A Survey of Software Testing Processes Used by Software Developers in Kenya

By

Koimur, Haron Chepchieng

A Management Research Project Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Business Administration, School of Business, University of Nairobi.

October 2006

DECLARATION

This research project is my original work and has not been presented for a degree program in any other university.

Signed...Date 09-11-2006

Koimur Haron Chepchieng

This research project has been submitted for examination with my approval as the university supervisor.

Signed: lelei.....................................Date 13/11/2006

Joel K. Lelei
Lecturer,
Department of Management Science
School of Business, University of Nairobi

DEDICATION

This research project is dedicated to my wife Sophie J. Koimur and my children Faith Jesire, Davies Monir and Emmans Bowen

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUT:	Application Under Test
CSQA:	Certified Software Quality Analyst
CSQE:	Software Quality Engineer Certification
CSTE:	Certified Software Tester
CSTP:	Certified Software Test Professional
CTM:	Certified Test Manager
DOD:	Department of Defence
ICT:	Information and Communications Technology
IEEE:	Institute of Electronic and Electrical Engineers
GUI:	Graphical User Interface
IS:	Information Systems
ISO:	International Standards Organisation
LAN:	Local Area Network
MIS:	Management Information Systems
NIST:	National Institute of Standards and Technology
PC:	Personal computers
SPSS:	Statistical Analysis Software Package
SAMATE:	Software Assurance Metrics and Tool Evaluation
SDLC:	Systems Development Life Cycle
SOA:	Service Oriented Architecture
TPS:	Transaction Processing System
WAN:	Wide Area Network

ACKNOWLEDGEMENTS

First of all, I thank the almighty God for opening the opportunity for me to do the MBA program and it has been by his grace that I was able to finish the course and this project.

I am also deeply indebted to my supervisor, Mr. Joel K. Lelei for his invaluable guidance throughout this research project. He took keen interest in the work, contributed key ideas and actually remained steadfast and focused throughout the research project.

I am highly indebted to my wife Sophie, daughter Faith, sons Davies and Emmans for their moral support and encouragement and the fact that they understood why I was many times away from home when I should have been with them during the evenings and weekends.

A special word of thanks goes to my brothers and sisters for their encouragement and more so my brother Stephen who understood whenever I missed crucial family functions to be able to attend classes.

I cannot forget to thank my co-workers in the service of God in my church for their prayers and also being ready to take up my assigned responsibilities whenever there were classes or examinations that were occasionally fixed for Sundays.

I also extend my appreciation to all the respondents who took time to give their views through the questionnaires, without which this research would not have been complete.

Thanks a lot and may God bless you all.

ABSTRACT

As more and more organizations turn towards computer based information systems for decision making, it becomes necessary to ensure that such systems are error free. To be able to come up with an error free system, there are a number of things to be done to eliminate any errors that may exist.

There is need to enhance detection and prevention of software errors and bugs in order to avoid the catastrophic effects that arise if the system is not well tested. The focus of this study was on the software testing processes by software developers in Kenya. While doing this study, some elaborate background is provided at the beginning regarding the growing use of computers. The problem that is being addressed by the study is the effects that an error can cause to businesses which can be catastrophic. The major problem noted is the effects that untested software may cause if not well tested. Tools that are used in software testing process are also explained.

A key aspect of the study was to attempt to establish the tools and processes used in software testing by Information and Communication Technology developers in Kenya. The results show that, ICT Consultants perform software testing to a large extent whenever they are developing software. In addition, the findings revealed that, respondents view the system testing as important in the system development process. Attempts were also made to find out if there are any existing challenges facing software testing process in Kenya.

The report provides some descriptions of the methodology and the data analysis techniques which were used such as tables, pie charts and some frequency tables. The findings and recommendations arising out of the research are included in the explanations in each of the sections.

Findings of the study show that about two thirds of the software developers in Kenya do actual testing of the software. Most of the firms interviewed carry out their software testing processes internally within the organization. However, those customizing or performing agency roles of re-selling already developed software make use of experts from one of the partners of the company that developed and owns the software. The tools used for software testing processes are imported and thus a firm has to plan properly to avoid shortage of the same.

Notable among the challenges is the lack of skills on software testing. Also, the process of software testing is not put as part of the project activities when implementing a new system.

There were no serious limitations to the study but the limited time and budget restricted the researcher from attaining a 100% response rate. Some of the software testing tools and processes are relatively new and may not be well understood locally. It is possible that some of the respondents could have given their response with this limitation. To understand the driving force behind adoption of certain software testing tools requires a more in-depth and specific study.

This study however, ends with recommendations as regards the importance of the software testing and that there is need to have software testing processes entrenched in the ICT Policy for all organizations. It should even be made as a formal activity to be carried out at the time of project completion.

CHAPTER 1: INTRODUCTION

1.1 Background

Advancements in the Information and Communications Technology (ICT) globally and the over-reliance on computer generated information by all organizations imply that failure of computer systems can lead to disastrous results. To alleviate this problem, well tested systems are needed. The computer experts are now turning much attention to it. The continued use of computer generated reports for decision making therefore render organizations dependent on computers. Information is a critical resource for organizations, as fundamental as energy or machines (Burch and Grudnitski, 1986).

Due to the increased over-reliance on computer systems, there is a great degree of exposure in the event that anything happens to the computing systems. It is therefore inevitable that systems failures can be very disastrous to the organization. Schultheis and Sumner (1995) stated that, the traditional organization is being transformed into the information-based organization, which uses Information and Communication Technology to produce significant changes in the work pattern. They further stated that ICT will in future become an integral part of business. This view has become a reality and therefore it is because of unpredictability of system availability and the fact that there is dire need to minimize on the problems arising from systems failure that it has become absolutely necessary to do thorough tests of any application that is being installed.

Thus all software developers ensure that they go through testing as part of the process in the software development. It is hard to create a perfect program the first time it is done; even major software developers may have bugs and errors on their operating systems. The importance of software testing can never be underestimated given its impact on business operations thus there is always need to work out the bugs prior to release of any software for use. Exhaustive and thorough testing must be conducted to ascertain whether the system produces the right results (Laudon and Laudon, 2001).

There are a number of explanations of the term software testing. Some related terms that are relevant to software testing such as test data among others are also explained or defined by individual authors as follows:

Schulmeyer and James, (1998) defined the term software quality in multiple ways but concludes with this definition: "Software quality is the fitness for use of the software product." This definition implies the evaluation of software quality related to the specification and application of software quality. The Institute of Electrical Engineers' (IEEE) *Standard Glossary of Software Engineering Terminology* defines quality as "the degree to which a system, component, or process meets specified requirements, and customer or user needs or expectations." Software quality assurance is the confirmation of the expectations against the output that has been achieved.

On the other hand, Myers (1995) defined the term "testing" as a process of executing a program to find faults. He further continued to elaborate the fact that testing is actually a destructive approach that is "The tester is an enemy of the program". The term is also used to refer to the phase in the system development life cycle in which the system inputs, processes, and outputs are tested to make sure they work correctly.

Jorgensen, (2002) describes test data as the actual values used in the test or that are necessary to execute the test. Test data instantiates the condition being tested (as input or as pre-existing data) and is used to verify that a specific requirement has been successfully implemented (comparing actual results to the expected results).

Beizer (1995) defined Performance testing by explaining that it is a process that can be undertaken to: show that the system meets specified performance objectives, tune the system, determine the factors in hardware or software that limit the system's performance, and to project the system's future load-handling capacity in order to schedule its replacements".

Asbock (2004) defined the term stress test as a test that is designed to determine how heavy a load the Web application can handle. A huge load is generated as quickly as possible in order to stress the application to its limit. The time between transactions is minimized in order to intensify the load on the application, and the time the users would need for interacting with their Web browsers is ignored. A stress test helps determine, for example, the maximum number of requests a Web application can handle in a specific period of time, and at what point the application will overload and break down.

Perry (1995) defined the term bandwidth testing as a process of Testing a site with a variety of link speeds, both fast (internally connected LAN) and slow (externally, through a proxy or firewall, and over a modem); sometimes called slow link testing if the organization typically tests with a faster link internally.

As a result of the realization that companies could face a lot of problems in case of systems failures, it has become inevitable that testing of systems is a critical requirement in the process of systems development. Several organizations have set standards for software quality. The standards that are being set appear to clearly follow the software product life cycle processes or stages. Some of the organizations that have attempted to set standards include; the Institute of Electronic and Electrical Engineers (IEEE), the United States of America's Department of Defence (DOD) and the popular International Standards Organization (ISO) (Schulmeyer, 1992).

It is important to state here that ISO developed a number of standards that are absolutely necessary for an effective systems testing that will lead to guaranteeing on the outcome of a given system. According to Pressman (2000), one important item that the ISO indicated is the fact that the inspection and testing process adds substantial value to the management of the quality of a system being set up. For an organization to be registered to ISO 9001, it must establish policies and procedures guiding to address each of the ISO requirements and then demonstrate the actual application of these policies and procedures (Wachira, 2003).

Davis and Olson (1985) set out the aspect of system testing by stating that a policy may establish different testing and documentation guidelines depending on the importance of the problem, the duration of the system, and the task inter-dependencies involved. Davis and Olson (1985) then set out four levels of testing that may be used under such situations as:

1. No testing or non-developer review;
2. Minimal testing and peer review;
3. Medium testing and external review and finally;
4. High level of testing and formal external review.

Software testing starts long before the software is released to end users or market. Usually, project cycles are tight, customers are demanding, and because competition is fierce, one's standards should be high, because one believes in what one is doing. But the risk of failure is always present, and failure has a way of blindsiding those who aren't adequately prepared for it. Software is usually regarded as the company's catalyst for innovation, the backbone of its customer service, and the fundamental source of its competitive advantage (Schultheis and Sumner, 1995).

Davis and Olson (1985) further states that when a firm has produced software that operates according to the customer's expectations, that handles peak loads readily, and that responds expeditiously to requests for information, one will have achieved some critical business goals.

The benchmark for this scenario continuing is that one should have:

1. Improved the reliability of one's systems for both customers and department staff.
2. Minimized system downtime.
3. Reduced exposure to risk in the marketplace.

When a firm is working at the leading edge, good software testing is a sure way to improve the firm's bottom line. Testing as a process encompasses all aspects of systems development cycle. Although all stages of systems life cycle requires testing to be done, software development stage has overtime been considered as the most critical process that requires thorough testing to be done. Software testing will not only reduce the effects of systems failures but also gives assurance to the system's users of the integrity of the system they are using (Myers, 1995).

Because it costs more money to fix problems once the software has been released to the public, it is imperative to have thorough software testing before deployment since the costs could be greater than planned. Software Testing is the best way to find problems with a program, a good testing service can save thousand in the long run.

Myers (1995) explains that the purpose of testing cuts across the entire systems development life cycle. It entails ensuring that all the processes of setting up systems have been adhered to properly. Consequently, if this is the case, then one may raise the question of why test if all processes are followed? The fundamental thing that every developer has to underscore is the fact that the final quality of the software one is designing has to undergo the process of testing. This will give the developer and the software user some level of confidence about the system.

According to Yeates and Daniels (1992) the results of testing will prove that the system is working correctly and will thus:

1. Give added confidence to the systems designer and his team;
2. Inspire the confidence of the users in the new system;
3. Prevent holdups and frustrations during implementation and reduce the requirement for maintenance.

1.2 The Growth of Information and Communication Technology in Kenya

The period of the mid 1980's to date has heralded unprecedented increase in the usage of information and communication technology. Various commercial enterprises, large and small alike have institutionalized the concept of information technology. The media, learning institutions, banks, airlines, Telecommunications sector, health service providers and even the Kenyan Government have all seen the need to prioritize aspects that relates to information and telecommunication technology. This fast growth and dependence on ICT has inspired the Government to set up a fully pledged department that handles the information technology issues due to its crucial role in economic growth.

Laudon and Laudon (2001) emphasized the importance of ICT by stating that "one reason why systems play a larger role in organizations, and why they affect more people, is the growing power and declining cost of information technology."

There are a lot that points to the Government's efforts in creating an enabling environment for the growth in the ICT sector. In 2003, the government committed itself to review the legal framework to remove constraints that have discouraged adoption and use of e-commerce. It further went ahead to develop a master plan for e-government, meaning that the ICT will play a critical role in the performance of all the sectors in the economy. Further, the Minister of Finance in his budget for financial year 2006/2007 that was read in June 2006 came out explicitly in support of this Government initiative by actually waiving all the Value Added Tax on ICT equipment and accessories. The prices of Personal Computers (PCs) have actually reflected this in the recent past. With all the above efforts, the Kenyan economy is definitely poised for an all time high growth rate in ICT development.

Prahalad and Krishnan (1999) argued that firm's software applications are rapidly tending towards operating like the humans central nervous system. Based on the trend stated above as regards the direction of Kenya's policy towards ICT, this argument can not be ignored. Consequently, it is necessary to make every effort in guarding against the catastrophic effects of systems failures.

1.3 Statement of the Problem

Computerized business information systems put firms at better competitive position in any industry within the economy. Consequently, the need to embrace ICT fast may drive firms into implementing various systems. But if the systems are not well designed, it may lead to disaster in the business. To determine if the system is well designed, a process of testing has to be properly administered. It is considered that the part that is critical in the testing is the software itself, thus will require thorough testing (Laudon and Laudon, 2001).

The process of software testing will only be considered to have been thorough if the necessary tools are utilized coupled with sound processes and techniques. Software testing process is expected to be a challenging task and would need well trained persons who can manage the process and utilize the tools appropriately. There are a number of issues arising out of these which include whether appropriate software testing tools are available in Kenya. There is also the issue of whether the available tools are actually utilized by ICT developers in Kenya. It is clear that with technological growth there are a lot of software testing tools that have been developed and needs to be utilized to ensure accuracy of systems that are being implemented. Testing then answers the question "Will the system produce the desired results under known conditions? (Laudon and Laudon, 2001).

A general view of the Kenya's economic growth indicates that there is heavy usage of computer applications and every firm would like to be at the forefront to edge off competition. However, the aspect of software testing process and the associated tools seems to be hidden in SDLC and not attract much needed attention. It is due to this reason that there is a critical need to establish the extent of systems testing. While testing is being done, there are bound to be a number of challenges in the process.

According to Yeates and Daniels (1992), it is important to test a new system thoroughly before it is implemented, to prove that it will perform the tasks it has been designed to achieve. Therefore, the questions that this study seeks to address are:

- What procedures and tools are used to ensure accuracy in systems testing?
- Are there any challenges being encountered in the process?
- Are the established procedures and tools being applied by software developers in Kenya today?

Previous studies that have been done on software quality include one done by Wachira (2003). He dwelt on software quality assurance issues and did not focus attention on the tools and processes of software testing. He however stated that his research was not exhaustive of the issues touching on software hence it is due to this gap that there is need to do a research on software testing.

1.4 Objectives of the Study

The objectives of this study are as follows:

1. To establish the tools and processes used in software testing by Information and Communication Technology developers in Kenya.

2. To establish the challenges of software testing process in Kenya.

1.5 Importance of the Study

The findings of this study are of profound interest to a number of target groups that include the following:

Management of organizations: This group will find it invaluable for it makes them aware of the processes of testing systems and the requisite tools for doing the same. It enables them to be well informed in situations when they are selecting suitable systems testing methodologies and tools.

Systems developers and Vendors: The findings will broaden their understanding of the available tools for software testing and also their usage. This study will go along way in assisting the Software implementing companies to come up with appropriate methodologies and processes that should be used in testing software as and when they are implemented.

Government: The Government would be able to go through this write up and will find it invaluable for use in providing advisory service to the publics on the aspect of embracing software testing. On establishing the challenges of software testing, the government may attempt to alleviate some of the software testing problems such as the high cost of the testing tools.

Academics and Researchers: The findings in this study will provide an insight and basis of further research by the academics and the researchers. It will also form the basis for further research on Information systems software testing. It will point to them the proper process to be followed when doing systems testing thus enabling them to make informed research.

Computer Society of Kenya: This group will specifically find this study an invaluable source of information that is critical in their studies. As a result of this, many people will get informed through the society's newsletters and thus making the community informed on aspects of systems testing. Since this document will be available in the library facilities, it will be of great help to the society and all other interested parties.

1.6 Chapter Summary

This chapter served as an introductory note to this study. There are a few definitions provided on relevant areas that are being covered in this study. It also highlighted the significance of ICT in a growing economy with specific attention to the prevalent ICT growth in Kenya. The chapter also put to light the researcher's concerns on the fact that there is seemingly minimal attention on the issue of testing in the process of system implementations. It is important to note that the study is basically focusing on the Kenyan situation.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this part, attention is given to the Kenyan situation on how the processes and tools of software testing have been applied in the Software Development Life Cycle (SDLC). It is clear that nothing is introduced into the market for use without proper testing. Even beverages have to be tested physically by using the tongue signifying the importance that is attached to testing. The process of testing done to ensure that the product which is being produced is of good quality. Otieno (1999) demonstrated the extent to which ICT has infiltrated into the Kenyan market. He explains that most companies have adopted ICT in their core business processes. It is in this regard that much effort has to focus on the process and tools of testing the software to assure the developers and users of the systems capability.

2.2 Software

According to O'Brien (2002), the term "software" refers to computer programs and procedures concerned with the operation of an information system. O'Brien (2002), defined hardware as machines and media, physical equipment as opposed to computer programs or methods of use; mechanical, magnetic, electrical, electronic, or optical devices. Software can be defined as computer instructions or programming commands of data. It is generally anything that can be stored electronically. The storage and display devices for software are referred to as hardware.

Schulmeyer (1992) defines the term "software" as comprising of computer programs, procedures, rules and associated data and documentation pertaining to the operation of a computer system. According to Schultheis and Sumner (1995), software consists of the instructions that the hardware uses to process information. Schultheis and Sumner (1995) further explain hardware as the computer devices that support data processing, communications processing, and other computer related activities. According to O'Brien (2002), software is often divided into Systems software and Application software. O'Brien (2002) explains the two categories as follows:

Systems software includes the operating system and all the utilities that enable the computer to function. It refers to the operating system and all utility programs that manage computer resources at a low level. Systems software consists of low-level programs that interact with the computer at a very basic level. This includes Operating system, compilers, loaders, linkers, debuggers and utilities for managing computer resources.

Application software includes programs that do real work for users. For example, word processors, spread sheet, and database management system fall under the category of applications software. Applications software is also called end-user programs and includes database programs, word processors, and spreadsheets. Figuratively speaking, applications software sits on top of systems software because it is unable to run without the operating system and system utilities. It generally comprises programs designed for an end user, such as word processor, database systems, and spreadsheet programs. Sumner and Schultheis (1995) state that application software is programs that perform specific data or text processing function.

2.3 Software Testing Process

O'Brien and Laudon and Laudon (2001) describe the process of software testing as one that seeks to protect business entities from undesirable system down time or failures. It noted the cost implications of this to the business operations.

Burch and Grudnitski (1986) state that testing newly developed or modified system is one of the most important activities in the systems development methodology. It is an implementation activity that, similar to training personnel, requires careful planning and application. The goal of testing is to verify the logical and physical operation of all building blocks to determine that they operate as intended.

Software testing is the art of establishing existence of bugs in a given program. It is meant to establish if the software program is free from any error. This clearly can not be overlooked for it may be detrimental to the organizational business orientation. Davis and

Olson (1985) explains that testing is a difficult and time consuming process and users are likely to ignore the range of testing that is desirable because of their confidence in the system and in simple cursory testing.

According to Laudon and Laudon (2001), it is essential that all aspects of testing be carefully thought out and as comprehensive as possible. To ensure this, the software development team works with users to devise a systematic test plan. The test plan includes all the preparations for the series of tests which cover the following three activities:

Unit Testing, or Program Testing

This consists of testing separately each program in the system. While it is widely believed that the purpose of such testing is to guarantee that programs are error free, it is realistically impossible. Testing should be viewed instead as a means of locating errors in programs, focusing on finding all the ways to make a program fail. Once pinpointed, problems can be corrected (Laudon and Laudon, 2001).

System Testing

This tests the functioning of the system as a whole. It tries to determine if discrete modules will function together as planned and whether discrepancies exist between the way the system actually works and the way it was conceived. Among the areas examined are performance time; capacity for file storage and handling peak loads; recovery and restart capabilities; and manual procedures (Laudon and Laudon, 2001).

Acceptance Testing

This provides the final clarification that the system is ready for conversion. Systems tests are evaluated by users and reviewed by management. When all parties are satisfied that the new system meets their standards, the system is formally accepted for installation.

Laudon and Laudon (2001) stated that time allocated to testing process has traditionally been underrated in systems project planning. It actually requires upto 50% of the entire software development budget. Testing is considered to be time consuming since test the data must be carefully prepared; results reviewed and corrections made in the system and in many instances, parts of the system has to be redesigned. The risks of overlooking these steps are clearly enormous.

Software testing process forms an integral part of the systems development life cycle (SDLC). However, it is important to note that professional approach to the processes being undertaken is considered as a critical success factor. This certainly points to the need for proper and acceptable approaches.

O'Brien (2002) described systems testing as a process that involves testing hardware devices, testing and debugging computer programs, and testing information processing procedures. O'Brien (2002) further stated that programs are tested using test data that attempt to simulate all conditions that may arise during processing.

According to Myers (1995), the testing principles or methodologies are organized into ten categories which can be stated below:

1. Requirements-Phase

This phase of testing puts high considerations on the testing effort. It is important in the requirements phase for all stakeholders, including a representative of the testing team, to be involved in and informed of all requirements and changes.

2. Test-Planning Activities

This include ways to gain understanding of the goals of the testing effort, approaches to determining the test strategy, and considerations related to data, environments, and the software itself. Planning must take place as early as possible in the software life cycle, as lead times must be considered for implementing the test program successfully (Myers, 1995).

3. Testing Team

The make up of the testing team is usually considered as a core aspect of any successful testing program. A successful testing team has a mixture of technical and business domain knowledge, as well as a structured and concise division of roles and responsibilities. Continually evaluating the effectiveness of each test-team member throughout the testing process is important to ensuring success. The use of test team composed of user department managers, internal auditors, and various systems personnel, should be independent from the designers and programmers who developed the system (Burch and Grudnitski, 1986).

4. Architectural set up

Under this, considerations of the architectural for the system under test are given emphasis. This part is often overlooked by the professional testers. However, these factors must be taken into account to ensure that the system itself is testable, and to enable gray-box testing and effective defect diagnosis.

According to Burch and Grudnitski (1986), a computer must be able to process the variety of jobs that make up the total system. Some of the tools available to do this kind of tests are job accounting systems, hardware and software monitors and various performance utilities.

5. Test Procedures

It is important that testing teams ensures that there is an effective design and development of test procedures, including considerations for the creation and documentation of tests, and discusses the most effective testing techniques.

Procedures include all the things users do to interact with the system. Users range from order entry clerks to the chief executive officer. The reason for procedure testing is to see if the objective has been met (Burch and Grudnitski 1986).

6. Developer Unit Testing

For any testing process to be considered effective, the role of developer unit testing in the overall testing strategy has to be handled carefully. Unit testing in the implementation phase can result in significant gains in software quality. Testing each unit separately within the program is essential (Laudon and Laudon, 2001)

7. Tools for Testing

There are a number of automated testing tools issues that are used in the process of ensuring that a system is properly designed. The issue here is consideration of need to have the proper types of tools to use on a project, the build-versus-buy decision, and factors to consider in selecting the right tool for the organization (Myers, 1995).

8. Automated Testing

Automated software testing is the process of creating test scripts that can then be run automatically, repetitively, and through much iteration. Done properly, automated software testing can help to minimize the variability of results, speed up the testing process, increase test coverage (the number of different things tested), and ultimately provide greater confidence in the quality of the software being tested.

O'Brien (2002) indicated that there are selected best practices for automated testing such as the proper use of capture/playback tools, test harnesses, and regression testing.

According to Laudon and Laudon (2001) there are, some things for which automated software testing is not appropriate. These include:

(a) End user usability testing which is not typically a good candidate for automated testing.

(b) Tests which will not be run more than a couple of times are typically not a good candidate for automated tasting, since the payoff of in test automation comes after many test executions.

(c) Tests for areas of the application which experience a lot of change are also not a good candidate for automation since this can lead to substantial maintenance of test automation scripts. Such areas of the application may be more effectively tested manually.

It is important to note that test automation is software, and just like the software being built for internal or external customers, it must be well-architected. Good test automation architecture, such as a keyword-driven testing framework, will reduce the overall cost of ownership of test automation by minimizing maintenance expense and increasing the number of automated tests. This allows the user to run more tests (and achieve higher quality) for the same investment of time and money.

9. Testing Non-functional Aspects of a Software Application

This refers to the need to test the non- functional aspects of a software application. This entails ensuring that non-functional requirements are met, including performance, security, usability, compatibility, and concurrency testing, adds to the overall quality of the application (Myers, 1995).

10. Managing Testing Process

For any software tests to be successful, a strategy for managing the execution of tests, including appropriate methods of tracking test-procedure execution and the defect life cycle, and gathering metrics to assess the testing process needs to be employed and followed to the later (Yeates and Daniels, 1992).

2.4 System Development Life Cycle (SDLC)

Schultheis and Sumner (1995) define the SDLC as the steps followed in designing an information system, also referred to as the system development methodology. A systems development methodology establishes a set of procedures that conform to a life cycle. Without a methodology specifying what events and activities should occur in what order, systems development projects are likely to be out of range and time.

Schultheis and Sumner (1995) summarized the activities that are performed in the systems development life cycle as follows:

(a) Problem Definition

Problem definition refers to the examination and evaluation of the problems of the current system that is in use.

(b) Feasibility Study

This stage deals with the development of objectives and a logical model of the proposed system. Preliminary analysis of alternative design options, including the technical and economic feasibility of each alternative. The development of the recommendations for the system project including a projected schedule and proposed costs is done at this stage (Schultheis and Sumner, 1995).

(c) Systems Analysis

The systems analysis phase deals with detailed design of the current system, including its procedures, information flows, and methods of work organization and control. Development of a logical model of the current system is also done (Schultheis and Sumner, 1995).

(d) Systems Design

During this stage, the following are done:

1. Development of objectives for the proposed system
2. Development of a logical model of the proposed system, including process logic definition, logical data dictionary, and logical database design.
3. Evaluation of alternative design options
4. Development of a cost-benefit analysis to evaluate the economic implications of each alternative.

(e) Detailed Design

The detailed design stage encompasses the following processes:

1. Development of specifications for the physical system; including record design, file design, input design, and forms design.
2. Design of program specifications

3. Development of an implementation and test schedule.

(f) Implementation

The implementation covers the following processes:

1. Coding and documentation of programs
2. Evaluation and selection of hardware
3. Development of security, audit and control, and test procedures.
4. Development of training programs.

(g) Maintenance

Maintenance refers to the ongoing support, changes, and enhancements for the system. Also evaluation of user acceptance evaluation is done her.

2.5 System Software Testing phases

From the foregoing Systems development Life Cycle, it is clear that there is no fixed stage in a software development that testing is confined. Usually, the purpose of software testing is to find the most problems with the software as possible before it is released to the general public (Yeates and Daniels, 1992). Due to this some authors have summarized the stages into two broad phases namely alpha and beta.

(a) Alpha Testing

In software testing the alpha phase is when the software is tested in house, usually by the developers of the software. This is done before it is ever released to the public. During this phase of testing, it is expected that many bugs will be detected. Therefore, during the early stages of alpha testing, known as white box testing, the alpha testers often have access to the source code of the software. This allows them to make corrections as needed (Myers, 1995).

(b) Beta Testing

Beta testing occurs when a limited number of beta versions of the software are released to a population of the general public known as beta testers. These are often end users of the product who have volunteered to test the software. This phase of testing is very important and the developers of the software can never simulate all of the various situations that the software will be placed in (Myers, 1995).

Other stages within the Beta stage are as follows:

The Training Stage

This is one of the beta stages whereby at the point of training users, there are a number of steps that may have been missed out. Thus the users can provide some issues that needs to be included in the entire process (Myers, 1995).

The Preparation for go-live

This is also still beta stage of systems development, whereby all processes are subjected to testing to ensure that they comply with the user requirements and meets the desired solution. Here, issues to do with speed of transaction processing is given emphasis by doing a test run to ensure that users get the system when it is capable of running the required number of transactions at the same time. It is therefore evident that the software testing process can not easily be done by anyone. It is usually done by a specialized systems tester who has extensive experience in software development and if possible ought to have been certified as systems tester (Myers, 1995).

2.6 The Growing Importance of Software Testing

Burch and Grudnitski (1986) explained that testing, as a major development activity, is increasing in importance for a number of reasons which are enumerated as follows:

1. The increased dependency on computer generated information, by all levels of users within the organization in their decision making and problem-solving activities relates the organization's performance directly to the system's performance.

2. Increased usage and familiarity with computer-based systems has resulted in higher expectations by users of the system.
3. The inflationary trend in the cost of other development activities can be halted with improved test procedures.
4. The investment in systems maintenance resources can be reduced with improved testing procedures before the system is installed.
5. The trend toward a higher degree of integration of systems within an organization requires each new system implemented to perform successfully initially, not only for its own purposes, but so as not to degrade other existing systems.

According to Yeates and Daniels (1992), whether it is off the shelf system or in-house designed application, all systems are prone to some degree of failure. Such failures will definitely result in heavy losses to the organization that includes the following:

(a) Loss of time since processes will have to be done manually.

(b) Customers relationships will be adversely affected sending a wrong signal to the prospective clients

(c) There will be some costs that are incidental as a result of system failure such as consultancy costs, and even the cost of delaying customer invoices.

Additionally, Myers (1995) further indicated that, in some cases, testing can also be done by the end users of the given application that is being implemented since what is critical is the acceptance and ownership of the system. Software testing being a core process has even led to establishment of specialized institutes whose core training program is basically software testing. Such institutes include the International Institute of Software testing (IIST). As a result of establishment of a number of institutes, there are a number of software testers' certifications which have been created.

These include the following:

(a) The Certified Software Tester (CSTE)

(b) Certified Software Quality Analyst (CSQA)

(c) Certified Software Test Professional (CSTP)

(d) Certified Test Manager (CTM)

(e) Software Quality Engineer Certification (CSQE)

There are many other certifications that relates to software testing. It is notable that one can not underestimate the importance of software testing in any installation of computer applications.

2.7 Types of Software Testing

According to Myers (1995) the existing practice has overtime categorized the testing principles into the following areas: Requirements/development testing; Initial Planning test; The testing team; The system architecture; Test design and documentation; Unit testing; Automated testing tools; Automated testing: Selected best practices; Nonfunctional testing; and Managing test execution.

These commonly applied principles or types of testing can be explained as follows:

(a) **Development Testing**: This involves a process of unit testing which if not done results to defects that arise at some future date and may be very difficult to detect or catch.

(b) **Initial Testing:** Under this category, there are three sub tests that are performed. This can be enumerated as follows: Configuration testing; Compatibility / Conversion tests; Instability testing

(c) **On-going Testing:** Just like the initial testing, this process also has three areas to be tested namely: Functionality testing; Facilities testing; and Security testing

(d) **Performance Testing:** This has four areas that require critical testing to be done which includes: Volume testing; Stress testing; Load testing; Performance testing: If this test is not done then there will be possibility of slow response time which could have been avoided by doing the test.

(e) **Post Functional Testing:** The following three areas of tests relate to post functional aspects: Usability testing; Reliability testing; Recovery testing: This tests the system to establish its possibility of recovering as a result of disaster.

(f) **Post Installation Testing:** This test is done through the following:

Serviceability Testing: This test seeks to ensure that the system will be able to be serviced even after it has been rolled out. If it is not done, it leads to inability to make fixes and changes on the system.

Completion Testing: This test is done through the documentation testing process: This test seeks to ensure that the system will be fully documented to enable new users to understand or make changes to the application. If this is not done, it creates difficulties for new users.

In addition to principles stated by Myers (1995), Pirozzi (2000) presents software testing as consisting of several other sub categories of testing, each of which is done for different purposes, and often using different techniques. Pirozzi (2000) further states that software testing categories include:

Functionality Testing: Used to verify the proper functionality of the software, including validation of system and business requirements, validation of formulas and calculations, as well as testing of user interface functionality.

Forced Error Testing: This is a process where the developers attempts to break and fix the software during testing so that customers do not break it in production. Myers (1995) repeats the same aspect by stating categorically that “testers are enemies of the software”. This is due to their destructive processes at time of testing software.

Compatibility Testing: This type of test is done to ensure that software is compatible with various hardware platforms, operating systems, other software packages, and even previous releases of the same software.

Performance Testing: The performance test is used to see how well software performs in terms of the speed of computations and responsiveness to the end-user.

Scalability Testing: The scalability is applied to ensure that the software will function well as the number of users and size of databases increase.

Stress Testing: Used to see how the system performs under extreme conditions, such as a very large number of simultaneous users.

Usability Testing: This is applied to ensure that the software is easy and intuitive to use.

Application Security Testing: Security tests are done to make sure that valuable and sensitive data cannot be accessed inappropriately or compromised under concerted attack.

In some cases, there may even have to be other types of testing such as regulatory-compliance testing, depending on the type of software and intended industry.

2.8 Software Testing Tools

According to Myers (1995), there are a number of tools that are used for software testing. They are however categorized based on the specific test that is being performed. These categories are enumerated as follows:

(a) Test Design Tools

Tools that help one or the developer decide what tests need to be executed. Test data and test case generators are commonly used as tools under design.

(b) GUI Test Drivers

Myers states that the GUI tools automate execution of tests for products with graphical user interfaces. Client/server test automation tools, including load testers are applied here as well.

(c) Load and Performance Tools

Load and performance tools are tools that specialize in putting a heavy load on systems (especially client-server systems). These tools are often also referred to as GUI test drivers.

(d) Test Management Tools

This refers to the testing tools that automate execution of tests for products without graphical user interfaces. Also tools that help one work with large test suites.

(e) Test Implementation Tools

Miscellaneous tools that help developers to implement tests. For example, tools that automatically generate sub-routines are used here. Similarly, tools that attempt to make failures more obvious are applied to establish the success of implementation.

(f) Test Evaluation Tools

Tools that help evaluate the quality of one's tests. Code coverage tools can be applied or used in doing the evaluations.

(g) Static Analysis Tools

These are tools that are used to analyze programs without running them. Metrics tools fall in this category. Even with the extreme software testing that a program goes through, it can never be considered "bug free." Since the software cannot have been placed into all possible situations, there is always the possibility that other bugs exist that hasn't been discovered yet.

It is however important to note that there are so many tools of software testing and are so diverse that their application can be done at the various stages or groups of system implementation.

According to Pohjolainen (2002), it is necessary to divide the areas where tests can be done into some global grouping and in each group there are various tools that can be used simultaneously; Design, GUI (Graphical User Interface), Load and Performance, Management, Implementation, Evaluation, Static Analysis and outside of inspection: Defect Tracking, Web Sites and Miscellaneous.

These may be summarized by the diagram below which clearly demonstrates areas where the software testing process has to be undertaken.

Figure 1: The Division of the Tools

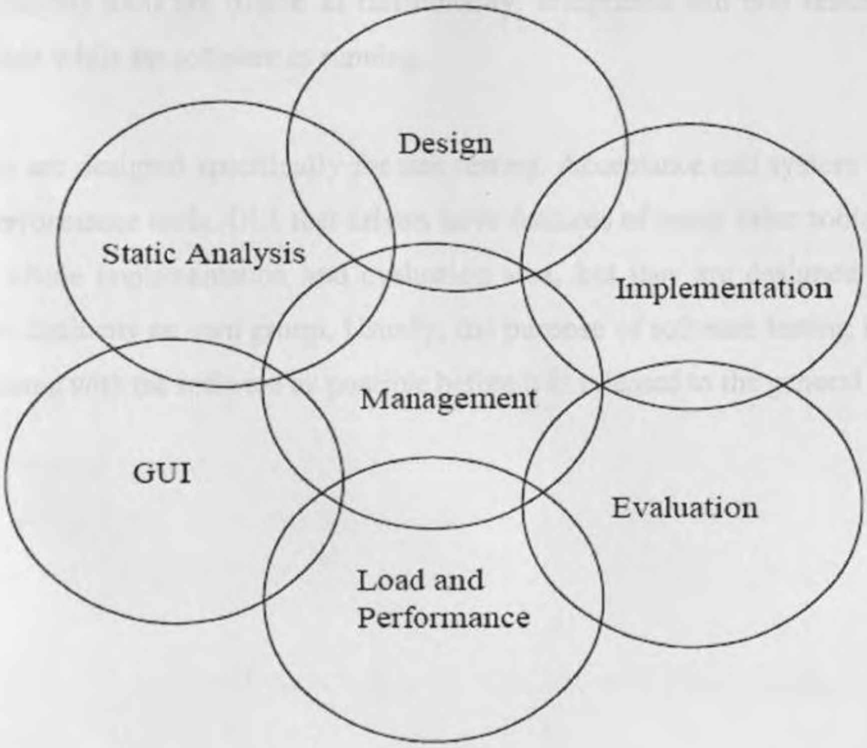


Figure 1. The Division of the tools

Source: *Write up by Pentti Pohjolainen, Department of Computer Science and Applied Mathematics, University of Kuopio March 2002*

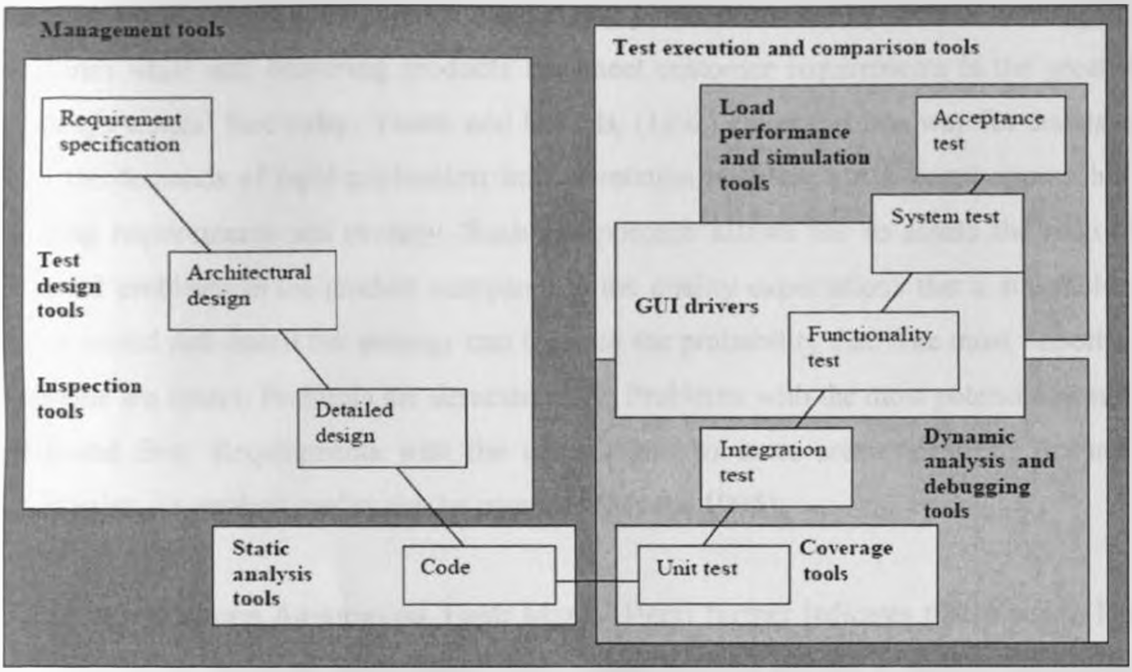
From Figure 1, it can be noted that the Management is at the centre of everything. This means that the management must be fully involved in every testing process to ensure availability of the testing tools and ownership of the tested system. The computer systems installations can either be procured off the shelf / pre-written or are designed within the organization, which is also commonly referred to as bespoke. However, tests will always be inevitable to ensure success of the system.

2.9 Tools Used in various stages of Software Development

According to Fewster and Graham (1999) and Tervonen (2000), test management tools can be used in the whole software development life cycle. Test design and inspection tools can be used in requirement specification, in architectural design and in the detailed design phases. The static analysis tools help testing in the coding phase. Execution and comparison tools can be used overall on the right side of the displayed V-model in Figure 2. Dynamic analysis tools are usable in functionality, integration and unit testing. They assess the system while the software is running.

Coverage tools are designed specifically for unit testing. Acceptance and system tests fall in load and performance tools. GUI test drivers have features of many other tools and are useful in the whole implementation and evaluation area, but they are designed for GUI testing and are distinctly an own group. Usually, the purpose of software testing is to find the most problems with the software as possible before it is released to the general public.

Figure 2: **Division of the tools in the software development life cycle (V-model)**



Source: Fewster, M., Graham, D.: Software Test Automation. ACM Press, New York, 1999.

2.10 Software Testing Methodologies

Burch and Grudnitski (1986) established a number of software testing methodologies which can be categorized into the following summarized areas of tests:

Prioritizing Security Testing*:* Security testing is more than just adding negative test cases to the automation process. It requires a specific process to get the most out of valuable testing resources. The attackers have an advantage. They only need to find one vulnerability yet one has to find them all to block, which is clearly an impossible feat.

Test Automation*:* O'Brien (2002) indicated that the traditional test automation techniques have been known to be maintenance intensive, fragile in nature, require technical capabilities of those working with test automation scripts, and typically allowed for only static data to be captured in the test scripts. Additionally, the scripts recorded in test automation programs were usually tightly coupled to that program and to the application under test (AUT).

Risk Based Testing Strategies: With the rapid pace of application development in the e-business world, testing has become a challenging proposition. Trying to meet even tighter deadlines while still delivering products that meet customer requirements is the greatest challenge testers' face today. Yeates and Daniels, (1992) states that one way for testers to meet the demands of rapid application implementation is to use a risk-based approach to defining requirements and strategy. Such an approach allows one to assess the risks of potential problems in the product compared to the quality expectations that a stakeholder has. A sound risk-based test strategy can increase the probability that: The most important problems are found; Problems are detected early; Problems with the most potential rework are found first; Requirements with the most impact to users are tested first; Accurate information on product quality can be provided (Myers, 1995).

Creating one's own Automation Tool: Myers (1995) further indicates that it seems like everyone wants or needs to automate their manual testing and for a variety of reasons. Companies will go out and spend a lot of finances to buy an off-the-shelf tool only to find the tool sitting on the shelf years later. So why spend the money on a tool that is not being used. This will allow one to plug in products that need to be tested. The automation framework contains detailed logging and records test results into a relational database. It is also completely data and action driven. Testers can change the setup, execution, clean up, and/or expected results verification without changing a line of source code.

Database Validation Testing: This addresses some real life situations whereby testing discovered major problems prior to production and deployment of major applications. Database validation testing will tackle the database testing challenges for both situations and how the projects succeed despite their challenges.

Software Assurance Metrics and Tool Evaluation (SAMATE): The project SAMATE is an idea developed at National Institute of Standards and Technology (NIST). The project develops standard tests for software assurance tools and techniques, particularly those for security. To develop tests the SAMATE project is developing a reference dataset of thousands of flawed programs. This can clearly help a number of organizations who can't manage to develop there own testing software (Myers, 1995).

A Code Analysis Tool: Code analysis tools play an important role in the production of secure and robust software. The reality of tool integration is that it takes more than just buying a tool to find success. Real software development teams have processes and habits that can be difficult to adapt to new tools. As such, there are a few important steps that an organization should take in order to successfully integrate a code analysis tool. Logically following, it can then be determined who; in an organization are the right people to be using the tools. Another crucial piece is finding the right place in the software life cycle to add the tool (Myers, 1995).

Technical Experience of a Software Tester: There are a number of questions raised here which include: Should all Testers be Programmers by another Name? Inevitably, should every tester be a programmer? Some test managers think so. Some Software Development Life Cycles (SDLCs) seem to even imply that, there is no need for testers. However, this is a critical measure that increases the assurance of system quality (Myers, 1995).

Managing the Testing Effort: Laudon and Laudon, (2001) state that many testing efforts succumb to management and project pressures and become chaotic in their focus and work quality. It is simply the nature of the end game phase of software development projects, where anything goes in pushing for the delivery of a product and it's usually quality that goes first. Beyond the product quality impacts, the team usually suffers too with low morale and little empowerment. Myers (1995) states that Scrum is one of the Agile Methodologies and it focuses on project management in agile and iterative development efforts. It can be successfully applied to testing efforts to renew their focus and drastically improve overall results.

Learning to Finish Testing Process: Most people have experienced more than one software project that ended badly. Either the requirements were misunderstood or implemented poorly. Or overall quality targets couldn't be met because there were simply too many defects. Many projects fail in their last stages during testing. Not because of the testing per se, but because of the massive discovery of defects and functional gaps that indicate the true viability of the project. Providing experienced guidance that will increase the success of delivering a project is necessary under such circumstances (Burch and Grudnitski, 1986).

Cost Effective Test Automation: Automated testing is typically the most expensive kind of testing an organization does. Automation script authors need to have strategies for coding their scripts that result in returns on the costs incurred in creating them.

Testing Web Based Services: Web Services is a popular choice today to solve complex problem of integrating various business applications. However, developing services and offering them as web services is easier said than done especially when the business process change quite often and has to be modified. As more and more applications move towards Service Oriented Architecture (SOA), testers need new tools and techniques to test these rather large face-less or GUI free interfaces (Burch and Grudnitski, 1986).

Test Automation in the Test Processes: Many companies are trying to modernize their testing methods; many have already done so. By integrating their testing team more firmly into the SDLC, they are starting to reap the advantages of early test planning and design, and getting more leverage from their testers in the organization. The integrating automated tests into the rest of the test processes that the team uses enhance ownership of the process. It also makes the test results available to all – including developers. Saving artifacts from the automated test process and investigating the quality of automated testing hence obtaining meaningful metrics from the automation (Burch and Grudnitski, 1986).

Evaluating Requirements for Testability: For a test engineer, perhaps the most important measure of requirements quality is testability. By improving testability during requirements development, one not only will make test design easier, but will have gone a long way toward building better software for less cost. It's much easier for developers to design and code from good requirements. It is also critical to identify the requirements problems that reduce or improve testability: ambiguity, incompleteness, inconsistency, incorrectness, and compoundness. According to Schulmeyer and James (1998), the developers concerns dwell mainly on the following areas for testing purposes:

1. **Software Reliability:** This is established by ensuring that the software is resistant to failure during execution and that there is no crashing, hanging, and memory leaks.

2. **Software Functionality:** This is established by ensuring that the software executes the required use cases (desired behavior) and also the fact that it does not execute non-required behavior

3. **Software Performance:** This emphasizes the fact that the system executes and responds in a timely manner in real world situation without intervention.

4. **Fault Tolerance:** The developer is concerned with the fact that the system behaves reasonably in abnormal situation and that it degrades gracefully while at the same time being able to report the degradation

2.11 Chapter Summary

This chapter has elaborated various issues on software testing. Some explanation has been provided as regards the process of software testing. Detailed explanation on the importance of system testing is also outlined, which then emphasizes the need for this study. Further, there is an explanation of the processes that are followed in the system development life cycle (SDLC) and a clear integration of the two is illustrated whereby the testing process is intertwined with the system development cycle. Also explained are the tools that are used in testing systems software and the recommended methodologies used in performing system software tests.

Clearly, achieving the right blend of software testing process is typically a mix of test types, executed through a combination of manual and automated testing. The mix and number of tests is determined by the quality requirements of the application. Each method (automated or manual) is used for what is really appropriate at any given time.

CHAPTER 3: RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

This section gives the methodology used in this research project. This includes the research strategy or approach, Population of the study, type of data collected, method of data collection and also data analysis procedures applied.

3.2 Research Design

This study sought to establish the tools and processes used in software testing by Information and Communications Technology developers in Kenya. It also sought to establish the challenges that are faced in the process of doing the software testing. The research methodology adopted was a survey.

A survey design was chosen based on the questions posed and given that no documented studies have been conducted in relation to software testing tools and processes in Kenya. The researcher also wanted to get a broad picture of the entire software testing processes that are used by ICT Consulting firms in Kenya. This design was also suitable for this kind of research because the study intended to collect data meant to ascertain facts about processes of software testing and tools being used.

3.3 The Population

The population of this study consisted ICT firms that provide systems development consultancy services to various firms. The list of the chosen firms was obtained from the yellow pages of the Kenya telephone directory 2005 and the Nation business directory 2005. In addition to listings in the directories, snowball method was used to identify firms that may not have been captured through the directories.

The respondents were ICT Managers, Project managers or their appointed representatives or assistants. The choice of this target group was based on the fact that they have the knowledge about the firms' practice and also provide authoritative and reliable data.

The choice of consulting firms or ICT Consultancy firms was because they are in the field of building or developing Information systems for various organizations. They also consult a lot in deployment of solutions to firms. Consultants are used by organizations because of the complexity and the rapid changes that occur in information technology. There is also the difficulty of firms recruiting and retaining ICT staff with the required skills. Consulting firms also keep their consultants on the cutting edge of technology and train them with new techniques to better compete for business. Thus the knowledge being sought through this study is in line with latest trends in technology.

3.4 Sampling Procedure or Technique

Due to the number of firms and the fact that they are all based in Nairobi, the study conducted a survey of all the consulting firms. A census was conducted while judgmental sampling was employed in focusing on the respondents. Ninety two questionnaires were administered to the respondents in the selected firms and Sixty (60) completed responses were received giving a response rate of 65%. Organizations within Nairobi were targeted in the survey due to time and cost constraints and also the fact that the headquarters for many consulting firms is Nairobi where policies and guidelines for the firms are formulated.

3.5 Data Collection Technique

Primary data was collected using a structured questionnaires, which was self administered through drop and pick method. The researcher pre-tested and discussed the questionnaire with two experienced systems developers and made relevant changes prior to release to the respondents. To facilitate wider coverage, the researcher worked with two research assistants who were trained for the exercise. The researcher and his assistants contacted the respondents on phone and agreed on when to drop the questionnaires. On drop off, the researcher and his assistants ensured that the questionnaire document was intact and explained to the respondents what was expected of them. They then agreed with respondents as to when the questionnaire was to be picked.

The questionnaire consisted predominantly closed ended questions with a few open ended questions. There were three sections in the questionnaire labeled as Section A, Section B, and Section C. Section A was used to collect data on he profile of the respondent and the organization. Section B was used to collect data on the tools, techniques and processes used in software testing. By making use of a five point Likert scale, respondents indicated the extent to which their firms make use of Software testing tools and the level of importance that the firms attach to certain factors considered necessary when selecting software testing tools.

Section C was used to collect data on the extent to which challenges stated in the questionnaire are faced by the firms.

3.6 Data Validation

The data collected were coded, collated and edited accuracy, uniformity of responses, consistency and completeness. Demographic variables were grouped to avoid having to deal with values with wide coverage. Seven of the target consultancy firms have since closed their firms and their contacts were still in the telephone directory. In addition, five returned questionnaires were found to be incomplete on some items which were of material value to the study. They were then rejected at this stage.

3.7 Data Analysis

The data analysis involved summarization of the data using exploratory data analysis approaches and emphasized the use of diagrams to explore and understand the data. Diagrammatic presentations were used to provide visual contents.

The data collected on Section A regarding the demographics captured about the respondents and the organization was analyzed by making use of Microsoft spreadsheet application, excel, to manipulate, and process data for graphical viewing. Bar charts were used to show the frequency of occurrence or distribution of variables. To further emphasize the proportion, pie charts were used. Further analysis of the demographic factors was done in order to provide different perspectives for interpretation of the findings.

Section B of the questionnaire collected data on the extent of use of various software testing tools and processes. These were presented using simple tables to show the frequency of occurrence in line with the first objective of the study. Further analysis was performed on the data with respect to various software testing tools and the results presented using percentage component bar charts for ease of interpretation. This same analysis was used to establish the extent of importance that firms attach to certain software testing activities.

Section C of the study was to establish the challenges faced in the software testing tools and processes by ICT Consultancy firms. In view of the number of variables involved, the findings in respect to this section were subjected to factor analysis to establish the most critical challenge. Factor analysis is a statistical technique for classifying a large number of interrelated variables to a limited number of factors. It is an ideal method for re-organizing the items under review that a researcher is investigating into conceptually more precise groups of variables. The analysis was performed using statistical analysis software package (SPSS) in order to ease the work of generation and formatting of the outputs.

3.8 Chapter Summary

This chapter gives highlights of the methodologies of data collection and techniques. It explains the nature of the population under study with some details as to how data can be obtained. The chapter also gives an indication of the data analysis methods that was applied. Some elaboration is given as regards the contents of the questionnaire.

CHAPTER 4: DATA ANALYSIS AND FINDINGS

4.1 Introduction

This chapter presents the results of the analysis and findings of the study. Of the total of 92 questionnaires distributed, 65 were received back representing 70% of what was distributed. Out of the 65, five (5) were rejected for being incomplete on material items. Consequently, the study was based on 60 filled questionnaires representing 65% of what was delivered. This was considered sufficient for the data analysis which is then presented in the various sections in this chapter.

The first section of data analysis corresponds to Section A of the questionnaire where tabulations of demographic information about the respondent and the firms are provided. The reason for analyzing this section is to provide an overview of the respondents' demographic factors. In addition, the information was used in ascertaining the validity of the results.

The second section deals with the analysis of the responses to questions involving extent of use of software testing tools and processes. It attempts to address the first objective which seeks to establish the software testing processes and tools used by the firms that are responding to the questionnaires.

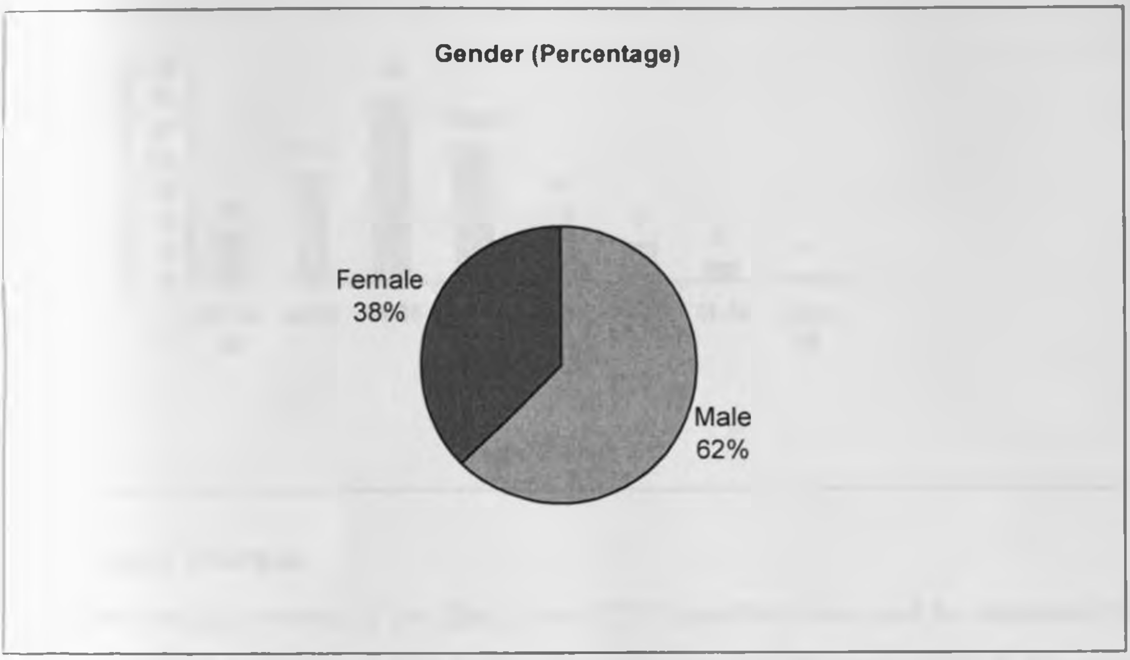
The last section deals with the analysis of the responses to questions on the extent to which the firms face challenges relating to software testing tools and processes.

4.2 Respondents Profile

4.2.1 Gender

Out of the total respondents, 62.5% of the respondents were males and 37.5% were females. This can be graphically represented in the pie chart in Figure 3. This outcome reflects similar scenario as the ones previously established in other local studies on ICT consultancy by Wachira (2003) and Muhanji (2005).

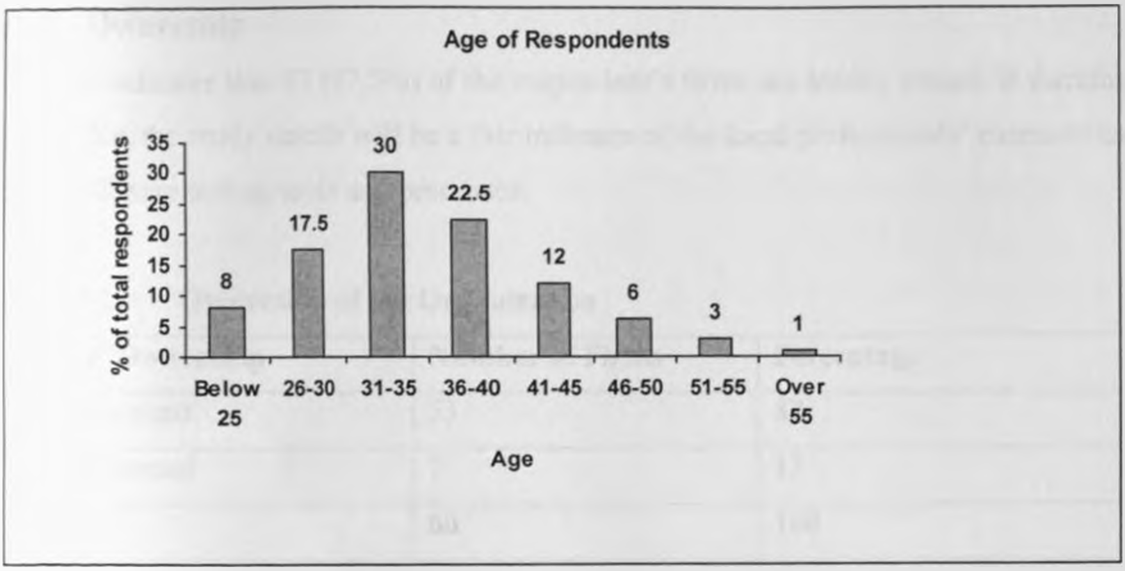
Figure 3: Gender Distribution



4.2.2 Age

The age of the respondents is represented by Figure 4. The lowest age of the respondents was 23 years and the highest was 44. However, as displayed in the Figure 4, the highest concentration of the respondents' age ranges between the age brackets of 26-30 with 17.5%, 31-35 with 30%, 36-40 with 22.5%. This age distribution shows that ages between 31 and 35 years comprise the highest number of the persons in the ICT consultancy.

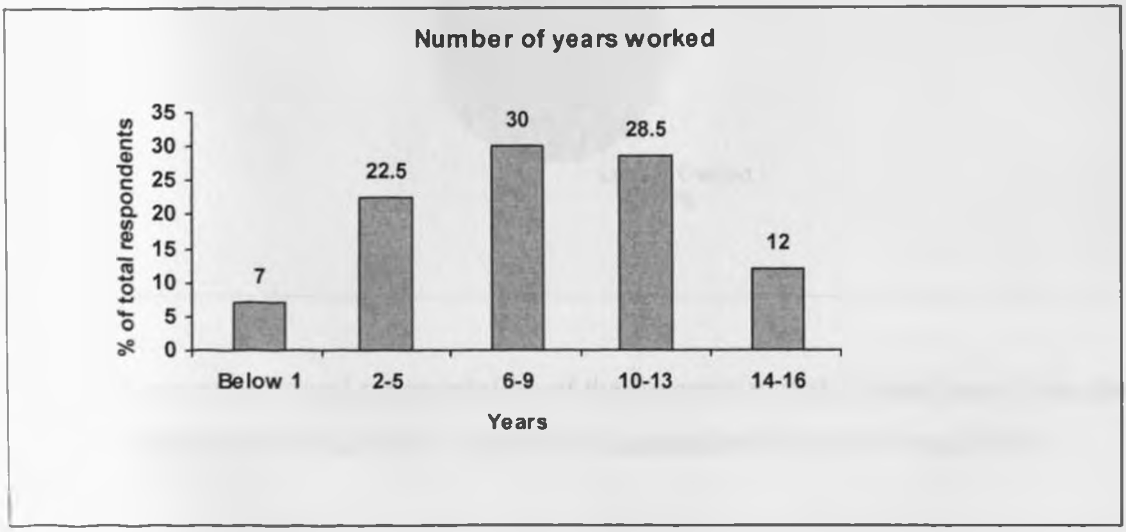
Figure 4: Age Distribution



4.2.3 Years Worked

The years one has worked in the firm as an ICT Consultant was used to determine the experience of the professional. The range of years worked by the respondents is between 1 and 16 years. Figure 5 show that, the distribution appeared to be more concentrated on higher number of years worked. It therefore provides an indicator that the respondents are quite experienced and thus are qualified to provide the necessary information being sought.

Figure 5: Years of Service



4.3 Organizations Profile

4.3.1 Ownership

Figure 6 indicates that 53 (87.5%) of the respondent's firms are locally owned. It therefore means that, the study results will be a fair indicator of the local professionals' extent of use of the software testing tools and processes.

Table 4.1: Ownership of the Organization

Type of Ownership	Number of Firms	Percentage
Locally owned	53	87
Foreign owned	7	13
Total	**60**	**100**

Figure 6: Ownership Distribution

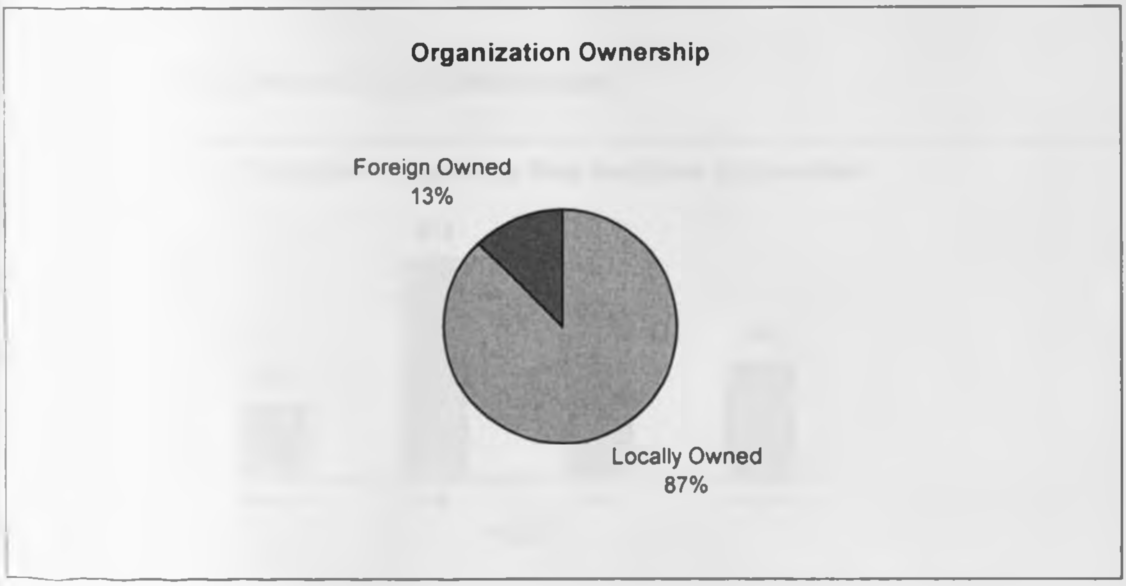


Figure 6 shows proportional representation of the ownership of the consultancy firms that have been interviewed in this study. It portrays a domination of locally owned firms.

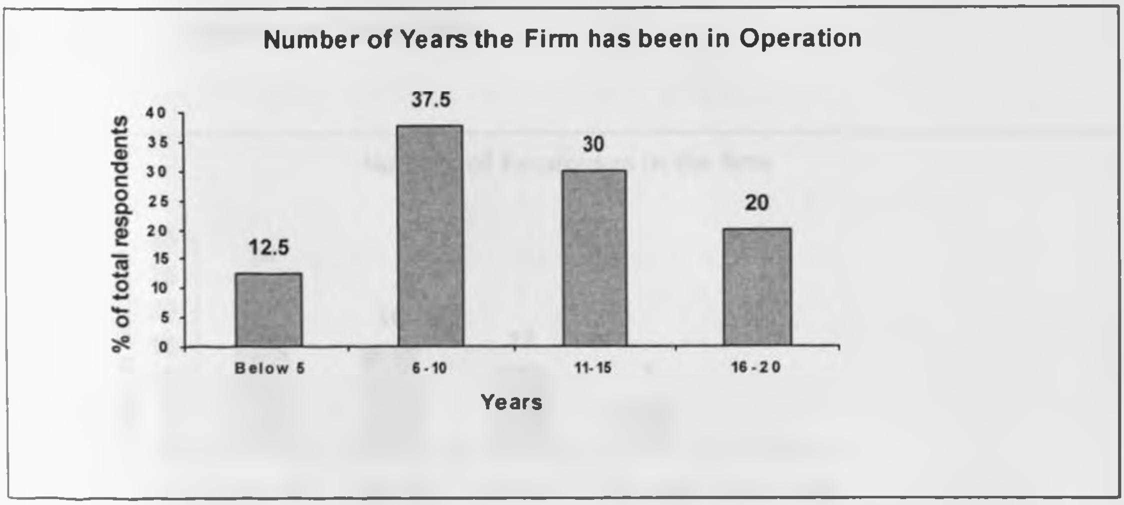
4.3.2 Years in Operation

Figure 7 shows the organization's age since inception. It shows that the duration ranged between 4 and 25 years. Most of the firms were however reflected at the range of 6 to 10 years. The organizations with less than 6 years were 5 while those over 15 years were 8 representing 12.5% and 20% respectively. The bulk of 67.5% represents companies that have been in operation for between 6 and 15 years. Due to the maturity of the firms, the responses to the questionnaires were presented with some reasonable experience, thus will be useful to this study.

Table 4.2: Years in Operation

Years in Operation	Number of Firms	Percentage
0 to 5	14	23
6 to 10	25	42
11 to 15	13	22
16 to 20 years	8	13
Total	**60**	**100**

Figure 7: Operational Years Distribution



4.3.3 Number of Employees

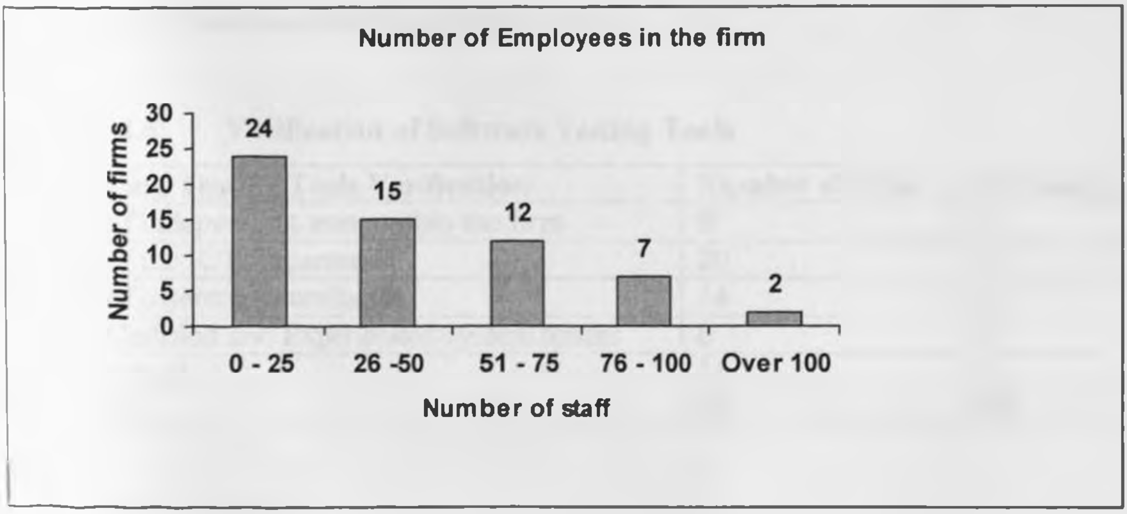
Table 4.3 shows that most of the respondent firms have few employees ranging between 1 and 25 staff. This represents 40% of the total respondent firms. This number of staff is manageable especially in consulting firms as established in previous studies research by Wachira (2003). Only 21 firms representing 35% of total respondents had more than 50 staff. This means that many consulting firms employ few staff.

Table 4.3: Employees

Number of Employees	Number of Firms	Percentage
01 to 25	24	40
25 to 50	15	25
50 to 75	12	20
75 to 100	7	12
Above 100	2	3
Total	**60**	**100**

Figure 8 shows that the higher the number of staff the fewer the firms. This observation confirms the fact that consultancies provide service with few staff.

Figure 8: Number of Employees



4.4 Tools, Techniques and Processes in Software Testing

4.4.1 Standards Applied in Software Testing

The respondents were required to indicate the standards that govern the firms' software testing function. From Table 4.4, responses received shows that 70% of the respondents indicated that they make use of internally defined rules and procedures. Only 10% use internationally recognized standards such as ISO 9001, while 4% do not apply any existing standards. On overall, these results give an indication that software developers rely on set guidelines.

Table 4.4: Software Testing Standards

Standards	Number of firms	Percentage
Internally defined rules and procedures	42	70
Locally set standards	9	10
International standards such as ISO 9001	6	10
None at all	3	5
Others		
Total	**60**	**100**

4.4.2 Software Testing Tools Verification

Table 4.5 shows that there is low level of usage of Certified and Experienced Software Testers as there were only 6 firms that made use of them. 33% made use of their ICT Department to perform the tests. Out of the total respondents, 11 (19%) of them do not perform verification of software testing tools.

Table 4.5: Verification of Software Testing Tools

Software Testing Tools Verification	Number of firms	Percentage
Use of Independent team within the firm	9	15
Use of the ICT Department	20	33
Use of external Consultants	14	23
Use Certified and Experienced system testers	6	10
None at all	11	19
TOTAL	**60**	**100**

4.5 Software Testing Tools and Processes

There are various software testing tools that are used to authenticate software. In line with the objective of this study, these tools and processes can be analyzed in the following parts.

4.5.1 Extent to Which Software Testing Tools are Used

The first objective of the study was to establish the tools and processes used in software testing. The respondents were asked to indicate the extent to which some software testing tools and approaches are used in their firms. A likert scale of 5 was used to capture the data as follows:

1 Not at all
2 Small Extent
3 Moderate Extent
4 Large Extent
5 Very Large Extent

For the purpose of this analysis, the levels were reduced to:

Not at all 1,2 Moderate 3 Large Extent 4,5

The use of software testing tools is important to the success of software testing process. Table 4.6 shows responses on the usage of software testing tools by the responding firms. A listing of the commonly used software testing tools was provided the respondents indicated the extent to which their firms make use of each one of them.

Table 4.6: Software Testing Tools (Percentages)

Types of Software Testing Tools	No extent at all (%)	Moderate extent (%)	Large extent (%)
Load and performance test tools (Tools that specialize in putting a heavy load on systems -especially client-server systems)	17	37	46
Graphical User Interface (GUI) Test drivers (Tools that automate execution of tests for products with GUI)	10	50	40

Types of Software Testing Tools	No extent at all (%)	Moderate extent (%)	Large extent (%)
Requirement tools (Requirement-based or requirement definition related tools)	5	55	40
Component Tools (Tools which have some relationships with component-programming)	6	39	55
Test Management Tools (Tools that automate execution of tests for products that do not have GUI)	5	28	57
Regression test tools (Tools that are used to test software after modification)	10	39	51
Object-oriented Tools (Tools used specifically with object oriented programs)	0	15	85
Test Evaluation Tools (Tools which help one evaluate the quality of tests)	20	30	60
Static Analysis Tools (Tools that analyze programs without involving running of the program)	10	38	52
Interactive debugging (Debugging programs by interacting manually with the running of the program)	0	56	44
Trace and Snapshot (Tools that monitor the path of a program run and each statement of the source program)	23	45	32
Test Generators (Tools that help one decide what tests need to be executed. The tests being executed are on test data and test case generators)	0	54	46

Table 4.6 indicates the extent of use of the software testing tools during the information system development process as rated by the respondents. The findings support the common view that advocates use of testing software at all stages of development.

Table 4.7 shows further analysis of these figures on software testing tools by calculating the percentages values in relation to the total population of 60 respondents and the reduced Likert scale. From Table 4.7, it can be deduced that, object oriented software testing tools are highly utilized by the software developers with a rating of 4.75 while the least critical software testing tool was established to be Trace and Snapshot (Tools that monitor the path of a program run and each statement of the source program) which scored the lowest rating of 3.48.

Table 4.7: **Software Testing Tools (Proportion)**

Types of Software Testing Tools	No Extent at all	Moderate Extent	Large Extent	Sum of Proportions
Load and performance test tools (Tools that specialize in putting a heavy load on systems -especially client-server systems)	0.28	1.23	2.30	**3.82**
Graphical User Interface (GUI) Test drivers (Tools that automate execution of tests for products with GUI)	0.17	1.67	2.00	**3.83**
Requirement tools (Requirement-based or requirement definition related tools)	0.08	1.83	2.00	**3.92**
Component Tools (Tools which have some relationships with component-programming)	0.10	1.30	2.75	**4.15**
Test Management Tools (Tools that automate execution of tests for products that do not have GUI)	0.08	0.93	2.85	**3.87**
Regression test tools (Tools that are used to test software after modification)	0.17	1.30	2.55	**4.02**
Object-oriented Tools (Tools used specifically with object oriented programs)	-	0.50	4.25	**4.75**
Test Evaluation Tools (Tools which help one evaluate the quality of tests)	0.33	1.00	3.00	**4.33**
Static Analysis Tools (Tools that analyze programs without involving running of the program)	0.17	1.27	2.60	**4.03**
Interactive debugging (Debugging programs by interacting manually with the running of the program)	-	1.87	2.20	**4.07**
Trace and Snapshot (Tools that monitor the path of a program run and each statement of the source program)	0.38	1.50	1.60	**3.48**
Test Generators (Tools that help one decide what tests need to be executed. The tests being executed are on test data and test case generators)	-	1.80	2.30	**4.10**

4.5.2 Level of Importance Attached to Software Testing Activities

There are numerous factors that are considered important in the process of selecting and applying the software testing tools. A set of these factors were given a rating by the respondents as regards the level of importance that their firm attaches to each of the identified factor. Table 4.7 shows that, some activities are given high rating by some firms while the same are rated low by various other firms. Consequently, an evaluation is done in order to identify such scenarios and grouping of similar trends and believes, thus establishing the most preferred option. The concern therefore is to come up with acceptable considerations that are perceived to be important by the firms responding. The tabulations of the evaluations seeks to resolve a set of variables relating to software testing activities that are considered as important by all responding firms.

The likert scale of 5 that has been used to measure the level of importance is represented as follows:

1 Not Important
2 Somewhat Important
3 Important
4 Very Important
5 Extremely Important

For the purpose of this analysis, the levels of importance were factored on the basis of the Likert scaling and the total number of firms that responded which totaled 60. Table 4.7 shows an analysis of the summary of the responses that have been factored against the number of firms and each item considered as important by the responding firms.

Table 4.8: Level of Importance Attached to each Factor

Factors Considered Important in software testing process	Proportional distribution of firms responses				
	Not Important	Somewhat important	Important	Very Important	Extremely Important
The accuracy levels that the firm requires in the process of testing	-	0.20	0.85	0.93	1.08
The ease of availability of the software testing tools in Kenya	0.02	0.30	0.75	0.80	0.25
The cost of acquisition of the software testing tools	0.05	0.40	0.55	0.67	0.33
The availability of trained staff to use the specific testing tools	0.05	0.30	0.75	0.80	0.08
The performance load of the software testing tool	0.02	0.20	0.55	1.00	0.58
The compatibility of the testing tool to the software that is being developed	0.05	0.40	0.55	0.67	0.33
Extent to which the software testing tool is used in Kenya	0.03	0.43	0.45	0.80	0.33
The function of software that is being developed	0.07	0.30	0.70	0.87	-

Further analysis of the Factors in Table 4.8 reveals that firms attach a lot of importance to the accuracy levels that the firm requires in the process of testing. On the other hand, the list considered factor is the one on the function of the software being developed.

4.5.3 Software Testing Activities

Table 4.9 shows the extent to which responding firms perform certain software testing activities. The responses to this part of the questionnaire show that all firms that responded perform at least one software testing function. This, therefore, is a good indicator of overall performance of firms in ensuring quality of product delivery. The likert scale of 5 was also used to capture data that finds out whether firms use certain processes in software testing Processes and tools. Table 4.9 shows that, establishment of software testing tools (Total rating of 3.5), evaluation of software testing implementation (total rating of 3.4) and enforcement of software functional testing requirements (total rating of 3.4) are considered to a large extent by the firms responding. Table 4.9 also shows that establishment of formal review and reporting procedure on tested software is least considered in the performance of software testing process for it has a total summation of 2.93.

Table 4.9: Performance of Software Testing Processes

Software Testing Process or activity	% Proportion of each factor				
	No Extent at all	Small Extent	Moderate Extent	Large Extent	Very large Extent
Development of software testing plan	0.08	0.43	0.70	1.07	1.00
Implementation of software testing plan	0.10	0.27	0.95	1.07	0.92
Evaluation of software testing implementation	0.12	0.27	0.75	0.93	1.33
Establishment of software testing requirements	0.12	0.27	0.70	1.27	1.00
Enforcement of software functional testing requirements	0.08	0.23	0.95	1.07	1.08
Establishment of software testing performance requirements	0.08	0.40	0.95	0.87	0.92
Enforcement of software testing performance requirements	0.15	0.23	0.80	1.20	0.83
Establishment of software testing tools	0.10	0.27	0.65	1.07	1.42

Software Testing Process or activity	% Proportion of each factor				
	No Extent at all	Small Extent	Moderate Extent	Large Extent	Very large Extent
Enforcement of usage of the software testing tools	0.10	0.33	0.85	1.00	1.17
Monitoring and evaluation of the usage of the software testing tools	0.08	0.30	0.80	1.20	1.00
Establishment of software testing requirements	0.10	0.27	0.95	1.07	0.92
Enforcement of software testing requirements	0.12	0.27	0.75	0.93	1.33
Establishment of software testing operation procedures	0.08	0.40	0.95	0.87	0.92
Establishment of formal review and reporting procedure on tested software	0.18	0.40	0.60	1.13	0.67
Enforcement of formal review and reporting procedure after testing software	0.08	0.23	0.95	1.07	1.08

4.6 Analysis of Challenges to Software Testing

The second objective of the study was to establish the challenges that firms face in software testing process and the tools being used. The respondents were asked to indicate the extent to which some identified challenges affect them in software testing process. A Likert scale of 5 was used to capture the data as follows:

1 Not at all

2 Small Extent

3 Moderate Extent

4 Large Extent

5 Very Large Extent

For the purpose of this analysis, the levels were reduced to:

Not at all 1,2 Moderate 3 Large Extent 4,5

From Table 4.10, each of the challenges was experienced in the systems software testing processes and testing tools. Basically, the predominant challenges were: Lack of software testing tools; Non inclusion of software testing process in the firm's policy; and Inadequate software testing knowledge and skills by consultants with a rating of 3.52, 3.37 and 3.35

respectively. Table 4.10 also shows that, the lack of management support in software testing process scores as the least of the challenges with a rating of 2.97

Table 4.10: Challenges to Software Testing Processes (Proportions)

Software Testing Challenges	No Extent at all	Small Extent	Moderate Extent	Large Extent	Very Large Extent	Sum of factor
Lack of management support in software testing process	0.12	0.43	1.00	1.00	0.42	**2.97**
Lack of software testing material resources such as finances, people and time	0.12	0.53	0.75	0.93	0.67	**3.00**
Lack of software testing tools	0.07	0.23	1.05	0.67	1.50	**3.52**
Use of weak software testing tools	0.12	0.43	0.95	1.07	0.42	**2.98**
Lack of provision for software testing in the structure of project activities	0.13	0.37	0.95	1.07	0.50	**3.02**
Lack of awareness on the need to have system testing	0.07	0.40	1.00	1.13	0.58	**3.18**
Inadequate software testing knowledge and skills by consultants	0.08	0.33	0.75	1.27	0.92	**3.35**
The non-existence of timely software testing support by the designers of the software testing tools	0.12	0.53	0.75	0.93	0.67	**3.00**
Lack of support by software testing tools designers at time of software testing problem resolution	0.15	0.40	0.70	1.27	0.58	**3.10**
Lack of software testing standard procedures and policy	0.10	0.57	0.65	1.07	0.67	**3.05**
Inadequate communication within the firm regarding policies on software testing	0.13	0.37	0.95	1.07	0.50	**3.02**
Non inclusion of software testing process in the firm's policy	0.08	0.37	0.65	1.27	1.00	**3.37**

4.7 Chapter Summary

This study shows that most of the newly established firms face more challenges in software testing process as compared to those that have been in operation for long where new are those of less than 5 years in operation.

The findings reveal that majority of the firms carry out continuous software testing as the system is being designed and developed. Thus, there is no specific period in software development life cycle that is singled out as the software testing phase. The firms that do continuous testing accounted for 70% (30) of the total respondents.

There is general lack of skills in the software testing processes and tools by most of the software developers. This is important and emphasis needs to be put on it in terms of training of the software developers on testing and emphasis and it is also crucial that it is emphasized when performing development of software.

CHAPTER 5: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This being the final chapter presents a summary of the findings, conclusions and recommendations. Suggestions for further research have also been provided. Within the conclusions, some of the software testing requirements are stated.

5.2 Summary of the Findings

5.2.1 Demographic Information

Information on demographic data was collected from the respondents, analyzed and is useful for crosschecking the findings of the study.

It was established that 77.5% (47) of the firms use some set standards when performing the software testing process. This indicates a relatively high level of performance of software testing process by the software developers in Kenya.

It is also established that there are very few professionals who are able to perform software testing adequately. Of the 60 respondents, there were only 3 whose role is closely linked to software testing. This represents a mere 5% of the respondents.

Majority of the software developers carry out the testing processes by themselves. Out of the total respondents, only 15 firms (25%) make use of external persons or firms to perform the software testing process. It became clear that very few firms utilize the services of specialist software testers. This may lead to compromise of the system quality since the common trend is that the same persons perform all processes.

5.2.2 Software Testing Tools and Approaches

The aim of the study on this part is to determine the tools and techniques used in software testing by firms in Kenya. The findings show that about half (30) of the respondents have set policies as regards software testing function.

It was also established that most firms in Kenya makes use of external firms to certify the adequacy of the tools that are used in software testing. In fact out of the 60 respondents, 45 of them indicated that they make use of external resources to establish the software testing tools. It also became apparent that the software developers are driven by the project timelines and would like to finish the project on time. Thus, would in most cases rush to completion and continuously makes use of object oriented tools and others applying the Load and performance tools. These approach of utilizing this types of tools is meant to eliminate time taken in the process of software testing.

5.2.3 Challenges to Software Testing

The second objective of the study focused on the challenges that are experienced in software testing processes. The leading obstacle that has been mentioned by many is the lack of software testing tools in addition to difficulty of getting skilled and experienced software testers. This is a clear indicator that most developed software in Kenya may not have been adequately tested. It may not be a big surprise since the country is just on its initial stages of fully embracing ICT in all its sectors.

5.3 Conclusions

The finding is that most software developers in Kenya do not give specific time for software testing in the software development project activities. Instead, software testing is made to be part of the software development and thus errors arising after software are released are addressed as and when they arise. This is a poor way and hence exposes the software user to major catastrophes that can be avoided by conducting proper testing processes at all stages of software development.

There are no specific set policies and guidelines that are used in software testing processes. Similarly, there are no specific software testing tools that developers are advised to use. Consequently, each developer applies his / her experience in performing the testing. This may impact negatively to the software development firms in case the people with experience leave.

5.4 Scope and Limitations

The scope of this study was covering software testing processes, tools and challenges encountered in performing the same. Limitation experienced is that of time which was short thus the researcher could not personally go round interviewing the entire respondents, but instead did it with the help of research assistants who delivered and picked some of the questionnaires from the respondents.

There was a significant limitation on the knowledge of the respondents as regarding software testing processes and tools since very few could tell the existing software testing tools. This is due to the fact that most of the software developers have dwelt quite a lot on testing system performance as they develop the software.

Another major limitation was the attitude of the software developers. Some blatantly say that they don't have time to make one earn a masters degree. Others cite their busy and high level of concentration on their software development tasks thus do not want anything to distract. Despite the introductory letter from the School of Business, it was still a demanding task convincing the respondents of the fact that the research was for academic purposes. Most explain that the researcher just want to tap knowledge from them and disappear.

The researcher also faced the challenge of getting adequate literature on software testing tools. The tools in available literature were not well explained.

5.5 Suggestions for Further Research

This study concerned software testing processes and tools. A more in-depth study on application of specific software testing tools to off-the shelf software and bespoke software could be done. This will then provide relevant information of the types of software testing tools on each category of software development.

In addition, research could be done on types of software testing tools in different sectors of the economy. Such a study could help in determining the relevant testing tools and processes to be applied in each sector such as banking, manufacturing, services, utilities and even Government.

REFERENCES

Abbot B. (2000), Software can lead to Financial Catastrophe, Infoworld Test Center, http://www.infoworld.com

Asbock, S. (2004), Load Testing: Functional Testing of Software Systems", J. Wiley and Sons, New York.

Baker, E.R. and Fisher, M.J.(1992), *Software Quality program organization*, in G.G.Schulmeyer and J.I. McManus (Eds.), Handbook of software quality assurance (2nd ed.) The Netherlands: International Thompson Computer press.

Beizer B. (1995): "Black- Box Testing: Techniques for Functional Testing of Software Systems", J. Wiley and Sons, New York.

Burch J. G. and Grudnitski, G. (1986), Information systems: Theory and Practice. (4th ed.),John Wiley and Sons, New York.

Davis G. B. (1985), Management information systems: Conceptual Foundations, Structure, and Development. (2nd ed.),McGraw-Hill, Inc., New Delhi.

Desai, M. S., Ogden J. and Richards T. C. (2002), Information Technology Litigation and Software Failure, The Journal Of Information, Law and Technology, Issue No.2

Fewster, M. and Graham, D. (1999), Software Test Automation. ACM Press, New York, 1999.

Flowers, S. (1996), Software Failure: Amazing stories and Cautionary Tales, Brookfield, Rothstein: Wiley, John and Sons Incorporated, New York.

Government of Kenya (2001), Report of the sector working group on Information Technology, Government Printer, Nairobi

Government of Kenya (2003), Economic recovery strategy for wealth and Employment Creation, Government Printers, Nairobi.

Government of Kenya (2006), A handbook of the Government of Kenya Budget for the year 2006/2007, Government Printers, Nairobi.

Harman, H. (1976), Modern Factor Analysis, (3rd ed.), Chicago: University of Chicago Press.

Hervey G. T. (2002), Assessing Liability for software Failure, SRM Magazine Vol.2 No. 1, Cigital Inc. http://www.srmmagazine.com

Hower R. (2003), Software Quality Assurance and Testing-Frequently Asked Questions, Software Quality Assurance Testing Resource Centre, http://www.softwaregatest.com

IEEE87 (1987), Software Engineering Standards, IEEE Press

Jorgensen, P. C. (2002), Software Testing:A Craftsman's Approach, (2nd ed.): CRC Press, New York

Kenya Telephone Directory and Yellow Pages (2005) Kenya Postel Directories limited, Nairobi

Khosropour, M. (1989), Systems investigation in micro-computer development, Journal of systems Management

Kinyanjui, J. (2001), A survey of work values and the use of information systems: A case of selected business firms in Kenya, unpublished MBA Research Project, University of Nairobi

Laudon, K. C. and Laudon, J. P. (2001), Essentials of Management information systems: (4th ed.), Prentice Hall, New Jersey.

Liebowitz, J and Beckman, T. (1998), Knowledge Organisations: What every Manager should know, St. Lucie Press, New Delhi.

Muhanji. E. M. (2005), TA survey if information system development techniques and tools used in practice by information systems consulting firms in Kenya, unpublished MBA Research Project, University of Nairobi

Myers, C (1995), Programme testing Methodologies. (5th ed.), Oldenbourg 1995

Nation Business Directory (2006) A Publication of the Nation Media Group Limited, Nairobi

O'Brien, J. A. (1993), Management information systems: A managerial End user perspective. (2nd ed.), Richard D. Irwin, inc., New Delhi.

O'Brien, J. A. (2002), Management information systems: Managing Information Technology in the e-business enterprise (5th ed.), Tata McGraw-Hill, Inc., New Delhi

O'Brien, J. A. (2000), Management information systems: Managing Information Technology in the Internet-worked Enterprise (4th ed.), Richard D. Irwin, inc / McGraw-Hill, Inc., New Delhi

Peng, W. W. and Wallace, D. R. (1993), Software error analysis, National Institute of Standards and technology, special publication 500-209, Gaithersburg: United States Department of commerce, Washington DC.

Perry W. (1995), Effective Methods for Software Testing; John Wiley and Sons, New York.

Pressman R. S. (2000), Software Engineering-A Practitioners Approach (4th ed.), China: McGraw-Hill, Inc., Hong kong

Reitzfield, M. (1989), "On what should management systems department concentrate", Journal of systems management.

Schulmeyer, G. G. (1992), Standardization of software quality assurance, In G. G. Schulmeyer & J. I. McManus (Eds.), Handbook of software Quality Assurance, (2nd ed.), The Netherlands: International Thompson Computer Press

Schulmeyer, G. G. and James I. M. (1998), A Handbook of Software Quality Assurance, (3rd ed.) Prentice Hall, New Jersey.

Schultheis, R. and Sumner, M. (1995), Management information systems: The Manager`s View. (3rd ed.), Richard D. Irwin, inc., New Delhi.

Subramanian.N. (2002), Introduction to computers: Fundamentals of Computer science Vol 1.(19th reprint), New Delhi: Tata McGraw-Hill, Inc.

Tervonen, I. (2000), Lecture notes in University of Oulu, 2000.

Thomas C. (1990), Management Development and IT: "Bridging the gulf" Management and Data systems, Vol.3

Wachira, E. K. (2003), The Practice of Software Quality Assurance By Computer consulting firms in Kenya, unpublished MBA Research Project, University of Nairobi

Yeates, D. and Daniels, A. (1992), Basic Systems Analysis. (3rd ed.), Pitman Publishing, London.

APPENDIX I: INTRODUCTION LETTER

Dear respondent,

TO WHOM IT MAY CONCERN

A Survey of Software Testing Processes by Software Developers in Kenya

I am a postgraduate student in the school of business, University of Nairobi, pursuing a Masters in Business Administration degree program. I am undertaking a research on Software Testing processes and Tools utilized by Information system consulting firms in Kenya. It is aimed at establishing the extent to which the software testing tools and processes are used and also the challenges that are associated with its usage.

You have been selected as one of the respondent. I therefore kindly request you to fill in the attached questionnaire. The information from the questionnaire is needed purely for academic purpose and will therefore be treated with utmost confidentiality. Your name and that of your firm will not appear anywhere in the final report. A copy of the final report can be made available to you upon request.

If you require any further information, do not hesitate to contact me by email hkoimur@yahoo.com or cell phone 0722 922 433.

Thank you in advance for your cooperation.

Yours faithfully,

....

Haron C. Koimur

APPENDIX II: QUESTIONNAIRE

Questionnaire
No:________

A SURVEY OF SOFTWARE TESTING PROCESSES BY SOFTWARE DEVELOPERS IN KENYA

*(Kindly respond to this questionnaire with reference to **the firm you are working with**)*

SECTION A: DEMOGRAPHIC INFORMATION

Respondent's profile

1. Kindly indicate your title or position in the firm ....

2. How many years have you worked in the firm's ICT Consultancy? ....

3. Kindly indicate by ticking appropriately your highest level of education
   (a) O-Level (Form IV) [ ]
   (b) A-Level (Form VI) [ ]
   (c) Bachelors Degree [ ]
   (d) Masters Degree [ ]
   (e) PhD Degree [ ]
   (f) Others, specify
   ....

4. Your Gender: Male [ ] Female [ ]

5. Indicate by way of ticking the range of your age:
   (a) Below 25 years [ ]
   (b) 26 to 30 years [ ]
   (c) 31 to 35 years [ ]
   (d) 36 to 40 years [ ]
   (e) 41 to 45 years [ ]
   (f) 46 to 50 years [ ]
   (g) 51 to 55 years [ ]
   (h) Above 55 years [ ]

6. Which of the following best describes your role in the firm that you work for?
   (a) Chief Executive Officer [ ]
   (b) Chief Information Officer [ ]
   (c) Projects Manager [ ]
   (d) Quality Assurance Consultant [ ]
   (e) Software developer /Programmer [ ]

(f) Systems Analyst [ ]
(g) Functional consultant [ ]
(h) Training Consultant [ ]
(i) Documentation consultant [ ]
(j) Others, specify ……………………………………………………………………

Company's Profile

7. How many years has the firm been in operation? ……………………………………

8. What is the number of branches that the firm operates? ……………………………

9. Specify the ownership of your firm by ticking as appropriate the relevant category as stated below.
   1. Locally owned [ ]
   2. Foreign owned [ ]

10. What is the approximate number of employees in the firm? ……………………………

11. What is the average annual turnover of the firm in Kenya shillings? ………………
……………………………………………………………………………………………

SECTION B: TOOLS, TECHNIQUES AND PROCESSES USED IN SOFTWARE TESTING

12. Is there a set policy or guidelines as regarding software testing function in the firm?
Yes: [ ] No: [ ]

13. Does the firm have a specific plan for software testing in the process of software implementation?
Yes: [ ] No: [ ]

14. From the following functions of a consultant, tick the ones that are applicable in your firm.
(a) Development of new software [ ]
(b) Writing the feasibility study [ ]
(c) Provision of consulting services [ ]
(d) Performing software quality control process [ ]
(e) Training of users on new software [ ]
(f) Supervising implementation of new software [ ]
(g) Marketing the software to clients [ ]
(h) Doing documentations of the software processes [ ]
(i) Others, specify …. …………………………………………………………
……………………………………………………………………………………

15. From the list of consultants' functions stated in 14 above, under what function does the firm place the software testing process?

....

16. Under which standards does the firm carry out the software testing function? (You may tick more than one where applicable)

a) Internally defined rules and procedures [ ]
b) Locally set standards [ ]
c) International standards such as ISO 9001 [ ]
d) None at all [ ]
Others,Specify...
....

17. Which of the following aspects are applicable in the firm's software testing processes (you may tick more than one where applicable)?

(a) Continuous testing process in all development phases [ ]
(b) Use of set guidelines [ ]
(c) Test at pre-determined intervals [ ]
(d) Ad hoc testing [ ]
(e) Others,specify...
....

18. How does the firm verify the adequacy of the software testing tool(s) that are used in the process of developing new software?

(a) Use of Independent team within the firm [ ]
(b) Use the ICT Department staff in evaluating the system [ ]
(c) Use of external consultants [ ]
(d) Use Certified and experienced system testers [ ]
(e) None at all [ ]
(f) Others, specify...
....

19. The following is a listing of some of the factors that are considered necessary when selecting software testing tools. Please state the level of importance that the firm attaches to each one of them.

		1	2	3	4	5
		Not important	Somewhat important	Important	Very important	Extremely important
a.	The accuracy levels that the firm requires in the process of testing					
b.	The ease of availability of the software testing tools in Kenya					

		1	2	3	4	5
		Not important	Somewhat important	Important	Very important	Extremely important
c.	The cost of acquisition of the software testing tools					
d.	The availability of trained staff to use the specific testing tools					
e.	The performance load of the software testing tool					
f.	The compatibility of the testing tool to the software that is being developed					
g.	Extent to which the software testing tool is used in Kenya					
h.	The function of software that is being developed					
i.	Others, specify					

20. The following is a listing of some of the software testing tools and approaches. Kindly indicate by ticking the extent to which they are used in software testing in the firm.

		1	2	3	4	5
		No Extent at all	**Small Extent**	**Moderate Extent**	**Large Extent**	**Very Large Extent**
a.	Load and performance test tools *(Tools that specialize in putting a heavy load on systems - especially client-server systems)*					
b.	Graphical User Interface (GUI) Test drivers *(Tools that automate execution of tests for products with GUI)*					
c.	Requirement tools *(Requirement-based or requirement definition related tools)*					
d.	Component Tools *(Tools which have some relationships with component-programming)*					
e.	Test Management Tools *(Tools that automate execution of tests for products that do not have GUI)*					
f.	Regression test tools *(Tools that are used to test software after modification)*					
g.	Object-oriented Tools *(Tools used specifically with object oriented programs)*					
h.	Test Evaluation Tools *(Tools which help one evaluate the quality of tests)*					

		1	2	3	4	5
		No Extent at all	**Small Extent**	**Moderate Extent**	**Large Extent**	**Very Large Extent**
i.	Static Analysis Tools *(Tools that analyze programs without involving running of the program)*					
j.	Interactive debugging *(Debugging programs by interacting manually with the running of the program)*					
k.	Trace and Snapshot *(Tools that monitor the path of a program run and each statement of the source program)*					
l.	Test Generators *(Tools that help one decide what tests need to be executed. The tests being executed are on test data and test case generators)*					
m.	Integration Tools *(Tools used with integration testing. Integration testing is testing of combined parts of an application to determine if they function together correctly)*					
n.	Others, specify					

21. The following is a listing of software testing activities. Please rate (by ticking appropriately) the extent to which the software testing activities are performed in the firm.

		1	2	3	4	5
		No Extent at all	**Small Extent**	**Moderate Extent**	**Large Extent**	**Very Large Extent**
a.	Development of software testing plan					
b.	Implementation of software testing plan					
c.	Evaluation of software testing implementation					
d.	Establishment of software testing requirements					
e.	Enforcement of software functional testing requirements					
f.	Establishment of software testing performance requirements					
g.	Enforcement of software testing performance requirements					
h.	Establishment of software testing tools					
i.	Enforcement of usage of the software testing tools					
j.	Monitoring and evaluation of the usage of the software testing tools					
k.	Establishment of software testing requirements					

		1	2	3	4	5
		No Extent at all	Small Extent	Moderate Extent	Large Extent	Very Large Extent
l	Enforcement of software testing requirements					
m.	Establishment of software testing operation procedures					
n.	Establishment of formal review and reporting procedure on tested software					
o.	Enforcement of formal review and reporting procedure after testing software					
p.	Others, specify					

SECTION C: CHALLENGES TO SOFTWARE TESTING

22. The following is a listing of challenges to software testing that firms can face. Kindly indicate the extent to which each of the challenge is faced by the firm in software testing process.

		1	2	3	4	5
		No Extent at all	Small Extent	Moderate Extent	Large Extent	Very large Extent
a.	Lack of management support in software testing process					
b.	Lack of software testing material resources such as finances, people and time					
c.	Lack of software testing tools					
d.	Use of weak software testing tools					
e.	Lack of provision for software testing in the structure of project activities					
f.	Lack of awareness on the need to have system testing					

		1	2	3	4	5
		No Extent at all	**Small Extent**	**Moderate Extent**	**Large Extent**	**Very large Extent**
g.	Inadequate software testing knowledge and skills by consultants					
h.	The non-existence of timely software testing support by the designers of the software testing tools					
i.	Lack of support by software testing tools designers at time of software testing problem resolution					
j.	Lack of software testing standard procedures and policy					
k.	Inadequate communication within the firm regarding policies on software testing					
l.	Non inclusion of software testing process in the firm's policy					
m.	The lack of experienced software testers					
n.	Incompatibility of the software testing tools with the existing hardware					
o.	Inadequate time for software testing process					
p.	Lack of the software testing process reference materials.					
q.	Others, specify					

THE END

******Thank you very much for taking your valuable time to fill this questionnaire******

APPENDIX III: LIST OF TARGET RESPONDENTS

LIST OF ICT CONSULTING FIRMS IN THE STUDY

1. Afrosoft Technologies
2. Alien Technologies
3. Alphatech Microsystems
4. Alphax Infosys Ltd
5. Aequitas Technologies Ltd
6. Aren software systems
7. Alt+Tab Limited
8. Ankem Computer Services
9. Archways Technologies Ltd
10. Asa Computerized Information
11. Assured Agencies
12. Automated software systems
13. Business Connections And Technologies
14. Capital Computer Systems
15. Capital Technologies Kenya Ltd
16. Computer Capacities And Innovations
17. Dac-Net Communications Ltd
18. Dakel Computer Services Ltd
19. Dataflex Computer Consultants
20. Desktop Micro Services Ltd
21. Digital Systems Solutions
22. East Africa Software
23. Emerging Technologies Consultants ltd
24. Enterprise software solutions Ltd
25. Excel Integrated Solutions Ltd
26. Executive Information Systems Consultants
27. Executive Support Consultants
28. Extreme Computer Engineering
29. Fabit Automated Systems Ltd
30. Financial Application software Ltd
31. Finesse Technologies Ltd
32. Fintech Limited
33. Fishnet Technologies Ltd
34. Forecast 2000 Plus Electronics
35. Government Information Technology Services
36. Galileo Kenya
37. Graft Silicon Ltd
38. Hospitality Systems Consultants
39. Houston Technologies Ltd
40. Ibis Systems Ltd
41. Infoline Consultants
42. Infotech Systems and Services
43. Information Professionals Africa
44. Insight Technologies
45. Interpay Limited
46. Institute O Computer Applications Ltd
47. Institute Of Advanced Technology
48. Ken Data Systems Ltd
49. Laserview Info Systems
50. Lynx Integrated Systems Ltd
51. Malcotel computers systems and college
52. Mann Oscar Inner Connections
53. Micro Expert Ltd
54. Micro Interface Ltd
55. Mighty Micro Computer Services
56. Millennium Data Comm. Ltd
57. Mindsmith Software solutions
58. Nededge Computer Point
59. Neptune software
60. Niche Network Management Systems
61. Novacom Consultants
62. Ojanga Asego Systems Ltd
63. Online computer systems
64. Openview Business Systems
65. Orange East Afric.com
66. Pantronic Computer Consultants
67. Papustech Agencies
68. Passnet Consultants Ltd
69. Precission Software Consultants
70. Professional Technologies ltd
71. Prosoft Consultants
72. Protech Management & Consultants
73. Sera Software (EA) Ltd
74. Skyweb Technologies Ltd
75. Softcom Business solutions
76. Software Applications Ltd
77. Software Associates Ltd
78. Softwise (Kenya) Ltd
79. Software 2000 and beyond Ltd
80. Solution For Information Systems Ltd
81. Soluziona Systems
82. Sportsmark Business Systems & Services
83. Stack Systems Ltd
84. Stelews Agencies
85. Sunrays Electro Engineering Services
86. Synbase East Africa Ltd
87. Technology Interactive Learning Ltd
88. Todays Computers Ltd
89. Todays Online Ltd
90. Topaz Applications Ltd
91. Uninet Systems & Software Ltd
92. Vega Software Ltd