to the needs of the administrators will make the system more efficient than the faculties, but that's not to say we totally neglect the faculties perception, we always think of how we can also make them happy in the future, but in the short term, we satisfy administrators."*

Handling of Incomplete Grades (ICs) was another example of an issue that attained closure in USIS. Some faculties were of the view that resolving ICs should be done by the exams office (administration), whilst exams office also thought that it should be done by the faculty. So this resulted in a conflict that took the intervention of the pro-vice chancellor, who called a meeting involving ISTD, exams office (administrative staff), faculty and deans (faculty members) to discuss the issues of pending ICs. Eventually after several meetings concerning this issue, it was resolved that faculties should deal with any issues regarding ICs. Although this resolution was not favored by faculty members, closure was attained which led to the stabilization of the functionality that was to handle ICs on USIS. In a nutshell, although closure is attained on some aspects or modules of the USIS, it does not mean that the entire system has reached closure, which is even not a possibility in the case of UPSA because, requirements and information needs keep changing that in turn give rise to social and political conflicts within the University.

The case of VVU is analogous to that of UPSA. i-School could not be said to have reached full closure as there were constant revisions and modifications to it. This revisions and update were sometimes based on requests made by the stakeholders of the University, who felt that a particular feature of the system should work differently. The decisions to implement the requested modifications are usually vetted on and authorized by management of VVU based on existing policies or what they believed would work best for the University. Of course these decisions are met with several oppositions by other stakeholders who had alternative views of how the system should work, but the conflicts were always resolved either by negotiation, best-practices, organizational strategies and authorities. From both cases it is evident because large IS are implemented in academic institutions that usually have dynamic information needs, it is almost impossible to attain full closure, as the dynamism would stimulate contrasting views from the RSGs which in settling mostly require the destabilization of the IS.

6.6 Chapter Summary

This chapter performed a comparative analysis on the findings of the study to identify emergent themes and concepts which will be used for discussion in the chapter seven. The analysis was done based on SCOT which was used as the theoretical lens for this study. The next chapter presents a discussion on the themes and concepts that were identified in this chapter in relation to this study.

CHAPTER SEVEN

DISCUSSIONS

7.1 Introduction

This research is aimed at understanding why and how higher education institutions in a sub-Saharan African country choose bespoke software over packaged proprietary software. The previous chapter presented evidence of how two higher education institutions in Ghana carried out a bespoke software development and implementation. In order to address the research questions posed in chapter one, the chapter drew on SCOT to analyse the findings that was presented to reveal the RSGs that influenced the construction of the systems, their different views, opinions and perception of the systems, how these different views were resolved and consolidated and the resulting consequences. This chapter discusses the analysis of the findings and the reviewed literature in order to address the research questions for this study. The chapter begins by discussing the rationale for the development of bespoke software for higher education institutions. This is then followed by a discussion on the process of developing and implementing bespoke software in higher education institutions. Subsequently the consequences, both intended and unintended of deploying bespoke software in higher education institutions are also discussed. The chapter then concludes by presenting the researcher's views on the use of SCOT in the context of this study.

7.2 Rationale of Bespoke Software Development

As Yeo (2002) noted, IS acceptance goes beyond its usefulness and technical quality to engulf complex issues such as the social and cultural composition of the organization in question, including politics in the management of information. The analysis of the

research findings indicate that the case Universities in this study, not only based their choice of bespoke software development on technical issues but also took into consideration social and cultural factors of their respective institutions. More so, the reasons for a public University's choice to develop and implement a bespoke software are not far-fetched from that of a private University. The reasons gathered include cost effectiveness, tailored solutions, minimal downtime from technical error and capability building. Each of these rationales is discussed in detail in the following sections.

7.2.1 Cost effectiveness

Findings from this study suggest that organizations opt for bespoke software solutions to cut down on cost. higher education institutions are usually limited in terms of IS budget and as a result, senior managers are unhappy about the huge amount of money spent on the implementation and customization of packaged software and the increasing cost incurred any time upgrades have to be made to the software (Courant & Griffiths, 2006; Esteves & Pastor, 2005; Khelifi et al., 2009). The increasing cost of using the existing software (OSIS) to manage just a few of the activities of UPSA, drove managers of the universities to settle for in-house developed bespoke system. UPSA as at 2013, was incurring an annual cost of approximately $160,000 to its software vendor. This amount however, was to increase yearly depending on the size of the database used, which was not favorable to the University, because of the expected annual increase in student population. Apart from the annual cost that was a huge burden on the management of the University, there were also costs of technical support services and modifications to the software which significantly escalated the total annual budget towards the software, which to the dismay of management could not be determined from the onset. Developing an in-house bespoke system was considered cost-effective as it eliminates the increasing

yearly cost of managing information. Again since the developers are staff of the University, making modifications and upgrades to the system would not attract additional cost as staff are entitled to fixed salary and their jobs require them to make enhancements to the system. VVU, on the other hand, had no experience with any technological solution because the previous system in place was a manual one, but the cost of customizing packaged solutions and making modifications to suit the needs of the University propelled the choice of in-house bespoke development. The implementation of packaged software has been cited as having higher cost up front which informed the decision of VVU on a bespoke solution (Esteves & Pastor, 2005; Lupu et al., 2008).

7.2.2 Tailored Solutions

higher education institutions have unique ways of carrying out their processes and so information needs are not fully met by commercial software products (Khelifi et al., 2009; Lupu et al., 2008; Sagitova, 2012; Supriana & Permatasari, 2007). Courant and Griffiths (2006) attribute this to the fact that commercial software developers are distant from the actual users of the system, as a result do not fully appreciate the way and manner processes are carried out, and requirements for higher education institutions software. Most managers and leaders of higher education institutions are therefore dissatisfied with the performances of packaged solutions because they are not well tailored to suit their needs (Courant & Griffiths, 2006; Irani et al., 2001). Lupu et al. (2008) observed a powerful trend of the dominance of bespoke software development in Universities and attributed it to the fact that existing packaged software do not have adequate functionalities that meet their peculiar needs. The case is no different for UPSA, as OSIS could not accommodate the idiosyncrasies of the institution. Although OSIS had several modules peculiar to higher education institutions, UPSA only used the course registration

module, the students' record management module and the examination, because they considered the other modules not useful. As a growing university, a lot of changes were effected in the curricula of the University which became increasingly difficult to effect in the existing system even with the assistance of the software vendor. For instance because of the annual increase in students population, it became apparent that a weekend school that operates a trimester system be instituted to accommodate more students. This meant that the system must be able to manage regular students differently from weekend students especially in terms of course registration. This requirement was communicated to the vendor and the necessary conditions were set in the source codes to cater for it after several weeks. Several similar requests are made to the software vendor, which was always done at a cost hence informing the choice of bespoke solutions. It was believed that making modifications to the functionalities of the system would be much easier if the developers were in-house and much closer to the users of the system. Even though VVU did not have any previous experience with a technological system, the demonstrations of the application from potential vendors to the management of the university was enough to convince the managers to choose an in-house developed bespoke solution. As Irani et al. (2001) noted that bespoke software development was considered more suited to the idiosyncrasies of organizations' business processes, management of VVU realized that the software packages that were demonstrated would not fully meet the needs of the University, and therefore opted for a bespoke development. In order for their needs to be met, a lot of changes would have to be effected in the system, which in the long-run was going to make the system very expensive to manage. According to Courant and Griffiths (2006), commercial software developers do not appreciate the different ways in which software is used in higher education institutions because of the distance between them and users of a software and therefore

are not better able to meet the needs of the HE sector, hence the choice of a bespoke technological solutions by the case institutions in this study.

7.2.3 Minimal Downtime from Technical Error

From the findings of Effah and Abbeyquaye (2013), it was evident that the development of OSS application by the IS staff in collaboration with consultants ultimately eliminated the need to wait for technical errors to be resolved by software package vendors. This signifies that IS managed in-house has the potential to reduce or eliminate downtime resulting from technical errors. As in the case presented in Effah and Abbeyquaye (2013), having a bespoke software developed in-house allowed for the quick resolutions of technical issues in the case of UPSA. The use of OSIS introduced unnecessary delays and downtimes in their work processes, as a result of the inability of software vendor to resolve technical issues relating to the software. Because the vendor had other clients, getting the urgent support of the vendor on technical issues was not realistic, as a result forced users of the system to wait or look for alternative means to proceed with their work until the vendors had the time to attend to them. The University was always at the mercy of the vendor. In the case of VVU, the possibility of having to wait all the time for the vendor to resolve their technical problems was considered and so management thought that having an in-house developed bespoke solution would be the best option.

7.2.4 Capability Building

Regarding capability building, the literature on bespoke systems did not include capability building as part of the rationale behind an organization's choice for a bespoke software, however, the findings of this study shows that the need for higher education institutions to develop technological capabilities also contribute to the choice of bespoke

software development. Effah and Abbeyquaye (2013) study revealed that IS/IT staff developed adequate experience and capacity to provide full support for IS without the need to consult experts. The experiences were gained during the use of the IS as IS staff tend to resolve technical issues at source code level and participating in online open source communities. Analysis of findings revealed that, members of the case institutions in this study who were technologically inclined took up the challenge to develop an Information System for their respective institutions, because they believed that they could do it. Knowing the way and manner that processes and operations are carried out within the Universities and the challenges that are faced, these members who saw that going in for a generic software would not eliminate all the problems, took the initiative to develop a tailored system in-house that would be managed by staff of the Universities. This point is in agreement with the findings of Courant and Griffiths (2006) that many higher education institutions are driven to develop their own solutions because they have significant professional resources that are somewhat familiar with the particular needs of the Institution and that of Fowler and Gilfillan (2003) which says that bespoke creations in universities harness computational expertise within the institution. In the case of UPSA, this rationale emerged from the school of thought that held that technology should radiate from the Universities to the industry and not vice versa, and as noted by Courant and Griffiths (2006) that:

> *"…with the exception of the information technology industries themselves, higher education has more internal capacity to provide for itself than do other industries".*

Similarly, in the case of the VVU, this rationale emerged out of an interesting phenomenon. The University president got to know that some of his students formed part of the development team of the software company they had wanted to award the contract

of development to, as a result he thought that if students, who were interns at that software company could build such a software, then with the right kind of facilities and environment, the system can be developed in-house and ultimately capabilities would be built. The computer science department students of the University are taught software development as part of their courses and they are made to develop workable applications, so they already have some experience. Therefore together with their course instructors who have hands-on experience in this field to guide them, they already form a professional resource to VVU. More so because of their proximity and closeness to the users of the software they were able to easily acquaint themselves to the way things worked around the university and to determine specific requirements of the institution.

Findings from the two case institutions show that, whilst other organizations choose bespoke solutions (either externally developed or in-house developed) because of reasons other than capability development, higher education institutions that choose in-house developed bespoke solutions do so to build up technological competencies, experiences and exposures. Clearly, capability building and development is an important benefit of in-house development of bespoke solutions to higher education institutions (Courant & Griffiths, 2006; Effah & Abbeyquaye, 2013; Fowler & Gilfillan, 2003), however it is also needful for researchers to recognize that for higher education institutions, it forms part of the reasons for choosing bespoke solutions.

7.3 Bespoke Software Development Process

The analysis of the research finding highlights several activities in the development process that are worthy of note. These include the source of initiative, approval of the initiative, formation of the development team, requirement gathering, choice of technological platforms, deployment and maintenance of the bespoke systems.

7.3.1 Source of Initiative

Salimi and Dankbaar (2008) noted that the implementation of IS usually results from top management's decision in some organizations and a drive from the ITD in other organizations. It is evident from the analysis of both cases in this research that developing an in-house bespoke solution for the case Universities was an initiative of their respective technical teams. In relation to existing literature, there is an inference that the initiative of bespoke IS comes from the technical team of an organization (Irani et al., 2001). Management of both Universities identified the need for an information system that will serve the institution better and in response to that need, sought for external solutions. Software vendors were invited to demonstrate their products, so that a choice could be made and acquired. Although both case institutions did not have development teams, some members of staff who were knowledgeable in IS and thought that the purchase of a software package would be a waste of money decided and made an appeal to management to develop a bespoke software that would meet the needs of their respective institutions. Initially, the request to develop an in-house bespoke solution was turned down by management of both cases for reasons such as the unavailability of development teams, and the lack of trust in their capabilities, if even there were any. Contrary to the findings of Irani et al. (2001) where the manager was against the development of bespoke solution due to its greater implementation cost as compared to packaged system, findings of this study indicated that managers were much more concerned about the success and performance of the development. In both cases it was clear that management did not consider the possibility of having a system developed by the technical departments in-house and would have purchased any one of the software packages available for purchase if the technical team had kept mute. The analysis of the cases in this study indicates that developing in-house bespoke solution results from the technical team's initiative and

willingness of an organization to offer a better solution than vendor software packages have to offer. This finding is consistent with that of Irani et al. (2001) and Salimi and Dankbaar (2008). The analysis also revealed that the desire to have a department full of staff that are skilled enough to handle any technological problems of an organization without having to outsource gives rise to the initiation of in-house bespoke development of software. This research adds to existing knowledge since it not only provides the merits of the software which is well researched but traces the reasons for its development.

7.3.2 Approval of the Initiative

Information System is considered a product of the coalition of stakeholders (Yeo, 2002) which implies that it requires the support and commitment of promoters within an organization to be implemented successfully. According to Salimi and Dankbaar (2008), the drive to implement a particular information system is usually as a result of top management's decision or a push from the IS department of an organization. Findings from this study indicated that the drive to develop a bespoke software came from the technical departments of both case institutions, however, the systems wouldn't have become a reality if they had not gain the support and approval of top management of the Universities. The in-house development and implementation of bespoke software in both case institutions wasn't a direct decision. It took some back-and-forth process to finally settle on in-house development of the information system they needed. The case of VVU's approval of the development of i-School was mainly as a result of a demonstration of the i-School prototype to the stakeholders. A lecturer from the CSD of VVU negotiated with the stakeholders to give him two weeks to form a team and come up with a prototype of the proposed software, which was done and appreciated. The University president's visit to a software development company also motivated the

approval of the development of i-School. This was because he discovered that some of his students make up the development team in that software company and so immediately after the incident, gave approval for a technical team to be formed for the development of the system, although some stakeholders did not agree to this choice. In the case of UPSA, however, the development of USIS was not approved at all. A development team was formed regardless to proceed with its development. The development team capitalized on the lengthy procurement process that existed in public institutions to develop USIS, hoping that it would be developed to a stage that it can serve the University before the procurement process of the software package is completed. A demonstration of the first functional version of USIS was made to the RSGs of UPSA, which gained their admiration and hence its approval by management. The approval process of these bespoke systems goes to show that for in-house development of bespoke systems to be implemented, members of the technical team must prove their worth, by having something to show, in order to gain the trust of users of the system.

7.3.3 Formation of the Development Team

The development of IS may require the acquisition of more skill set and this could be obtained through the following means; sending existing staff off for external training; employing skilled and experienced IS professionals; or recruiting students with knowledge in either software engineering, IS and computer science who are on industrial attachment (Irani et al., 2001). It can be observed from the analysis of the research findings that both Universities formed development teams to carry out the analysis, design, development, implementation and maintenance of the bespoke systems. UPSA then had an IT department which was transformed to a directorate (ISTD) but only with staff members who had core competencies in networking. The decision of management

of ISTD to develop USIS required that a development team be forms for this purpose. Similar to the findings presented from the study of Irani et al. (2001), the case organization, did not have adequate human resources with the required skill set to carry out the development, hence had to send staff out for training as well as recruit experienced professionals and knowledgeable students. Contrary to the case of UPSA, the case organization did not have any challenge with staff recruitment. UPSA, being a public institution had a challenge with an embargo placed on it by the government to recruit new staff, because it had exceeded its staff recruitment quota for that period, as a result did not have the capacity to employ any software developer. The management of ISTD then decided to form the team by calling upon University graduates who had just completed their national service and have still not found jobs and students in the field of technology relevant to IS who were willing to work with the directorate on industrial attachment. The call yielded responses and the respondents were each interviewed and examined by way of developing simple applications to justify their inclusions in the team. This recruitment was done at the level of ISTD and not the University's human resource level. The management of ISTD encouraged and motivated the newly formed team by giving them stipends from their own coffers at the end of each month just so the development of USIS could be materialized. VVU, unlike UPSA, did not have any IT department to boast of, however because the University offered a computer science program, the team formed was made up of two lecturers of and some very good students of the computer science departments who voluntarily worked hard to see the development of i-School become a success.

7.3.4 Requirement Gathering

Evidently from both cases requirement gathering was not a major phase of the development of the bespoke systems in the institutions as opposed to what has been commonly established in previous studies (Alves et al., 2007; Engu et al., 2012; Saiedian & Dale, 2000). Considering the manner in which UPSA's development team embarked upon the development of USIS, gathering user requirements did not really contribute to its design. The team attempted to gather requirements from system users and stakeholders, however because the development of the system was not approved, there was no ultimatum to continue with this exercise, even as users and stakeholders' cooperation was minimal. This resulted in the development team's resort to the use of best practices from benchmarking systems that they acquired from online and believed that it would best suit the needs of UPSA. However, when management of UPSA gave approval for the USIS to be used, the team gathered requirements from system users to fine tune the system to better suit their needs. The initial development of i-School was done without the gathering of requirements. The development team relied on the knowledge and experiences they had about the University and the way processed were carried out to invent user requirements. Alves et al. (2007) and Engu et al. (2012) noted that the distance between software developers and users of a system in a package software development usually result in the invention of requirements. Findings from the VVU case however suggest that the invention of requirement is also possible in bespoke software development. The development team considered the requirement gathering process as not resourceful, considering the limited amount of time they had to demonstrate what they can do to management of the University. Although there were some implications later on, both teams did develop functional systems that subsequently led to the acceptance of bespoke solutions in the case institutions. This implies that where

the software developers are familiar with the operations of the organization whose software they are developing, the development process may commence without requirement gathering. This research therefore highlights the importance of proximity to producers and consumers of software.

7.3.5 Choice of Technological Platform

This section discusses the findings on the choice of technological platforms used in both case institutions. The existing literature on open source in developing countries so far gives the impression that organizations adopt open source tools because if offers control over the implementation of software functionality to meet specific needs and liberty to use and distribute a software without having to worry about licensing issues (Gupta, 2012; Steiniger & Hay, 2009). In the case of UPSA, open source tools came in handy when the challenge of the process of public procurement was posed to the management of the institution and the technical team. Management of the University had started the procurement process of an information system, they believed would meet the needs of the institution, however because of the lengthy processes of the public procurement system, it meant that the process could take at least 8 months for the system to be finally procured, which implied that management would bear the cost of using the existing system for at least one more year. It is important to note that the implementation for a new information system was driven by the need of the University's management to cut down on the annual cost of using the existing system which was becoming more expensive because of the rising students' population. The technical team saw the possibility of the new system not being procured immediately and therefore took advantage of this to start developing the USIS which it believed would be ready before the procurement process was completed.

Open source tools came in handy in the following ways:

i) Since management of UPSA did not trust the capability of the technical team and therefore were not willing to acquire proprietary software tools for the development of the new system, the technical team saw open source tools as a great advantage to demonstrate their technical capabilities. This finding goes to prove the claim of Khelifi et al. (2009) that open source solutions are preferred to packaged solutions in developing countries where limited budget are allocated to IT solutions.

ii) The lengthy procurement process of proprietary software tools as against the free and readily available open source tools was also the reason to adopt open source tools.

It is evident from the above that, in the case of UPSA, the reasons for adopting open source tools were largely based upon the cost and procurement process of proprietary software tools and the desire of the technical team to showcase their technical competence.

In contrast to UPSA, VVU largely adopted proprietary software tools due to its perceived robustness and also due to the fact that the technical team were more comfortable with the selected proprietary tool as against the available open source tools. Also, since the University is a private institution, the public procurement law/processes does not apply to them and for that matter, there was no risk of delay in the implementation process that resulted from the long procurement processes in public institutions.

In sum, the findings from the study show that organizational culture plays a crucial role in the choice of technologies for information system development. The procurement process, for example is an organizational culture that greatly influenced the choice of technological platform especially with regards to UPSA, adding to the pool of IS

knowledge that social and organizational issues are also crucial to the choice of technologies to use in development.

7.3.6 Development

The literature reviewed in this study concerning IS development indicated that developers usually seek for the assistance of consultants so as to obtain successful results (Effah & Abbeyquaye, 2013; Irani et al., 2001). Findings from this study however do not establish the need for a consultant in ISD. Although the development team of both case institutions were newly formed and had virtually no experience, the bespoke IS development projects were successful, even without the help of a consultant. The development teams, however relied on existing knowledge. In the case of UPSA, where open source tools were employed for development, developers relied heavily upon OSS community forums and discussions for assistance (Effah & Abbeyquaye, 2013; Von Hippel, 2001). More also the team capitalize upon the experience and skills of the directors of the ISTD to develop and implement a successful bespoke IS. Using OSS tools enabled USIS to be built without license fees (Khelifi et al., 2009). Developers from VVU on the other hand employed the use of proprietary tools for their initial development and relied upon available online manuals and knowledge of the lecturers who had prior experience working with it. The use of proprietary tools posed some restrictions in the distribution of the software built as the software could only be installed and configured on three computers (Steiniger & Hay, 2009). As defined by Bijker (1993), the technological frame concept in SCOT is repository of knowledge, cultural values, goals, practices, and exemplary artefacts shared by a social group, and structures their attributions of meaning to objects and processes in technical innovation, and their subsequent actions. The cultural practices of UPSA, that is, the acquisition of items

through a structured and rigid procurement process led to the choice of OSS tools for bespoke development.

As noted in literature, ISD creates a platform for IS staff to develop their capabilities, the development teams of both cases considered their experience as a learning curve and have acquired new skills and experience (Effah & Abbeyquaye, 2013). Even though in the case UPSA, there was an opportunity for the developers to go for external training for the sake of the development project, time however did not permit them to have the training, as it was going to cause a delay in the deployment of the USIS. This also informed their choice of open source tools for development, since tutorials and majority of assistance needed were available online. In the case of VVU, there was no consideration for external training for the developers, the lecturers involved were experienced enough to see the development process through.

VVU had no reason to recruit students with knowledge relevant to ISD, since the University offered a computer science program and thought it wise to use and develop their own students, unlike the case of UPSA which is similar to that case presented by Irani et al. (2001), students with knowledge relating to ISD were recruited from other institutions as the University did not offer any IT/IT related program, hence did not have students in that field.

Findings from the cases presented in this study, indicated that there was no prior training or education to staff of the Universities before the development commenced, as a result users were ignorant about how their job functions could be affected by the implementation of the bespoke IS. According to (Irani et al., 2001) prior education and

training about the benefits and implications of IS organization staff before development is crucial to its success. The failure of the UPSA case to train and pre-inform their staff members led to the initial resistance of users to the USIS, however, upon persuasion and coercion by senior management, the system became appealing to the staff on their first use.

In the hope to align USIS to best-practice HEIS, the development team tailored USIS after the characteristics and functionalities of other HEIS they had identified as bench-marking systems, however findings from Irani et al. (2001) show that the case organization re-engineered the company's business process to model the bespoke ISs to best practice jobbing shop process.

7.3.7 Implementation

In terms of implementation, analysis from the research findings point out that both UPSA and VVU had their systems implemented a modular basis. The modules that were most needed were first implemented after several tests had been carried out on them locally. Users of successful IS continually measure and control its performance and effectiveness as well as appropriately managing human and organizational issues that arise effectively (Irani et al., 2001). In the case of UPSA, the major reason why USIS implementation was successful was as a result of its continuous evaluation by an audit team that was setup for that purpose. The audit team interviewed various users to ascertain their views on the USIS and how it was working for them. USIS was rated 99% successful based upon these evaluations and its implications. This however did not mean that there were no negative implications for using USIS.

The implementation of the bespoke systems no doubt brought about some changes in the case Universities, however Kim and Kankanhalli (2009) noted that providing system users with training, support, guidance and ample time and resources to learn about the new system enhances organizational support for change. Both VVU and UPSA organized training sessions and demonstrations for the system modules before their implementation. Although in the case of UPSA, it was evident that users of USIS were not given ample time to learn about the system before use, which resulted in its initial resistance by users. According to Caldeira and Ward (2003) management of an organization have the authority to influence other members of the organization and is likely to overcome every resistance to change. Evidently both case Universities had the support of management for the bespoke software implementation which reduced the impact of change and user resistance. It is however important to note that having top management support IS implementation does not mean that the rest of the users in the organization are also in support. Where this is the case, users tend to feel that they are being coerced to use a system and therefore left with no choice to raise objections against it. This situation is evident in the implementation of USIS. Users felt that managers compelled them to use USIS because of its financial benefits but failed to consider its usefulness to others. This is consistent with the findings presented in the case of Effah and Abbeyquaye (2013).

7.3.8 Maintenance

For both case Universities, the bespoke systems are constantly maintained and upgraded because of the changing nature of higher education institutions (Schneidewind, 1987). Members of the development team sometimes work at odd hours to ensure that changes required by users are effected and functioning properly. The close proximity of the

development team made it easier for the modifications and upgrades to be done on time, as a result, users of the system always expect that their request of a change or the addition of a functionality granted almost immediately. The development teams however find this unhelpful, as it makes it difficult to control the documentation and roll outs of versions of the system. Based on literature IS maintenance is regarded as the most expensive aspect in the IS life cycle (Bennett & Rajlich, 2000; Dekleva, 1992; Schneidewind, 1987). Contrary to this claims, the case Universities of this study do not find maintenance to be expensive. This is because the maintenance is done by in-house IS staff whose are on a fixed salary, as a result, the continual cost of maintenance is eliminated. In relation to the SCOT theory, the continual maintenance of the bespoke systems to adapt to the changing needs of the RSGs and due to the dynamic nature of the University business processes, makes it impossible for the systems to be stabilized and therefore attain closure. As noted by Schneidewind (1987) that a software undergoes changes throughout its lifecycle, it goes on to oppose the stabilization and closure concepts of SCOT.

7.4 Consequences of In-house Bespoke Software Development in Higher Education Institutions

The analysis of the research findings shows that the development and use of bespoke systems had some consequences, some of which were expected and planned for and others not expected, hence not planned for.

7.4.1 Intended Consequences

Intended consequences according to this study are consequences that the users of the bespoke system saw it coming and as a result had made adequate preparations for. The cases in this study present how the Universities were each affected by the development

and use of the bespoke systems. In the case of VVU, implementing i-School meant that fast internet service is made available to all areas of the campus so that accessing the system wouldn't be a problem. In view of this management of the institution ensured that necessary network devices and resources were procured so that internet can be extended to all areas on campus. The case is also similar for UPSA, although the campus was already equipped with internet facilities, the implementation of USIS required that the internet bandwidth of the University be increased from 15MB to 45MB. This increase was due to the fact that usage on the bandwidth was due to increase as a result of students use of the University's network to do online course registrations and to check their examination results. The existing system did not require the use of much bandwidth as it was only a few administrators and staff who used it for their work. The USIS however, because of its integrated components implied that it would be used by more staff members now from the various departments and directorates including even students. Management of the University made some financial commitments into getting a higher bandwidth for the purpose of the effective and efficient implementation of the USIS. The use of bespoke systems in both institutions evidently had implications on the infrastructure of the University's network.

The analysis of findings also revealed the implications of bespoke systems use on organizational structure. Although in reality, the organizational structure of both institutions had not change, the use of the bespoke systems created that impression. For instance the case of VVU reported that some staff members of the University felt that the use of the bespoke system had made other members too powerful and more important than they ought to be. The challenge was faced especially by the development team of

both case Universities because they were called upon every time to resolve issues of all kinds even if the issues were not related to the bespoke systems.

McGrath (2003) noted that closure cannot be attained in the development of an information system, since the possibilities of interpretive flexibilities are seemingly unending. There were frequent demands for modifications and upgrade of the bespoke systems due to the different and emerging technological frames help by the RSGs. Using bespoke software in both Universities implied that the software developers worked at unusual times to ensure that the system is tailored to suit the needs of its users. Because of the proximity of the developers to the users, also the request for changes seem to be unending. The numerous and in some cases last-minute request for certain modifications and updates usually exposes the system in production to errors and bugs. When requests for upgrades and changes were made so late and therefore demanded in a hurry, developers were usually constrained with time to deliver bug-free solutions. This was because they hardly had much time to build the requested functionality and to thoroughly test it against any errors or bugs before integrating it with the live version of it. Irani et al. (2001) noted that users conveniently apportion blame to an IS and were ready to dismiss it an go back to their old way of doing of life when things go wrong. This assertion is consistent with the findings of the UPSA case, because when users encountered errors during the use of IS as a result of their frequent and urgent demands for modification, they complained that the system was inefficient and advocated that it should be abandoned for the purchase of a packaged software.

In the case of VVU, because the implementation of i-School was their first instance of adopting technological solutions in managing information, of course some people felt

threatened by its use. There were members of staff whose roles were made less significant when i-School was introduced, therefore these members of staff stood vehemently against the system. For instance, the chaplaincy was a stakeholder group that formed part of the registry staff and were responsible for ensuring that all students had been cleared before they could be allowed to register during the use of the Green Card System. However i-School was designed in a way that students were automatically cleared, hence did not need to go through the chaplaincy unless there was the need to do so. The chaplaincy then perceived the i-School as system that made them less important because it had reduced the number of students who patronized their services, thus made them seem less useful to the University. The chaplaincy also claimed that i-School had eliminated the opportunity the officer in charge had to minister spiritually to students, because only few students would have the need to visit his office due to the automatic clearances made possible by the new system.

Analysis of the findings from this research reveals that the RSGs in both case institutions were mainly concerned about the sustainability of the bespoke systems developed. The concern was common mostly to the technical team of both institutions. They had fears of the future regarding the systems continually being in operation when the original developers were gone. In order to ensure that their hard work do not become a thing of the past, the functionalities and features of the systems were constantly reviewed and upgraded to meet the changing demands of users. Detailed documentation of the system's codes was done so that any software developer who goes through the document could easily make desired changes to the codes. More also both case Universities ensured that new developers were trained on the way the systems were designed so that they can easily manipulate and make changes to them. In the case of VVU good students from the

computer science department were identified and made to join the development team so as to gradually learn about i-School and soon be able to work around it to effect any required upgrade. For UPSA, the development team had regular training sessions both within and outside the University so that newer members of the team can gain better understanding of the system in order to be able to effect any changes, in the event that the lead developer was not available. Evidently, the ability of bespoke systems to continue to be in use is of great value to the developers of the bespoke software in this study and findings from both case institutions show how great efforts are made ensure that the systems are sustained through the foreseeable future.

7.4.2 Unintended Consequences

It can be inferred from Nasir (2005) that the implementation of IS can bring about structural adjustments in organization and these adjustments can have advantages and drawbacks. According to Markus and Pfeffer (1983) those who have access to and control the flow of information have power within an organization, more also because IS confers legitimacy and restrictions on staff, it is an indication of the level of power or authority of staff in an organization. The assignment of roles and privileges in the case of UPSA led to the conflict of who should be able to do what on the system. This consequence of the use of USIS was not in the least expected. Lecturers expressed great displeasure at having to go to administrative staff for their accounts to be created on USIS for. They felt that the USIS had reduced their status because they were expected to live by the controls exerted on them by the administrative staff through the time-bound functionalities of USIS. This effect of using USIS was dire so much that other RSGs came to believe that the system was developed only for the administrative staff to make their job easier and to address the concerns of management about the increasing cost of

using the existing system without taking into considerations the needs and views of other RSGs.

7.5 Reflections on the Use of the SCOT Theory

This section presents a reflection of the researcher on the use of the SCOT theoretical framework as the analytical lens for this study. Much of IS literature that make use of SCOT dwelt mainly on issues ranging from the selection, acquisition, implementation and to use of IS in organizations (Howcroft & Light, 2010; Kleinman & Klein, 2002; McGrath, 2003; Prell, 2009; Sahay & Robey, 1996). The application of SCOT in the development process of IS has not been touched on, which is why the researcher chose to use SCOT to analyze the development process of a bespoke software for higher education management in a sub-Saharan Africa context. The benefits and limitations of making use of SCOT in this study is discussed.

Concepts from the SCOT theoretical foundation have increased understanding on the development, implementation and use of an in-house bespoke software. In order to successfully develop and implement bespoke software to aid higher education management, it is important to understand the complexity of the social-technical system that existed in the case Universities.

Identifying the different RSGs in each case presented was necessary in exploring and understanding of the different interpretive flexibilities and technological frames and the use of the SCOT theory helped the researcher to understand how the development process was influenced by these concepts. This increased the appreciation of how these social groups perceived, used and interacted with the bespoke systems. For instance in the VVU

case, students perceived the i-School as system developed to enhance the process of course registration, whereas administrative staff perceived the system to go beyond making course registration easy to making the management of student records effective and efficient. The use of this theory also helped the researcher to understand that, as the information system gets to be used new RSGs emerges, which leads to the formation of new interpretations for the system (McGrath, 2003).

The technological frame concept of the SCOT theory also helped the researcher to understand why the case Universities chose particular technological platforms for the development of the software. The lengthy procurement process in the UPSA case demonstrated how the culture of an organization can shape the construction of a technology. It significantly informed the decision to develop USIS in the first place and also played a role in the choice of technological platform used for the development.

The use of the SCOT theory also guided the researcher in the collection of data, analysis of findings and the discussion. SCOT was however, unable to justify the stabilization and closure of the case systems. Based on the analysis of this study, it was observed that stabilization and closure has not been attained on the bespoke software of both Universities and this is a limitation of the SCOT theory also noted by McGrath (2003) as not being able to account for closure. The emergence of RSGs, hence the advent of new interpretive flexibilities which is informed by the technological frames held by these RSGs and the unique plus dynamic nature of operations in higher education institutions contribute to why closure and stabilization cannot be achieved with the bespoke software of the cases presented in this study (Lupu et al., 2008; McGrath, 2003; Sagitova, 2012; Supriana & Permatasari, 2007).

In all more knowledge has been gained on how RSGs and their different interpretive flexibilities which stems up from the different technological frames held shapes the development of a bespoke software in a higher educational environment.

7.6 Chapter Summary

This chapter discussed the analysis of the findings of this research and the reviewed literature in order to address the research questions posed at the beginning of this study. The chapter began by discussing the rationale behind the choice of bespoke software for the case institutions. Following this was a discussion on the activities involved the in development process of the bespoke software. Next the chapter discussed the consequences of using bespoke software for both case Universities and finally the chapter gave an evaluation of the use of SCOT as the theoretical lens for this study. The next chapter gives an evaluation and a summary of this study and goes on to discuss the contribution of this study to knowledge and its implications to research, practice and policy. Subsequently the chapter discuss the limitations of the study and provides researchers with recommendation for further research and then presents and general conclusion.

CHAPTER EIGHT

SUMMARY, CONCLUSION AND RECOMMENDATION

8.1 Introduction

The previous chapter discussed the empirical discoveries of this study in terms of findings from literature and research questions. The researcher's reflection on the application of SCOT in this study as the analytical lens is then presented. The final chapter presents the summary of this study and its conclusion. This chapter begins by reflecting on the research questions posed in the introductory chapter and how the study sought to address them. Following this, an evaluation of how the research was conducted and its contributions are presented. After, the chapter provides the implications for research, policy and practice. The limitations and recommendations for further studies are presented, after which the chapter presents the overall conclusion.

8.2 Review of Research Questions

This study set out with the aim to understand why and how higher education institutions in sub-Saharan Africa choose bespoke IS instead of packaged proprietary software. Implementing IS in higher education institutions has become a necessary option as it supports all business functions and offer accurate, clean and stable data and available in real time to users across multiple departments. In view of this, predominant emphasis have been laid on the selection, adoption, implementation, use and benefits of packaged information systems for higher educational management (Dutta & Burgess, 2003; Lupu et al., 2008; Pollock & Cornford, 2004, 2005; Ternai, 2003; Zornada & Velkavrh, 2005), thus the development of bespoke software for the management of higher education institutions remains under explored. Moreover the wide application of SCOT in IS

literature has centered mainly on selection and implementation of a technology in organisations without paying much attention to its relevance in bespoke software development (Howcroft & Light, 2010; Kleinman & Klein, 2002; Prell, 2009; Sahay & Robey, 1996; Ternai, 2003).

In response to the literature gap identified in chapter one, the study addresed the following research questions:

i) Why do higher education institutions in sub-Saharan Africa opt for bespoke information system software?
ii) How do higher education institutions in sub-Saharan Africa develop bespoke information Systems?
iii) What are the consequences of using bespoke software?

The research questions were tackled by the study as follows:

i. A literature review of relevant researches on IS, HEIS, ISD, bespoke software, open source software and packaged software was conducted in chapter two to help identify the various gaps that existed in the body of knowledge. The literature revealed that studies conducted on IS in relation to higher education institutions mainly focused on packaged software, thus there is minimal work on the development of bespoke software for higher education management. The chapter also revealed that bulk of HEIS research conducted in the sub-Saharan African context dwelt more on the implementation of ERP and software packages which was why this study, which is focused more on the development of a bespoke software, was limited to the context of a sub-Saharan African country. The chapter also disclosed that bulk of literature that dwelt on the influence of

social factors on technologies, neglected the area of development of rather emphasizing on the areas of selection, adoption, implementation and use of the technology, hence the choice of SCOT that provides a useful platform for understanding the influence of RSGs in the shaping and development of a bespoke software.

ii. Chapter three explored the theoretical foundation of this study. The chapter discussed the basic tenets of the SCOT theory. SCOT was chosen because of its capability to study complex technological innovations and multiple stakeholders' views and how they shape or are shaped during the process.

iii. Chapter four discussed the methodology adopted in conducting this research. The chapter also discussed three commonly debated IS paradigms in terms of their epistemological, ontological and methodological stance and selected the interpretive paradigm as appropriate for this study. The chapter also detailed how the study was conducted on the field, the research method used and how data was collected and analyzed.

iv. Chapter five presented empirical findings from two case Universities in Ghana that have developed and used bespoke software. Chapter five discussed the rationale behind the choice of a bespoke software and narrated the development process of the bespoke software by each University. The implications for using the bespoke software in both case Universities was also presented. The findings revealed that both case Universities choose bespoke software development for

similar reasons. The chapter also showed that in developing bespoke software, a number of activities and processes are involved.

v. Chapter six performed a comparative analysis on the findings of the cases using SCOT, the analytical lens through which this research was viewed, to identify salient issues regarding bespoke development of HEIS. The chapter revealed that the public University (UPSA) bounded by the procurement laws of Ghana, was compelled to use OSS for development whereas the private University (VVU), which did not have such limitations acquired proprietary software for the development of the bespoke software.

vi. Chapter seven discussed the analysis of the findings. In this chapter, the research questions were addressed in light of the reviewed literature in chapter two and the findings and analysis of the case studies presented in chapters five and six respectively. The chapter also discussed the researcher's reflection on the use of SCOT as the analytical lens for this study. In all, the chapter argued that higher education institutions are largely motivated to develop in-house bespoke software by the need to develop institutional capabilities and skills in technology.

8.3 Contribution to knowledge

Walsham (1995c) identified four ways an interpretive research can be generalized and can contribute to knowledge in IS. The four ways are development of concept, generation of theory, offering of rich insight and drawing of specific implications. This study did not develop new concepts, however the study did make some contributions towards

theory. The study also offered rich insights and drew specific implications. The implications of this study are further discussed in section 8.4.

8.3.1 Contribution to theory

Although no new theory was formulated, this study contributes to knowledge in the field of IS by applying an existing theory to generalize the reason behind the choice of bespoke software and the processes involved in the development of bespoke software in sub-Saharan African countries. The study contributes to knowledge by drawing on the SCOT theory to understand the influence of RSGs of an organization on the development of bespoke software. Prior studies that adopted this theory largely used it to analyze the selection, adoption and implementation of packaged software. Less attention was paid to software development and how the concepts from the SCOT theory might impact it.

SCOT theory posits that stabilization and closure are attained for every technological artefact, this study revealed otherwise for bespoke software development. It therefore calls on IS researchers to extend the SCOT theory to account for the non-stabilization and closure of bespoke software. Researchers can also examine other theories that can be used to account for stabilization and closure in a software development process.

8.3.2 Offering of Rich Insights

This study offers deep insight into the development of bespoke software as seen in the cases presented in chapter 5. The study extends the limited scope of HEIS from the dominant focus of ERP selection, adoption and implementation to the development of bespoke software for higher education management.

Previous study related to HEIS have failed to conduct comparative analysis between private and public higher education institutions (Chinyemba & Ngulube, 2005; Dutta & Burgess, 2003; Effah & Abbeyquaye, 2013; Esteves & Pastor, 2005; King et al., 2002; Sagitova, 2012; Zornada & Velkavrh, 2005). This study, however did a comparative analysis on the findings of the cases presented on the bespoke software development process of both the public University (UPSA) and the private University (VVU). The comparative analysis revealed that the public procurement process that exists within public organizations led to the choice of OSS tools for the development of USIS at UPSA. VVU, not bound to any public procurement policy went in for proprietary software tools in developing i-School. This is to say that existing organizational cultures and practices which cannot be easily subjected to change or ignored in the case of public institutions, have influence on the choice of technological platforms used in the development.

It has been argued that interpretive research embraces multiple realities and explains how the construction of reality is shaped by the organizational actors that are directly involved in it (Creswell, 2012; Goldkuhl, 2012; Hirschheim & Klein, 1989). More also, Andrade (2009) noted that the interpretive case study is used to create an integral and persuasive explanation about a phenomena that takes into account each participant's different perspectives. This study has helped in the explanation of the social dynamics of bespoke software development in the context of higher education institutions in a sub-Saharan African country.

In terms of having access to software source codes, there are no major differences between the use of open source and proprietary tools for in-house bespoke software development.

8.4 Implication for Research, Practice and Policy

This study has contributed significantly to research, practice and policy. These implications are presented in the following sub-sections.

8.4.1 Implications for Research

In regard to research, this study brings to bear the need for IS researchers to extend their studies in bespoke ISD beyond the educational sector to other sectors. This implication is considered very important as it extends the current limited scope of IS research from the over-concentrated area of packaged software adoption and implementation to the development of bespoke solutions.

Whereas the SCOT theory posits that stabilization and closure are attained for every technological artefact, this study revealed otherwise for bespoke software development. It therefore calls on IS researchers to extend the SCOT theory to account for the non-stabilization and closure of bespoke software.

8.4.2 Implications for Practice

In terms of practice, the study suggests that for any kind of IS implementation, whether packaged or bespoke, it is important that the technological frames of RSGs must be understood in order for their different interpretive flexibilities to be fully appreciated so that conflicts that emerge in the process can be resolved easily.

Another implication for practice is for careful considerations to be given to the choice and adoption of 'best-practices' and benchmarking systems as these may not be applicable in the context of the organization in question. Also mangers are to ensure that any adopted 'best-practice' offers optimal solutions to the problems identified through requirement gathering.

Again this study identified the need for organizations to build institutional capability and skills through the development, implementation and use of in-house bespoke software. Furthermore, ITD of organizations can initiate the development of bespoke software and form a development team in-house that would work to bring the software to a reality.

More also findings from the study implies that, in order for an effective maintenance of bespoke software developed in-house, managers are to develop strict policies and ensure that these policies are applied, so that upgrade and modification would be done in a more formal and structured way.

8.4.3 Implications for Policy

In regards to policy, this study advocates for government to institute new policies or amend existing policies regarding public procurement, to make it more flexible to acquire software.

8.5 Limitations of the Study

There were some identified limitations with this study which can offer directions for further research. This study focused on bespoke software development in higher education institutions, an educational sector, findings from this study may not be

applicable to other sectors, but can however offer a general understanding of bespoke software development. Furthermore, this study when viewed through another analytical lens, may yield different results, therefore limiting the application of its findings to studies of this nature that failed to use the SCOT theory. Moreover, the choice of case study as the method of data collection did not offer the researcher an opportunity to solve actual problems, however it did provide a holistic understanding of the in-house development of bespoke software. Employing a different qualitative research design, such as, action research can help the researcher solve actual problems.

8.6 Future Research Directions

This study offers a number of further research possibilities that are particularly remarkable, significant and can contribute to knowledge in the field of IS.

i. Since higher education institutions differ in structure and organizations, this study can further be extended by researching into how bespoke software development is conducted in much older and bigger higher education institutions with a wide array of departments and colleges. This will offer more generalizations on the development of bespoke software for higher education management.

ii. The contribution of this study lies within the scope of the development, implementation, use and maintenance of bespoke software. The study can be further extended by researching into the decommissioning process of a bespoke software that is beyond the scope of the theory used in this study.

iii. This study can also be extended beyond the scope of higher education institutions by researching into how organizations in other industries carry out bespoke software development.

iv. Other researchers can also extend this study by examining the security of bespoke information systems.

v. Other theories can be used in future research to unearth pertinent issues in bespoke software development and to offer more generalizations to the findings of this study.

8.7 Conclusion

This study began with the aim to understand why higher education institutions in a sub-Saharan African Countries decide to design and implement a bespoke information system in-house to manage information; how they carry out the entire development and implementation process; and the consequences faced as a result of using the in-house developed bespoke systems. The research therefore traced the experiences of two higher education institutions and their success at developing and implementing bespoke information systems in-house in a sub-Saharan African country context of Ghana. Previous studies on HEIS have focused mainly on the selection, adoption and implementation of ERP packages in higher education institutions. Therefore studies on the in-house development of bespoke systems for higher education management in sub-Saharan African countries remain under-explored. Although SCOT has been widely used in research to understand the influence of socio-technical factors on the development and implementation of IS in organizations, it has not dominated in the context of bespoke software development for higher education management.

In response to these identified literature gaps, this study drew on the interpretive case study approach and the SCOT theory to investigate the process by which bespoke IS are developed in-house for higher education management, so that the research questions of

this study can be addressed. Based on the findings from the case Universities, the research provided a rich insight into the birth of their bespoke systems, by first looking at their motivations for opting for in-house bespoke solutions; the process involved in the development; and the consequences for using in-house bespoke solutions.

To sum it up, the study revealed that in higher education institutions, apart from the cost effectiveness and tailored solutions of bespoke development, the motivation for developing bespoke solutions in-house stemmed up from the need to develop capabilities and skills of their technical staff, so that their information needs can be catered for in-house. The study also showed that as complex as the in-house development of bespoke systems may be, it can serve its purpose of meeting the specific need of the institution is tailored for, so long as there is the social and technical forces that exists within an organization are not detached from each other. The study further revealed, as opposed to findings from literature that argue that bespoke software development is not cost effective (Langat & Kamuren, 2013), higher education institutions find it more cost effective than implementing packaged software. Bespoke software developments enable the avoidance of modification and upgrades cost, which cannot be easily determined from the onset, hence budgeted for, because of the dynamic nature of higher education institutions. The study also pointed out that, private higher education institutions are at liberty to choose any technological platform suitable for development since they are not bounded by any procurement laws, whereas for public higher education institutions, the choice of technological platforms can be influenced by existing organizational practices which cannot be easily modified. Also, in terms of having access to software source codes, there are no major differences between the use of open source and proprietary tools for in-house bespoke software development.

REFERENCES

Abdallah, S. A., Zawiyah, M. Y., & Juzaiddin, A. A. (2013). The Impact of Information System Success on Business Intelligence System Effectiveness. *Journal of Theoretical and Applied Information Technology, 50*(3), 512-522.

Adewole, E., & Fakorede, S. (2013). Strengthening the Nigerian Higher Education System Through the Use of Information Communication Technology. *International Journal of Social Sciences & Education, 3*(4).

Afolayan, O. A., & Absalom, E. E.-S. (2010). Design and Implementation of Students' Information System for Tertiary Institutions Using Neural Networks: An Open Source Approach. *International Journal of Green Computing (IJGC), 1*(1), 1-15.

Alenljung, B., & Persson, A. (2008). Portraying the practice of decision-making in requirements engineering: a case of large scale bespoke development. *Requirements engineering, 13*(4), 257-279.

Alter, S. (2003). 18 Reasons Why IT-Reliant Work Systems Should Replace" The IT Artifact" as the Core Subject Matter of the IS Field. *Communications of the Association for Information Systems, 12*(1), 23.

Alves, C. F., Pereira, S., Valença, G., Pimentel, J., & de Andrade, R. V. (2007). *Preliminary Results from an Empirical Study in Market-Driven Software Companies.* Paper presented at the WER.

Anaper, C., Nihan, S., Ulucay, D., & Cabuk, A. (2013). Accreditation of Online and Distance Learning Programs: Online GIS Education Program Experience. *Turkish Online Journal of Distance Education, 14*(1), 231-244.

Andrade, A. D. (2009). Interpretive research aiming at theory building: Adopting and adapting the case study design. *The Qualitative Report, 14*(1), 42-60.

Aziz, H., Gao, J., Maropoulos, P., & Cheung, W. M. (2005). Open standard, open source and peer-to-peer tools and methods for collaborative product development. *Computers in Industry, 56*(3), 260-271.

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The Qualitative Report, 13*(4), 544-559.

Bennett, K. H., & Rajlich, V. T. (2000). *Software maintenance and evolution: a roadmap.* Paper presented at the Proceedings of the Conference on the Future of Software Engineering.

Biggerstaff, T. J. (1989). Design recovery for maintenance and reuse. *Computer, 22*(7), 36-49.

Bijker, W. E. (1993). Do not despair: there is life after constructivism. *Science, Technology & Human Values, 18*(1), 113-138.

Bijker, W. E. (1997). *Of bicycles, bakelites, and bulbs: Toward a theory of sociotechnical change*: MIT press.

Bijker, W. E. (2001). Understanding technological culture through a constructivist view of science, technology, and society. *Visions of STS: Counterpoints in science, technology and society studies*.

Bloomfield, B. P., & Danieli, A. (1995). The Role of Management Consultants in the Development of Information Technology: The Indissoluble Nature of Socio-Political and Technical Skills. *Journal of Management Studies, 32*(1), 23-46.

Boahene, M., & Bowles, S. (1999). *Information systems development methodologies: Are you being served.* Paper presented at the Proceedings of the 10th Australasian Conference on Information Systems.

Bowen, E., Price, T., Lloyd, S., & Thomas, S. (2005). Improving the quantity and quality of attendance data to enhance student retention. *Journal of Further and Higher Education, 29*(4), 375-385.

Brehm, L., & Markus, M. L. (2000). *The divided software life cycle of ERP packages.* Paper presented at the Proceedings of 1st Global Information Technology Management (GITM) World Conference.

Brey, P. (1997). Social constructivism for philosophers of technology: A shopper's guide.

Brynjolfsson, E., & Hitt, L. (1996). Paradox lost? Firm-level evidence on the returns to information systems spending. *Management science, 42*(4), 541-558.

Bumbac, R. P., & Orindaru, A. (2014). Is E-Learning a Genuine Setting Changer For the Education Field? *eLearning & Software for Education*(2).

Burstein, F., & Gregor, S. (1999). *The systems development or engineering approach to research in information systems: An action research perspective.* Paper presented at the Proceedings of the 10th Australasian Conference on Information Systems.

Caldeira, M. M., & Ward, J. M. (2003). Using resource-based theory to interpret the successful adoption and use of information systems and technology in manufacturing small and medium-sized enterprises. *European Journal of Information Systems, 12*(2), 127-141.

Carmel, E., & Sawyer, S. (1998). Packaged software development teams: what makes them different? *Information Technology & People, 11*(1), 7-19.

Chen, C. (2011). Quantitative Methodology: Appropriate use in Research for Blind Baseball Ergonomics and Safety Design. *Journal of Human Resources & Adult Learning, 7*(1).

Chinyemba, A., & Ngulube, P. (2005). Managing records at higher education institutions: a case study of the University of KwaZulu-Natal, Pietermaritzburg Campus. *South African Journal of Information Management, 7*(1).

Chism, N., Douglas, E., & Hilson Jr, W. (2010). Qualitative Research Basics: A Guide for Engineering Educators. *National Science Foundation2008*.

Ćirić, Z., & Raković, L. (2010). Change Management in Information System Development and Implementation Projects. *Management Information System, 5*(1), 23-028.

Corbin, J., & Strauss, A. (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory*: Sage.

Courant, P. N., & Griffiths, R. J. (2006). Software and collaboration in higher education: A study of open source software. *New York: Ithaca. Retrieved January, 30*, 2009.

Creswell, J. W. (2012). *Qualitative inquiry and research design: Choosing among five approaches*: Sage publications.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*: Sage.

Cronholm, S., & Hjalmarsson, A. (2011). Experiences from sequential use of mixed methods. *The Electronic Journal of Business Research Methods, 9*(2), 87-95.

Dafoe, A. (2005). *Theoretical reconciliation: Selectionism in the social construction of technology and the diffusion of (military) innovations.* Paper presented at the Paper to be presented at the DRUID Tenth Anniversary Summer Conference.

Darke, P., Shanks, G., & Broadbent, M. (1998). Successfully Completing Case Study Research: Combining Rigour, Relevance and Pragmatism. *Information systems Journal, 8*, 273-289.

Dekleva, S. M. (1992). The influence of the information systems development approach on maintenance. *MIS quarterly*, 355-372.

Dutta, R., & Burgess, T. (2003). Prioritising information systems projects in higher education. *Campus-Wide Information Systems, 20*(4), 152-158.

Dutton, W. H., Cheong, P. H., & Park, N. (2004). The social shaping of a virtual learning environment: The case of a university-wide course management system. *Electronic Journal of e-learning, 2*(1), 69-80.

Effah, J., & Abbeyquaye, G. (2013). Social Construction of Open Source Application for Student Records: The Case of Cape Coast University.

Elbana, A. (2013). Top Management Support in Multiple Projects Environment. *European Journal of Information Systems, 22(3)*, 278-294.

Engu, R., Vakkalanka, S., & Kumar, L. A. (2012). Bespoke, Mdre And Challenges In Mdre In Comparison With Bespoke. *Journal of Emerging Trends in Computing and Information Sciences, 3*(2), 173-178.

Esteves, J., & Pastor, J. (2005). Understanding an ERP System Implementation in a Higher Education Institution: A Grounded Theory Approach. *Qualitative Case Studies on Implementation of Enterprise Wide Systems*.

Fowler, A., & Gilfillan, M. (2003). A framework for stakeholder integration in higher education information systems projects. *Technology Analysis & Strategic Management, 15*(4), 468-489.

Gibson, N., Holland, C. P., & Light, B. (1999). *Enterprise resource planning: a business approach to systems development.* Paper presented at the Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on.

Gioia, D. A., & Sims, H. P. (1986). *The thinking organization*: San Francisco: Jossey-Bass.

Goldkuhl, G. (2012). Pragmatism vs interpretivism in qualitative information systems research. *European Journal of Information Systems, 21*(2), 135-146.

Gu, L., Guo, Y., Ruan, A., Shen, Q., & Mei, H. (2010). *SCOBA: source code based attestation on custom software.* Paper presented at the Proceedings of the 26th Annual Computer Security Applications Conference.

Guba, E., & Lincoln, Y. (1994). Competing paradigms in qualitative research. *Handbook of qualitative research, 2*, 163-194.

Guinan, P. J., Cooprider, J. G., & Faraj, S. (1998). Enabling software development team performance during requirements definition: A behavioral versus technical approach. *Information systems research, 9*(2), 101-125.

Gunasekaran, A., & Ngai, E. W. (2004). Information systems in supply chain integration and management. *European Journal of Operational Research, 159*(2), 269-295.

Gupta, P. (2012). User Friendly Open GIS Tool for Large Scale Data Assimilation-a Case Study of Hydrological Modelling. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1*, 427-430.

Haab, M. J., & Cramer, S. F. (2010). Enterprise Resource Planning Systems in Higher Education. *Technology Integration in Higher Education: Social and Organizational Aspects*, 236.

Hardcastle, E. (2008). *Business Information Systems*: Bookboon.

Harsh, S. B. (1998). Agricultural information systems: current applications and future prospects. *Agricultural Information Technology in Asia and Oceanic*, 1-8.

Hekkala, R., & Urquhart, C. (2013). Everyday power struggles: living in an IOIS project. *European Journal of Information Systems, 22*(1), 76-94.

Hippel, E. v., & Krogh, G. v. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science, 14*(2), 209-223.

Hirschheim, R., & Klein, H. K. (1989). Four paradigms of information systems development. *Communications of the ACM, 32*(10), 1199-1216.

Howcraft, D., & Light, B. (2010). The Social Shaping of Packaged Software Selection. *Journal of the Association for Information Systems, 11(3)*, 122-148.

Howcroft, D., & Light, B. (2010). The social shaping of packaged software selection. *Journal of the Association for Information Systems, 11*(3), 2.

Humphreys, L. (2005). Reframing social groups, closure, and stabilization in the social construction of technology. *Social epistemology, 19*(2-3), 231-253.

Irani, Z., Sharif, A. M., & Love, P. E. (2001). Transforming failure into success through organisational learning: an analysis of a manufacturing information system. *European Journal of Information Systems, 10*(1), 55-66.

Jakku, E., Thorburn, P., & Gambley, C. (2004). *Sociological concepts for understanding agricultural decision support systems.* Paper presented at the The Australian Sociological Association (TASA) Conference Proceedings. TASA, La Trobe University, Beechworth.

Jangra, K., & Dua, K. (2012a). Health Care Management Information Systems with Object Oriented Methodology. *International Journal of Information Technology, 5*(1), 234-239.

Jangra, K., & Dua, K. (2012b). Processing of Management Information System with In Small and Large Scale Enterprises. *International Journal of Information Technology, 5*(1), 224-226.

Jebreen, I., Wellington, R., & MacDonell, S. G. (2014). Understanding Feasibility Study Approach for Packaged Software Implementation by SMEs *Information System Development* (pp. 29-43): Springer.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. *Educational researcher, 33*(7), 14-26.

Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in information systems research: a case study. *MIS quarterly*, 571-586.

Ke, W., Tan, C.-H., Sia, C.-L., & Wei, K.-K. (2012). Inducing intrinsic motivation to explore the enterprise system: The supremacy of organizational levers. *Journal of management information systems, 29*(3), 257-290.

Khelifi, A., Talib, M. A., Farouk, M., & Hamam, H. (2009). Developing an initial open-source platform for the higher education sector—a case study: Alhosn University. *Learning Technologies, IEEE Transactions on, 2*(3), 239-248.

Kim, H., & Kankanhalli, A. (2009). Investigating user resistance to information systems implementation: a status quo bias perspective. *MIS quarterly*, 567-582.

Kim, K., Trimi, S., Park, H., & Rhee, S. (2012). The Impact of CMS Quality on the Outcomes of E-learning Systems in Higher Education: An Empirical Study. *Decision Sciences Journal of Innovative Education, 10*(4), 575-587.

King, P., Kvavik, R. B., & Voloudakis, J. (2002). Enterprise resource planning systems in higher education. *EDUCAUSE Center for Applied Research: Research Bulletin, 22*, 1-11.

Kirkup, G., & Kirkwood, A. (2005). Information and communications technologies (ICT) in higher education teaching—a tale of gradualism rather than revolution. *Learning, Media and Technology, 30*(2), 185-199.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, 67-93.

Kleinman, D. L., & Klein, H. (2002). The social construction of technology: Structural considerations. *Science, Technology & Human Values, 27*(1), 28-52.

Kline, R., & Pinch, T. (1996). Users as agents of technological change: The social construction of the automobile in the rural United States. *Technology and culture*, 763-795.

Krishnaveni, R., & Meenakumari, J. (2010). Usage of ICT for Information Administration in Higher education Institutions–A study. *International Journal of Environmental Science and Development, 1*(3), 282-286.

Kroeze, J. H. (2011). Interpretivism in Information Systems: A Postmodern Epistemology?

Kumar, V., Gordon, B. R., & Srinivasan, K. (2011). Competitive strategy for open source software. *Marketing Science, 30*(6), 1066-1078.

Kyriakidou, A., & Venters, W. (2007). The multi-disciplinary development of collaborative grids: the social shaping of mammogrid.

Landrum, H., Prybutok, V., Zhang, X., & Peak, D. (2009). Measuring IS system service quality with SERVQUAL: Users' perceptions of relative importance of the five SERVPERF dimensions. *Informing Science: the International Journal of an Emerging Transdiscipline, 12*(1), 17-35.

Langat, B., & Kamuren, F. (2013). *Information Systems Outsourcing Versus Systems Development: A Comparative Study of the Preferred Choices of the Different Banks in Nakuru Town.* Paper presented at the Scientific Conference Proceedings.

Lassila, K. S., & Brancheau, J. C. (1999). Adoption and utilization of commercial software packages: Exploring utilization equilibria, transitions, triggers, and tracks. *Journal of management information systems*, 63-90.

Lee, A. S. (1989). A Scientific Methodology For MIS Case Studies. *MIS Quarterly, 13(1)*, 30-50.

Light, B. (2001). The maintenance implications of the customization of ERP software. *Journal of software maintenance and evolution: research and practice*, *13*(6), 415-429.

Light, B. (2005). Potential pitfalls in packaged software adoption. *Communications of the ACM*, *48*(5), 119-121.

Light, B., & Papazafeiropoulou, A. (2004). Reasona Behind ERP Package Adoption: A Diffusion of Innovations Perspective. *ECIS 2004 Proceedings*, 80.

Liu, L., Feng, Y., Hu, Q., & Huang, X. (2011). From transactional user to VIP: how organizational and cognitive factors affect ERP assimilation at individual level. *European Journal of Information Systems*, *20*(2), 186-200.

Lucas Jr, H. C., Walton, E. J., & Ginzberg, M. J. (1988). Implementing packaged software. *MIS quarterly*, 537-549.

Lupu, A.-R., Bologa, R., Sabau, G., & Muntean, M. (2008). Integrated information systems in higher education. *WSEAS Transactions on Computers*, *7*(5), 473-482.

Lyons, M. (2005). Future ICT Systems—Understanding the Business Drivers. *BT technology journal*, *23*(3), 11-23.

Lyytinen, K., & Robey, D. (1999). Learning failure in information systems development. *Information systems journal*, *9*(2), 85-101.

Magni, M., Angst, C. M., & Agarwal, R. (2012). Everybody Needs Somebody: The Influence of Team Network Structure on Information Technology Use. *Journal of management information systems*, *29*(3), 9-42.

Markus, M. L., & Pfeffer, J. (1983). Power and the design and implementation of accounting and control systems. *Accounting, Organizations and Society*, *8*(2), 205-218.

McGrath, K. M. (2003). *Organisational culture and information systems implementation: A critical perspective.* London School of Economics and Political Science.

Melville, N. P. (2010). Information systems innovation for environmental sustainability. *MIS quarterly*, *34*(1), 1-21.

Mengesha, N. T. (2010). The Role of Technological Frames of Key Groups in Open Source Software Implementation in a Developing Country Context. *The Electronic Journal of Information Systems in Developing Countries*, *43*.

Mertens, D. M. (2010). Transformative mixed methods research. *Qualitative Inquiry*.

Meyer, U., & Schulz-Schaeffer, I. (2006). Three Forms of Interpretative Flexibility.

Myers, M. D., & Avison, D. (2002). An introduction to qualitative research in information systems. *Qualitative research in information systems*, *4*, 3-12.

Nasir, S. (2005). The development, change, and transformation of Management Information Systems (MIS): A content analysis of articles published in business and marketing journals. *International journal of information management, 25*(5), 442-457.

Negash, S., Anteneh, S., & Watson, R. T. (2012). A PhD in information systems for emerging economies: the Addis Ababa University model. *Information Technology for Development, 18*(3), 270-276.

Nelson, P., Richmond, W., & Seidmann, A. (1996). Two dimensions of software acquisition. *Communications of the ACM, 39*(7), 29-35.

Nelson, P., Seidmann, A., & Richmond, W. (1998). Software Acquisition: The Custom/Package and lnsource/Outsource Dimensions. *Advances in Computers, 47*, 341-367.

Neuman, W. L., & Robson, K. (2004). *Basics of social research*: Pearson.

Oliveira, L., Vasconcelos, N., Queiroz, F., Queiroz, J. V., & Hékis, H. R. (2011). Contribution of integrated management systems to university management: case study of the federal university of Rio Grande Do Norte. *Journal of Social Sciences, 7*(3), 415-422.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information systems research, 2*(1), 1-28.

Orlikowski, W. J., & Baroudi, J. J. (2001). Studying information technology in organizations: research approaches and assumptions. *Information Systems Research, 2(1)*, 1-28.

Orlikowski, W. J., & Gash, D. C. (1994). Technological frames: making sense of information technology in organizations. *ACM Transactions on Information Systems (TOIS), 12*(2), 174-207.

Panigyrakis, G. G., & Chatzipanagiotou, K. C. (2011). The impact of design characteristics and support services on the effectiveness of marketing information systems: An empirical investigation. *Review of Business Information Systems (RBIS), 10*(2), 91-104.

Paré, G. (2001). Using a positivist case study methodology to build and test theories in information systems: Illustrations from four exemplary studies.

Phahlane, M. M., & Kekwaletswe, R. M. Management Information Systems Use in Higher Education Environments.

Pinch, T. J., & Bijker, W. E. (1984). The social construction of facts and artefacts: or how the sociology of science and the sociology of technology might benefit each other. *Social studies of science*, 399-441.

Pollock, N., & Cornford, J. (2004). ERP systems and the university as a "unique" organisation. *Information Technology & People, 17*(1), 31-52.

Pollock, N., & Cornford, J. (2005). Implications of enterprise resource planning systems for universities: An analysis of benefits and risks.

Pollock, N., & Hyysalo, S. (2014). The Business of Being a User: The Role of the Reference Actor in Shaping Packaged Enterprise System Acquisition and Development. *MIS quarterly, 38*(2), 473-496.

Pollock, N., Williams, R., & Procter, R. (2003). Fitting standard software packages to non-standard organizations: the 'biography'of an enterprise-wide system. *Technology Analysis & Strategic Management, 15*(3), 317-332.

Pozzebon, M. (2001). Demystifying the rhetorical closure of ERP packages. *ICIS 2001 Proceedings*, 37.

Prell, C. (2009). Rethinking the Social Construction of Technology through'Following the Actors': A Reappraisal of Technological Frames. *Sociological Research Online, 14*(2), 4.

Rangarajan, K., & Tiwari, S. K. (2014). Packaged Software Application Services Industry: Evolution and the factors affecting the inductry. *IOSR Journal of Business and Management, 16(4)*, 33-38.

Ritchie, J., Lewis, J., Nicholls, C. M., & Ormston, R. (2013). *Qualitative research practice: A guide for social science students and researchers*: Sage.

Saadé, R. (2003). Web-based educational information system for enhanced learning, EISEL: Student assessment. *Journal of Information Technology Education: Research, 2*(1), 267-277.

Sagitova, G. K. (2012). Management information system for higher educational institutions during Kazakhstan transition to knowledge economy. *World Applied Science Journal*, 55-60.

Sahay, S., & Robey, D. (1996). Organizational context, social interpretation, and the implementation and consequences of geographic information systems. *Accounting, Management and Information Technologies*, *6*(4), 255-282.

Saiedian, H., & Dale, R. (2000). Requirements engineering: making the connection between the software developer and customer. *Information and software technology, 42*(6), 419-428.

Salami, P., & Ahmadi, H. (2010). Review of Farm Management Information Systems (FMIS). *New York Science Journal, 3*(5), 87-95.

Salimi, F., & Dankbaar, B. (2008). Strategic Approaches to ERP Implementation. *Journal of International technology and Information Management, 17*(1), 1.

Sarker, S. (2000). Toward a methodology for managing information systems implementation: A social constructivist perspective. *Informing Science, 3*(4), 195-206.

Sawyer, S. (2000). Packaged software: implications of the differences from custom approaches to software development. *European Journal of Information Systems, 9*(1), 47-58.

Schneidewind, N. F. (1987). The state of software maintenance. *Software Engineering, IEEE Transactions on*(3), 303-310.

Seeman, E. D., & O'Hara, M. (2006). Customer relationship management in higher education: Using information systems to improve the student-school relationship. *Campus-Wide Information Systems, 23*(1), 24-34.

Stamelos, I., Angelis, L., Morisio, M., Sakellaris, E., & Bleris, G. L. (2003). Estimating the development cost of custom software. *Information & Management, 40*(8), 729-741.

Steiniger, S., & Hay, G. J. (2009). Free and open source geographic information tools for landscape ecology. *Ecological Informatics, 4*(4), 183-195.

Supriana, S. I., & Permatasari, S. D. (2007). *New Integration Model of Information System on Higher Education Institution*.

Temizkan, O., Kumar, R. L., Park, S., & Subramaniam, C. (2012). Patch release behaviors of software vendors in response to vulnerabilities: an empirical analysis. *Journal of management information systems, 28*(4), 305-338.

Ternai, K. (2003). Integrated Systems in the Role of Integration of Education. *Informatics in Education-An International Journal*(Vol 2_2), 279-290.

Thong, J. Y. (1999). An integrated model of information systems adoption in small businesses. *Journal of management information systems, 15*(4), 187-214.

Thoreau, F. (2010). *The 'interpretive flexibility' of nanotechnologies in context: The case of a leading R&D center in Flanders, Belgium.* University of Maastricht, Maastricht, Netherlands.

Vlas, R. E., & Robinson, W. N. (2012). Two rule-based natural language strategies for requirements discovery and classification in open source software development projects. *Journal of management information systems, 28*(4), 11-38.

Von Alan, R. H., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly, 28*(1), 75-105.

Von Hippel, E. (2001). Learning from open-source software. *MIT Sloan management review, 42*(4), 82-86.

Von Krogh, G., & Von Hippel, E. (2003). Special issue on open source software development. *Research policy, 32*(7), 1149-1157.

Wagner, E. L., Scott, S. V., & Galliers, R. D. (2006). The creation of 'best practice'software: Myth, reality and ethics. *Information and Organization, 16*(3), 251-275.

Walsham, G. (1995a). The emergence of interpretivism in IS research. *Information systems research, 6*(4), 376-394.

Walsham, G. (1995b). The Emergence of Interpretivism in IS Research. *Information Systems Research, 6(4),* 376-394.

Walsham, G. (1995c). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems, 4*(2), 74-81.

Wang, E. T., Barron, T., & Seidmann, A. (1997). Contracting structures for custom software development: The impacts of informational rents and uncertainty on internal development and outsourcing. *Management science, 43*(12), 1726-1744.

Weber, R. (2004). The Rhetoric of Positivism Versus Interpretivism. *MIS quarterly, 28*(1).

Welke, R. J. (1994). The shifting software development paradigm. *ACM SIGMIS Database, 25*(4), 9-16.

Wijaya, A. F., & Manongga, D. (2012). Information Systems Strategic Planning to Increase Competitive Advantage of Higher Education Using BE VISSTA Planning Methodology (Case Study: SWCU SALATIGA). *The International Journal of Organizational Innovation*, 68.

Williams, R., & Edge, D. (1996). The social shaping of technology. *Research policy, 25*(6), 865-899.

Winner, L. (1993). Upon opening the black box and finding it empty: Social constructivism and the philosophy of technology. *Science, Technology, and Human Values*, 362-378.

Wynn Jr, D., & Williams, C. K. (2012). Principles for conducting critical realist case study research in information systems. *MIS quarterly, 36*(3), 787-810.

Yeo, K. T. (2002). Critical failure factors in information system projects. *International Journal of Project Management, 20*(3), 241-246.

Yin, R. K. (2003). *Case Study Research Design and Methods*. Thousand Oaks: Sage Publications.

Zornada, L., & Velkavrh, T. B. (2005). *Implementing ERP systems in higher education institutions.* Paper presented at the Information Technology Interfaces, 2005. 27th International Conference on.