ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF COMPUTING

TOWARDS DESIGNING A MODEL FOR EVALUATION OF BUSINESS CRITICAL SOFTWARE SYSTEMS: A TOOL FOR SOFTWARE PROCUREMENT

BY:

ASHENAFI DUGUMA IDO

ADVISOR:

TIBEBE BESHAH (PhD)

FEBRUARY, 2016
ADAMA, ETHIOPIA

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF COMPUTING

A THESIS SUBMITTED TO DEPARTMENT OF COMPUTING OF ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR DEGREE OF MASTER OF SCIENCE IN SOFTWARE ENGINEERING

BY:

ASHENAFI DUGUMA IDO

FEBRUARY, 2016
ADAMA, ETHIOPIA

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF COMPUTING

TOWARDS DESIGNING A MODEL FOR EVALUATION OF BUSINESS CRITICAL SOFTWARE SYSTEMS: A TOOL FOR SOFTWARE PROCUREMENT

BY:

ASHENAFI DUGUMA

Name and Signature of Members of the Examination Board

TIBEBE BESHAH (PhD)	
Advisor	Signature
External examiner's name	Signature
Internal examiner's name	Signature
Chairperson's name	Signature

DECLARATION

This thesis is my original work and i have agreed to make it freely available for the purpose of using as a reference and further permission is given to copy the thesis for the entire or some part of it. It is known that copying or use of my work or its part for making profit is not allowed without permission. In any kind of use of this thesis; recognition shall be given to me and to Adama Science and Technology University.

ASHENAFI DUGUMA IDO

FEBRUARY, 2016

DEDICATION

This thesis is dedicated to my family.

ASHENAFI DUGUMA IDO

FEBRUARY, 2016

ACKNOWLEDGMENTS

First and foremost, all praise and thanks go to God for giving me the strength and patience, and providing me the knowledge to accomplish this research study. There are number of people who advise me and give me a good assistance on contributing to complete this thesis and I would like to thank them. I would like to thank all my parents for always being there for me, supporting and encouraging my education and always striving to make my life better. Next my deepest gratitude goes to all friends, who were giving help whenever I needed. I thank my advisor Dr. Tibebe Beshah for giving great advices on the research. The backing of academic staff members of ASTU was essential for this research and without them it would not have been successful. I thank Ato Amanuel Zewgneh from department of computing. My special thanks to my good friends Tinsae Gizachew and Semere Gebretinsae from Department of Computing. I would like to thank that organization employee who helps me during data collection.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AHP	Analytical Hierarchical Process
CBR	Case based reasoning
CBSE	Component based software engineering
COTS	Commercial off-The Shelf
CRM	Customer Relationship Management
DOS	Disk operating system
ERCA	Ethiopian Revenues and Customs Authority
ERP	Enterprise Resource Planning
GIS	Geographical information system
GQM	Goal-Question-Metric
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineering
ISO	International Standardization for Organization
KBS	Knowledge based system
MCDA	Multiple criteria decision aid
MCDM	Multiple criteria decision making
NASA	National Aeronautics and Space Administration
NRC	National Research Council
OO	Object Oriented
PECA	Plan, Establish, Collect and Analyze
RBR	Rule based reasoning
SEI	Software Engineering Institute
STACE	Socio-technical approach for commercial off-the shelf software evaluation
UML	Unified Modeling Language
WSM	Weighted scoring method

ABSTRACT

The process of purchasing commercial software system needs systematic guidelines; so that the appropriate software system can be selected in order to provide a feasible and effective solution to the organizations. However, more of the existing evaluation and selection system for commercial software system focus more on functional characteristics of the system and do not give that much attention to nonfunctional characteristics of the system and to the mismatch between the user requirement and the desired software system. This study aimed to design evaluation model for business critical software system that focus on handling functional and nonfunctional feature of system and the mismatch with the expected characteristics of the desired software system. The study is conducted using the methodology which involves interview, questionnaire and literatures. Analytical hierarchical processing decision making technique is used for the evaluation process, handling of software mismatch and validating the model. The major results of the study are identifying common business software evaluation criteria attributes and evaluation process used to select the desired software system. The result shows that evaluation aspects of the model is feasible and demonstrate the potential of the system and practicality where it could be applied to the actual environment. Contribution of this study for the research and practical perspectives of software evaluation and selection would improve the decision making and providing systematic guidelines for handling issue in purchasing commercial software.

KEY WORDS

Software evaluation, Software selection, Analytical Hierarchical Processing (AHP), Commercial off-the-shelf (COTS)

CHAPTER ONE
INTRODUCTION

Critical systems is one that has to function correctly in order to guarantee the organizations survival by avoiding loss of life or damage to the health and environment and non-critical system is some types of software will simply not be classified as critical [1]. Obviously the best known non critical software systems are word processors for any machine, spreadsheets for solving problems and graphical design. Non critical tools are fault free, tolerable and stable; their damage will not affect the entire system that much typically which is in the worst case [2]. Selecting software system which is critical for organization needs an evaluation of different factors of the system; and that is difficult task. This research study is to develop a model for evaluating and selecting the appropriate business software system that satisfy the organization requirement. The process of designing the model is based on the requirement collected from organization and different sources. The designing process starts by identifying the basic component used to evaluate software system such as the common business software system evaluation criteria and the process to evaluate the software system alternative. The evaluation criteria and process are selected on the way they are used to achieve the proposed research work. Most of the evaluation criteria identified are from data collected during requirement identification and evaluation processes are selected since it shows a procedure for software evaluation.

1.1 Background

Software system which is functioning correctly and error free can avoid great financial losses for the organization. Business critical systems, security critical systems and safety oriented systems are three basic categories of critical system [2]. Business critical systems are system which have a great important to the organization daily activity and can be categorized to either administrative software system or can be operational software system. Administrative business critical system are work flow system, billing and invoice system and other such system that the organization relied on in orders it to be able to function and take a revenue. Software systems are becoming the most important in all kind of organizations for their day to day activity or processes to gain competitive advantage in their business [3].

However; designing and developing software systems for the market or different purpose have becoming a business trend. Selection of a system that will change the entire organization is a difficult task even in most organizations. When selecting a software system for use, there should be evaluation mechanism or checking way otherwise it could lead to various problems in the company if the desired software system is not selected [3]. The following problem might been observed if best software system is not selected: the system may not support an important business process in the organization, incorrect or incomplete in data produced from the system; customers of the company will not be happy, loss of sales and poor financial performance of the company can be the best examples for uneducated software selection.

1.2 Motivation

Most organizations formerly built software system for their organization which can solve a problem of their internal activities rather than purchasing commercial off-the-shelf package software system [4]. This commercial off-the-shelf software is in some case modified or extended to make fit to the organization, and in other cases the organizations accept the software as it is. There is a good trend in different business company and commercial activities in which built software system or developing software system is changed by software packages acquired in pay-by-use basis [4]. Attention should be given to provide a methods and way to obtain built software by collecting and organizing requirements from users or organizations, by constructing different specifications from the requirements collected from organization and literature reviewed, deriving the properties of the requirement and identifying the requirement into their appropriate grouping which can satisfy the criteria to evaluate the software system in detail. There is a problem in the way that can support the organization to formulate their requirements and procure software packages. The interest or the main motivation in this problem is data obtained from different organization; the difficulty they face when they want to find the appropriate commercial software system. Most organization have not built their own system because many times they may not have the skills to develop the system or design the system they need or they may face lack of resources to do so and finally they are going to purchase an existing off-the-shelf software package.

1.3 Problem statement

Evaluating the functionality and features of software system is basic task in the software procurement process [5]. Several methods have been proposed to handle the commercial off-the-shelf software system evaluation and selection problem but more of these evaluation process have a lack to be accepted and considered as best for evaluating and selecting commercial software in industry or business organization because the methods provided are complex and time consuming in evaluating and selecting the best system [6]. The evaluation and selection of commercial software system is still performed for one time action or it is temporary such as depending on the experiences of software developer or the organization management awareness which is mostly done in the relationship with in particular software system Providing Company or vendors [7]. The main problem in software selection is in the process of systematic evaluation and selection and identifying evaluation criteria. Even though there is an attempt on solving the problem; but still it keeps the organizations under the pressure since they have lack of experiences to plan for the selection process very well [7].

There are also some issues and problems that are still not considered on different software evaluation methods; such as lack of identifying user requirements and software system features, lack of handling general non-functional requirements to distinguish between commercial software system alternatives and lack of identifying mismatch between software alternatives [6]. Various literatures suggest that evaluating and selecting best software system that meets an organization's requirements is a difficult software engineering process. There are many challenges encountered by the organizations when using software system to expand their entire system [8]. The main challenge during expand and develop the system of the organization is the lack of abilities to select the most suitable software that meets their requirements. The inability to find the suitable software system challenge occurs due to availability of much similar software in the market with different capabilities and qualities characteristics. Inaccurate decision in selecting software can affect the organization as entire by increasing the cost, time, effort, performance and quality of the system. The designed model support the software selection decision making process by providing well defined evaluation criteria attribute with processes that guide to select the desired software system.

1.4 Research Questions

Based on the critical situation explained in problem statement as well as what has been discussed in background regarding the role of evaluation in software procurement process; designing and developing a model for evaluation of business critical software system has been selected as the main agenda of the study.

Accordingly the study tries to address the following questions.

- What are the criteria to identify the required software products?
- How effective were the evaluation technique?
- How to design a software evaluation model?

1.5 Objectives

1.5.1 General Objective

The general objective is to design a model for evaluation of business critical software systems which is used as a software procurement process which helps in software selection decision making processes for the system that can fit the organizations requirement.

1.5.2 Specific Objectives

- To identify the processes that supports the software evaluation and selection.
- To examine and propose suitable technique to address the mismatches between the software features and customer requirements.
- To determine the criteria or requirements which are required for successful evaluation and selection process.
- To design software evaluation model.
- To develop a prototype model for evaluation.
- To test or evaluate the system.

1.6 Research Methodology

The study is conducted in order to design and develop a model for evaluating software systems used as a tool for procurement of business software systems. Interview, questionnaire and methods were the research instruments used for the data-gathering [9].

1.6.1 Research Approach

Considering the characteristics of both approaches and the research questions of the study at hand both qualitative and quantitative research methods are used since the desired outcomes of this research will be achieved by focus on both; considering data from which is more subjective and assumption based and quantitative research method which is statistical estimation [10].

1.6.2 Data Collection

There are two main sources, known as primary data which is obtained from specific research and secondary data obtained through several catalogue, website, magazine and newspaper, for gathering data about a particular problem, situation and different phenomenon. Data obtained from these sources are called primary and secondary data [10].

For this research study primary data has been gathered through questionnaire, and by the means of interviews from selected organizations about the system they are using and the way they are going to evaluate and procure a software system for their organization. Secondary data are collected from published data such as existing literature, articles and published materials, catalogue, magazines, newspapers which are relevant to the research problem and research questions.

1.6.4 Research Instrument

In this research work, interviews and questionnaires are used as methods of data collection. Each of them is conducted as follows:

1.6.4.1 Interview

Interviews were designed to produce relevant information from different organization. Interviews facilitate a real interaction between both respondents and researchers. However, in order to establish an efficient interview, researchers need to know about the respondents' backgrounds, values and expectations [10-11]. The interviews questions for this study is prepared from those of the questionnaire and additional questions used to evaluate software systems.

1.6.4.2 Questionnaire

Questionnaire is a form of self-report which is designed to elicit information that can be obtained through the written responses of the subjects [11]. The information obtained through a questionnaire for the study is somehow similar to that of obtained by an interview.

The questionnaires provided were personally distributed to end user, admin and personnel on management level of those three selected organization for eliciting the needed information.
Most of the questions provided in the questionnaire are open-ended questions which were included in the interview guide instead. The questionnaire has 30 questions which had a choice type to select appropriate answers among the choices and the respondent tick on one of the provided answers. There were questions included in the choice which should be answered in a suggestion form. For the question which asked to add a suggestion, the respondent should write his/her comment for the questions asked accordingly in the comment box provided. The questions were designed by the data collected from different organization used to evaluate commercial software system and literature reviewed specially on ISO software quality model.

1.6.5 Programming technique and tools

Microsoft office applications are used to prepare the documentation. After defining the functionalities of the application and then prototype is implemented for demonstration purpose. The prototype system was completely programmed in the Java language. The program codes were written using java. After referring alternatives; MySQL database engine was selected. MySQL is selected because it is an open source, light and it is widely used with Java. Navicat GUI is used as a database manager. It is an intuitive and cost-effective database tool for MySQL. The main reasons for choosing object oriented approach is facilitates reusability, and it supports powerful programming languages such as java and visual basic [12-13].

1.6.6 Document Analysis

Different books and other reliable resources on the web related to the software evaluation and selection process were investigated. Documents, associated with procurement process of software systems and way of evaluate software system in different organization were also investigated. Relevant book to understand the way of solving problems in software evaluation and software selection techniques has been studied and evaluated.

Data collected from selected organization such as the properties of the software system they using, their trend on procuring software system and criteria attributes used for evaluation process are examined.

Other documents from vendors about the way they are providing software for the customer are also examined. The analysis of data has been carried out in a way that once all the data had been collected, conducted and then changed to meaningful form [14].

In order to develop the prototype system many tools and programming options were considered, selecting the appropriate one was a tedious task and required the researcher to refer numerous documents to come up with a good choice. The documents were mainly user manuals, tutorials and documentations of development tools.

1.6.7 Evaluation and testing approach

Case study on which organization need to select software system had been conducted to evaluation the prototype system. The user tested the system with selecting the criteria attribute provided for evaluate the software system and point out the aggregate output of the system. A set of evaluation criteria question were prepared and given to the user to give their rating. The user rate the system in terms of the basic software criteria evaluation attribute such as functionality, feature, the vendor capacity and the cost for using the system. The system is tested with the organization that needs to select appropriate software system. The question provided for evaluation was given to the user and then the user rate each questions according to the system characteristics. The ideal values provided for rating the evaluation questions are *Very high*, *High*, *Medium*, *Low* and *Very low* for the evaluation process. Finally the rating of the user is filled to the system to produce the relevant output for decision.

1.7 Mapping Methods with Specific Objective

The table below shows the map between the specific objectives and the method to address those objectives individually.

Table 1-1: Mapping methods with specific objectives

Specific Objective	Method to address	Remark
• To identify the processes that supports the software evaluation and selection	• Organizational study • Literatures review • Requirement elicitation	• Process for evaluating any system.
• To examine and propose suitable technique to address the mismatches between the software features and customer requirements.	• Literature review, • Survey and interview	• Selecting unmatched criteria with respect to the requirement

• To determine the criteria or requirements which are required for successful evaluation and selection process	• Requirement collection from organizations • Reviewed literature	• More of the identified evaluation criteria were from literature review and collected data from organization
• To design software evaluation model	• Using Word processor	• The architecture incorporates different components.
• To develop a prototype model for evaluation	• Using java	• Eclipse and MySQL server were used
• To test or evaluate the system	• Case study on organization	• Testing performance of the system

1.8 Procedures

The research is conducted using the following procedures.

Data and requirement collection: In data and requirement collection; interview and questionnaire methods were selected for collecting data used for design the solution in the research. Three different organizations personnel were selected for data collection.

Documenting data: The data collected from different source and organizations were well documented in organized form.

Analysis of data: In this research work, many sources of data such as books, journals and websites relevant to the study had been used and the collected data from organization had also been evaluated as of used for design the required model.

Design the model: After collecting the relevant data and find the main problem, the model was designed based on the data and requirement collected from different organization and collected data from different sources.

Construct the system: The system is implemented using one of an object oriented programming language called java and the decision maker can decide based on the system final output.

Test the prototype: The prototype was tested using case study on organization that needs to select software system. There are evaluation questions provided and the user is expected to select the values

or rate the questions provided. The system was expected to processes the input of the user and finally the decision maker use this to make a decision. The final output is aggregate values and mismatched criteria.

1.9 Significance of the study

The designed software evaluation model has a huge significance for many organizations especially using automated system. As it had been stated, one goals of the system is to help the decision making process when selecting a software solution for certain organizations in order to make a quality services for the user and customer. The other significant of the study is to provide software system that supports quality work and contributes to healthy work environment in the organization.

Before selecting a software system for purchase, one has to consider the work environment factor in order to achieve the best and healthy environment. The designed software evaluation model can have a lot of advantage for employees, management and other individual of organization. One of the advantages is for selecting an appropriate software system and also provides evaluation procedures for business software systems. The model contain business software system requirement which was collected from organizations for evaluation purpose. The output of this research study can also be an input for further research in this or related area.

1.10 Scope and Limitation of the Study

The evaluation system was limited to evaluating business critical software system such as ERP, CRM. The thesis will include the implementation of the prototype system with basic software functionality and had not included the office use software system like Microsoft word, Worksheet and any system software. The research had not included legal procedure and the vendor contract negotiation followed after a decision has been made.

1.11 Contribution of the Work

There is a gap in commercial software evaluation and selection which addresses business critical software system issues adequately. This research aimed on filling the gap in selection of software systems that meet organization requirement by studying the important processes to support the decision. The main contribution of the study knowledge acquisition is providing important processes that support commercial software selection and highlights relationships between the evaluation attributes and processes and thus facilitates the relationships between factors in different processes and their impact on success software selection. Therefore, it can be used for current and future academic research.

1.12 Structure of the Document

This research document contains seven chapters; Chapter one contains the overall introduction and background information. Chapter two contains review of literature and related works. The third chapter contains data analysis and organizational study. The fourth chapter describe about the design and implementation. In fifth chapter testing and interpretation is discussed. The sixth chapter contains discussion of findings. The seventh chapter closes the research with concluding remarks together with recommended future works.

CHAPTER TWO
LITERATURE REVIEW

2.1 Overview

The main purpose of the literature review worked was to collect data on previous studies on software evaluation and procurement process; this is to scope out the basic data collection requirements for the primary research conducted, and it formed part of the new research design process. The review approach was conducted in line with current practice in research work. It is now regarded as acceptable for researchers to familiarize themselves with existing research prior to collecting their own data [16]. Relevant literature such as books, journal articles, conference papers and resources from internet that are related to evaluation of software system had been studied. Moreover, to assess what others have done in the area and to understand the problem better, related research works were evaluated. The overall goal of this chapter was initially aimed to identify a place where a new contribution could be made.

2.2 Business critical software

Business critical software system is software that can provide companies more business operation in automated form. The main difference between a business-critical and a regular commercial software system is nothing but by the business. There was no established general definition telling us which software applications are critical to an operation [17]. It is considering the impact of relevant services from software on the business operations, and determines how much value each brings to the business and the impacts of such software parts being unavailable. The impact critical system can be lost revenue, corrupted data or lost user time, as well as indirect and more unidentifiable losses in customer status, goodwill, slipped deadlines, and increased levels of stress among employees and customers. The following are some examples of business critical software system.

Accounting software: are packages that have become commonplace for organizations in recording business transactions, preparing financial statements and analyzing operations. Accounting software has freed the accountant from the manual recording and presenting of financial data.

By using accounting software, financial transactions can be recorded quickly and accurately at a relatively low cost. Moreover, accounting software packages increased overall operational effectiveness by improving both the quantity and quality of management information available [18]. It was single function was to automate the task of double-entry accounting and produce a straightforward balance sheet a years ago. As computers became more robust and integrated databases standardized, accounting software developers added more functions-including cost accounting, manufacturing resource planning, customer resource management, human resources, and payroll.

Enterprise Resource Planning (ERP): is a software type that integrates business functions and data into a single system to be shared within a company. ERP is originated from manufacturing and production planning systems used in the manufacturing industry; it expands its scope into back-office functions such as human resources, finance and production. In recent years this software type incorporated other business extensions such as supply chain management and customer relationship management to become more competitive [19].

Customer Relationship Management (CRM): is a software type for improving profitability by focusing on customer needs and creating a good relationship with the customer [20]. It is a firm of consultants that representing the system which described as companies use to understand their customer groups and respond quickly and timely. This software technology allows businesses to collect and manage large amounts of customer data and carry out strategies based on the provided information. The system data also provide companies with important new insights into customers' needs and behavior and it allow the customer to modify products to target their segments.

2.3.2 Software Procurement

Software procurement is the series of activities and procedures necessary to acquire software system products and services to organization. Business software procurement is the process of purchasing the system that helps to automate the entire function or operation of organizations by providing activities such as providing and approving purchase orders, selecting and ordering a kind of product and service, receiving and matching the invoice and order and paying the bill electronically.

Enable the procurement department to see everything that is ordered, ensure that nothing can be ordered without correct approvals, and lets them get the best value by combining several orders for the same type of good or even getting suppliers to bid for the business [24]. The benefit of using procurement software system for organizations includes ease of administration and potential long-term cost savings.

With a single interface software procurement; management of the organization cuts down the time and effort required to monitor organizational spending and allows procurement managers to control the vendors used by the wider organization so all employees take advantage of the software system and other service.

The process for procuring or purchasing a large software system is long process and complex. It involves numerous internal and external participants in the process, including internal organizational units. Commercial off-the shelf software products bring fundamental changes to the way organizations do their work in terms of different making life easy in the organization [25]. There are some familiar software engineering activities are changed such as requirements analysis is considered as a standard for implementations, the design must be performed with the software package evaluation; and some activities become significant parts for development process, for example product adaptation and integration. The ultimate goal of the software procurement is to save money for any organization that uses the given software system.

The direct saving would be achieved via a decreased software development cost (consequently decreased labor cost from developing the system rather than customized the on the shelf system). The indirect saving would be gained through better and more cost efficient decision.

2.4 Related works

This section presents some related works concerned with the software evaluation issues developed in the previous section.

Khalid Eldrandaly and Soad Naguib [26] design a new decision making approach for solving GIS software selection problem. GIS is a promising branch of information system has achieved considerable success in recent year. This areas of IS has concentrated on the construction of computer based information system that enable capture, model, store, share and manipulate geographical presentations. The decision making approach was integration of expert system and MCDM and provide an advisory system to assist the user during the tool selection procedure. Regarding software selection consideration were observed in the design of the proposed approach and their procedural approach.

Justification phase used to justify the used of GIS software for the application under consideration. Screening phase used to recommend the suitable GIS software package/tool for specific project. Evaluation phase after identifying the GIS software package that are suited for the considered application in the screening phase selection of most appropriate software package can be made. The architecture, the development and implementation of the prototype advisory system were discussed.

Loveleen Kaur & Dr. Hardeep Singh [27] assess some of the various formal software selection techniques that have been introduced in the literature so far, pointing out the benefits and limitations of each and what could be the future work possible with respect to the already introduced techniques. Component based software engineering (CBSE) was the reuse-based, sub-discipline of software engineering which defines implements and integrates independent software components into a system.

Software component selection process has been primarily composed of the incorporation of selection criteria(s) and the calculation of the score of each alternative with respect to the selection criteria(s) which determines which alternative suits our needs the most. *Weighted Scoring Method (WSM)*: Weighted scoring method is one of the oldest techniques used for evaluation and selection of the software packages. Since it is a MCDM approach, this technique was applied to those situations in which there are 'n' number of candidate components and 'm' number of evaluation criteria. *AHP* is the other multi-criteria decision making method apart from WSM and was developed by Dr. Thomas Saaty during the 1970s. It has been applied in wide variety of applications in various fields like banking, education and manufacturing. This method allows consideration of both qualitative and quantitative factors in selecting the best alternative. In short, it uses an 'Eigen vector' approach to derive ratio scales from paired comparisons. Rule based reasoning (RBR) and case based reasoning (CBR) for the effective selection of software components. CBR and RBR are two radical reasoning methods of a knowledge base system (KBS). KBS provides a mechanism to systematically arrange the knowledge and produce a tool that assists decision makers in evaluation and selection of the software packages.

The main objective of the paper was to evaluate the short comings of the various existing techniques of COTS component selection, be it manual, semi-automated or automated. The automated techniques are obviously better than the manual ones because they provide us with a higher accuracy of correctly selecting the desired component out of a wider pool of prospective candidate components.

Table 2-1: Summary of related works

Author	Khalid Eldrandaly & Soad Najib	Loveleen Kaur & Dr. Hardeep singh
Year	2013	2014
Objective	• Designing a new decision making approach for selecting GIS software system	• Evaluate the short coming of the various existing techniques of COTS component selection.
Methods	• Integrate expert system and multi criteria decision making (MCDM) technique.	• Categorizing each technique as manual, semi-automated and automated.
Key finding	• Advisory system for GIS software system selection is designed.	• The automated technique is better than the manual one because it provide higher accuracy.

CHAPTER THREE
DATA COLLECTION AND ANALYSIS

3.1 Overview

In this chapter organizations requirement were identified and analyzed as a result of the requirement findings of the study. The collected data from organizations were processed accordingly in response to the problem discussed in problem statement.

3.2 Data collection

Relevant literature such as books, websites, documents, journal articles, conference papers, and other resources from internet were reviewed. In addition, to assess what others have done, to understand the problem better, and to mark a unique contribution to the area of study. A number of employees in organization with different work roles have been identified for collecting data in the organization. Because of their different roles in the organization, they all had their own concern with the system and thus had different experiences. The questions were prepared base on ISO Software quality model and by discovering an attributes or criteria used to evaluate a business software system from different literatures. Thus, the answers collected used for designing a model for evaluation system. The collected data are from Ethiopian Revenue and customs Authority Trans Ethiopia and commercial bank of Ethiopia.

The software system they are using is 'SIGTAS' to process financial and related activities of the country and 'ASHEHUDA' for controlling the properties imported from outside country. Trans Ethiopia and commercial bank of Ethiopia are also used for data collection about the properties of the system they are using.

3.2.1 Sampling size

Sample size has an effect on how the sample finding accurately represents the population. In this study non probability quota sampling technique has been used. The larger the sample is the more likely that the generalization are an accurate reflection of the population. Sample size depends on the factor such as time and money available to collect the data. For this study 55 questionnaire were distributed out of 44 responses were found usable.

Table 3-1: Distributed questionnaire rate

Location	System admin	Technician	User of system	Manager	Total
CBE	2	2	5	1	10
ERCA	4	10	19	2	35
Trans Ethiopia	1	1	6	2	10

Table 3-2: Completed and valid questionnaire

Location	System admin	Technician	User of system	Manager	Total
CBE	1	1	5	1	8
ERCA	3	8	15	2	28
Trans Ethiopia	1	1	4	2	8

3.3 Evaluation criteria

Evaluation criteria identification process was very important task for understanding, evaluating, and selecting the desired or suitable software system. Evaluation criteria are decomposed from the collected data from different organization as a requirement and reviewed literature. The evaluation criteria were organized in hierarchical decomposition, which starts from high level ranked requirements which is very necessary criteria for evaluation until less important for evaluation process producing pieces of well-defined measurement information [30].

The evaluation criteria designed include functional requirements what the software system expected to do, or the services of software and non-functional requirements of the system, interface, maintenance, support and cost [31].

Evaluation criteria are the facts or standards by which the fitness of products is judged. Software quality characteristic is a set of properties of a software product by which its quality can be described and evaluated. Not all the characteristics of a software product as defined by ISO 9126 are applicable to COTS software system but more of the criteria derived from this standard model.

Table 3-3: ISO 9126 Software Quality Characteristics

Characteristics	Sub characteristics
Functionality	• Suitability • Accuracy • Interoperability • Compliance • Security
Reliability	• Maturity • Recoverability • Fault tolerant
Usability	• Learnability • Understandability • Operability
Efficiency	• Time behavior • Resource behavior
Maintainability	• Stability • Analyzability • Changeability • Testability
Portability	• Installability • Conformance • Replace ability • Adaptability

3.3.1 Selecting evaluation criteria

Perhaps the most critical step in the entire commercial evaluation process is establishing appropriate criteria. The criteria selected will determine whether asked the right questions regarding the viability of the product in the system. There are so many thing which should be look for when considering investing business software systems such as financial management system.

A good choice of software can save a lot of time, money and frustration. Short implementation period, high-performance facilities to communicate with operational and other applications, User-friendly and logically built user interfaces make the need for end-user training, Rapid and efficient support by specialists and the maintenance contract guarantees quick updates and so on. The other way data has been collected from organization regarding the system using features and functionality so as to derive evaluation criteria. The overall data collected from the respondent were summarized in this section. The result of the analyzed data was grouping the criteria attribute according to their categories and identifies basic evaluation criteria used to design the model. The system lack for the intended task given was categorized under 'functional' software quality characteristics attribute for evaluation. The interface of the system whether it was user friendly or not is a grouped to 'Usability' attribute for evaluation. Response time of the system during processing was another software characteristic which was under 'Efficiency' attribute for the evaluation. Mechanism to protect data was grouped under 'Reliability' and identified as an attribute for the evaluation of software system. Capability of the system was grouped under 'Maintainability' software evaluation attribute and the last one for the question asked about whether the system support different platform when processing characteristics was fall under 'portable' software evaluation attribute. The above identified attributes are ISO software quality model used to evaluate any software system. The main identified criteria are divided into sub criteria for controlling purpose.

Additionally two main criteria were identified as basic for evaluation of business software system which are named as vendor criteria and cost criteria. Those criteria were identified during data collection and taken as a result of the study from collected data. Therefore In addition to the software quality model criteria; vendor and cost criteria were also identified as a basic for business software evaluation. Finally the results of the collected data are identified software system evaluation attributes used for design the evaluation model are summarized below.

Based upon the characteristics any business software system necessarily has; the following criteria are selected from the quality characteristics and data collected discussed above to evaluate software system alternatives. Some of the characteristics are changed as to make the criteria used to evaluate business software system. As previously mentioned in the quality characteristics of software system more of the criteria are changed to use to achieve the objective of the study.

Functionality: This characteristic maintains the same meaning for any software system. It expresses the ability of a software system to provide the required services and functions, when used under the specified conditions. For this study sub-characteristic compliance has been discarded from this model and user satisfaction had been added which indicate whether the software system could satisfy the user or not. The following are identified sub criteria for functionality.

Service satisfaction: This property or characteristics determine or is used to evaluate the performance of the software system for intended task of the organization.
User satisfaction: Characteristics used to evaluate the overall environment of the software system whether it satisfy the user in the organization.
Access control: Property used to evaluate the security policy of the software system to protect from unauthorized access threat.
Interoperability: This characteristic determines and is used to evaluate the capability of the software system to communicate with other system for information exchange.

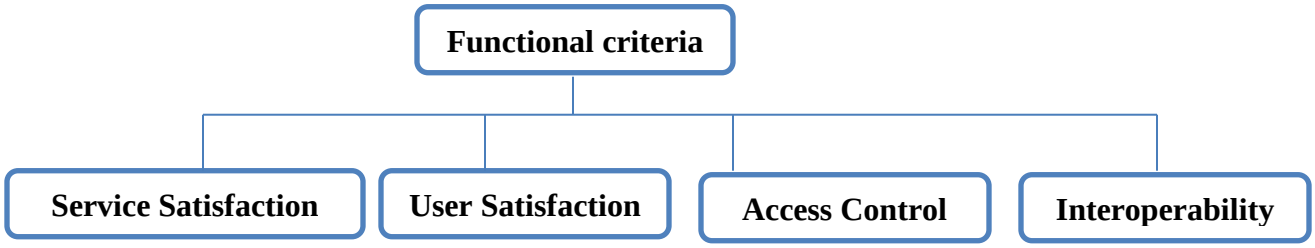

Fig 3-1: functional criteria

Reliability: This characteristic is directly applicable to software component essential for performing the given operation without failure. The Maturity sub-characteristic is changed or discarded form and two different sub characteristics is added backup and correctness. Those added sub characteristics are in terms of the capacity of the system to store data and produce accurate data. On the other hand, recoverability tries to measure whether the system is able to recover from unexpected failures. Fig xxx shows the derived reliability criteria.

Backup: This characteristic determines the capability of the software system to take back up file for later use.

Recovery: This characteristic express the software system capability to recover lost data or damaged data.

Robustness: This characteristic determine the software system whether to perform the operation for long period of time without fail.

Correctness: Property that determine the software system whether the output produced is accurate or not.

Error reporting: This characteristic determines messaging capability of the system if an error occurred.

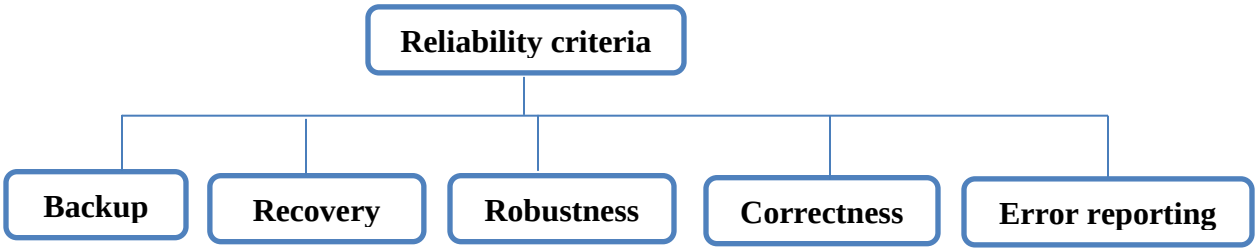

Fig 3-2: Reliability criteria

Usability: This characteristic and all its sub characteristics are example of characteristics because it is where the user has to interact with system mostly. Operability sub characteristic is discarded from and data visualization is added which indicate the clarity of the data. Under this characteristic; three different sub characteristics that used to evaluate business software systems are derived. Fig 3-3 shows us reliability criteria for evaluation software system.

User Type: This characteristic expresses and used to evaluate the capability of the software system whether it supports all kind user of the organization such as beginner, intermediate and expert.

User Interface: This characteristic is determined used to evaluate the interface of the software system.

Data Visualization: This characteristic determines and used to evaluate the clarity of the data produced from the software system.

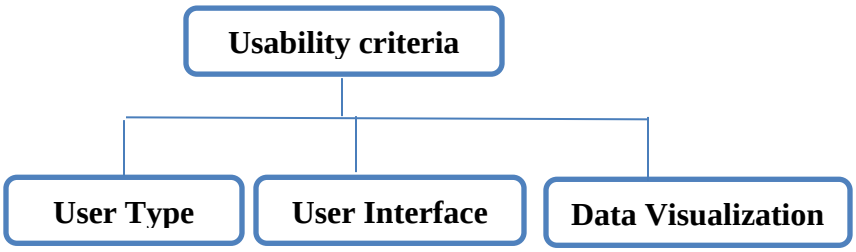

Fig 3-3: Usability criteria

Efficiency: This characteristic describes a measure of the extent to which input is well used for an intended task or function (output). For this study resource behavior is not used rather the times behavior to produce data, although many people prefer to talk about performance and use as sub-classifications. Fig xxx show us the efficiency criteria for the evaluation system

Performance: This characteristic expresses and used to evaluate the overall performance of the system during operation.

Throughput: This characteristic is used to evaluate the rate of the system when producing output from the system.

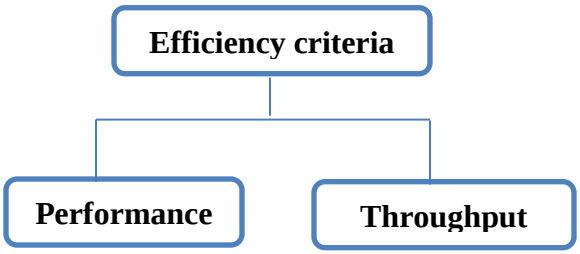

Fig 3-4: Efficiency criteria

Maintainability: This characteristic describes the ability of a software system to be modified including corrections, improvements or adaptations to the software, due to changes in the environment, in the requirements, or in the functional specifications. Thus, changeability is selected for this study as sub-characteristics that must be measured for software system and number of workstation supported.

Upgradability: This characteristic expresses and used to evaluate the scalability of the software system when need to extend or customize part of the system.

Number of workstation: This characteristic determines and used to evaluate the number of workstation or node the software system support.

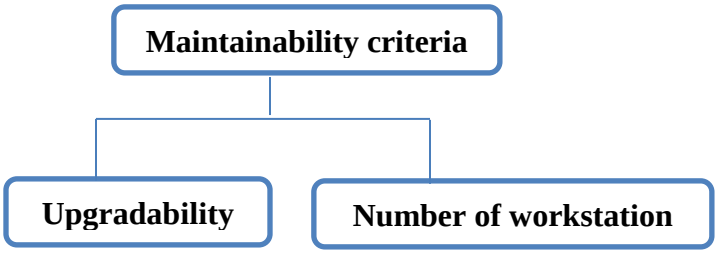

Fig 3-5: Maintainability criteria

Portability: This characteristic is defined as the ability of a software system to be transferred from one environment to another. In this study two different sub characteristics user compatibility and Installability for the evaluation system.

Compatibility: This characteristic determines capability of the software system to run on various computers.

Installability: This characteristic expresses and is used to evaluate the installable procedure of the software system

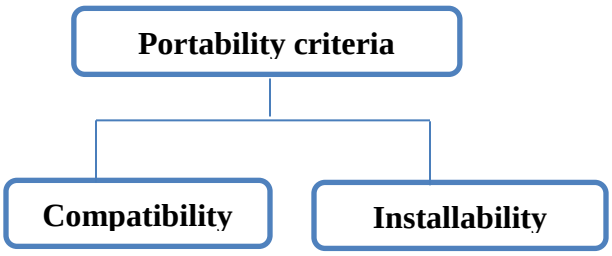

Fig 3-6: Portability criteria

The criteria discussed above are software quality characteristics criteria derived from ISO 9126 quality characteristics. More of the criteria are changed accordingly by selecting criteria used to evaluate business software system. There are also criteria need to include for evaluate business software system regarding the capacity of the software provider (vendor) and cost of the system.

The vendor and cost criteria are from data collected from organization and different sources. The following are vendor and cost criteria used for evaluation process with their sub criteria selected.

Vendor: This criterion characteristic is to measure the capacity of the software provider (vendor). Four different sub criteria are derived for the study.

Financial status: This criteria attribute determine and is used to check the financial status of the vendor.

Provide tutorial: This characteristic expresses or used to evaluate whether the vendor provide a training and tutorial about the system.

Business skill: This characteristic is used to evaluate business running skill of the vendor.

Past experience: Characteristics used to check the past experience of the vendor in developing business software system.

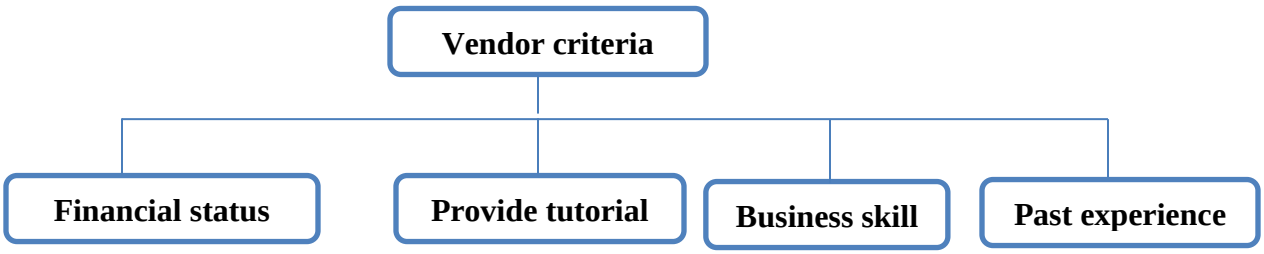

Fig 3-7: Vendor criteria

Cost: This criterion characteristic is to measure the cost of the software system when using. Three different sub criteria are derived for the study.

Training cost: This characteristic determines the cost of training for the software system.

Installation cost: This characteristic used to check the installation cost of the system.

Maintenance cost: This characteristic expresses or used to check the maintenance cost of the system.

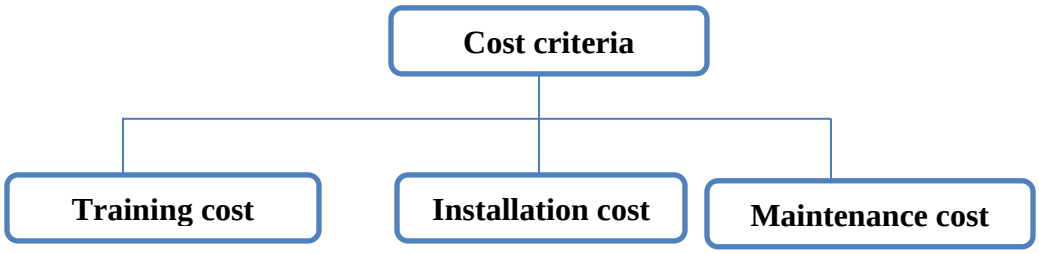

Fig 3-8: Cost criteria

Therefore; for this study 25 different criteria are selected for further evaluation process including functional, nonfunctional or quality, vendor and cost criteria are selected to design the evaluation system.

Table 3-4: Summary of evaluation criteria

Criteria	Sub Criteria	Criteria type
Functionality	• User satisfaction	Functional
	• Service satisfaction	Functional
	• Access control	Functional
	• Interoperability	Functional
Reliability	• Backup and recovery	Non functional
	• Robustness	Non functional
	• Error reporting	Non functional
	• Correctness	Non functional
Usability	• User type	Non functional
	• User interface	Non functional
	• Data visualization	Non functional
Portability	• Compatibility	Non functional
	• Installability	Non functional
Efficiency	• Performance	Non functional
	• Throughput	Non functional

Maintainability	• Upgradability	Non functional
	• Number of workstation	Non functional
	• Financial status	Vendor
Vendor	• Provide tutorial	Vendor
	• Business skill	Vendor
	• Past experience	Vendor
	• Training cost	Cost
Cost	• Installation cost	Cost
	• Maintenance cost	Cost

3.4 Evaluation process

Many organizations faced with the view of evaluate the available commercial products to determine the suitability of the system. The experiences of some organizations are the hard lessons of commercial product evaluation which still have problems that can be traced to the quality of their evaluation process.

3.4.1 The PECA Process

An evaluation process defined by the SEI and National Research Council Canada (NRC), called PECA (Plan, Establish, Collect, and Analyze), helps organizations make carefully reasoned and sound product decisions. The process can be tailored by each organization to fit its particular needs, and is flexible enough to be used within many organizations and with many COTS-based development processes. Although the PECA process was derived in part from ISO 14598, the process was freely adapted to fit the needs of COTS software product evaluation. The process begins with initial planning for an evaluation of a COTS product (or products) and concludes with a recommendation to the decision maker. The decision itself is not considered part of the evaluation process the aim of the process is to provide all of the information necessary for a decision to be made.

PECA was named for the four main activities that make up the process:

- **P**lanning the evaluation
- **E**stablishing the criteria
- **C**ollecting the data
- **A**nalyzing the data

As illustrated in Figure 3-9, the elements in the PECA process are not always executed sequentially. Evaluation events, such as a need for new criteria to distinguish products, unexpected discoveries that lead to the start of a new iteration, or inadequacy of collected data, will direct the process flow. One of the hallmarks of the PECA process is this flexibility to accommodate the realities of COTS-based systems. The process is flexible enough to be used within many COTS based development processes, but is particularly this process has a lack on fully support the decision making process especially on the process of evaluating the software product and on mismatched criteria identification.

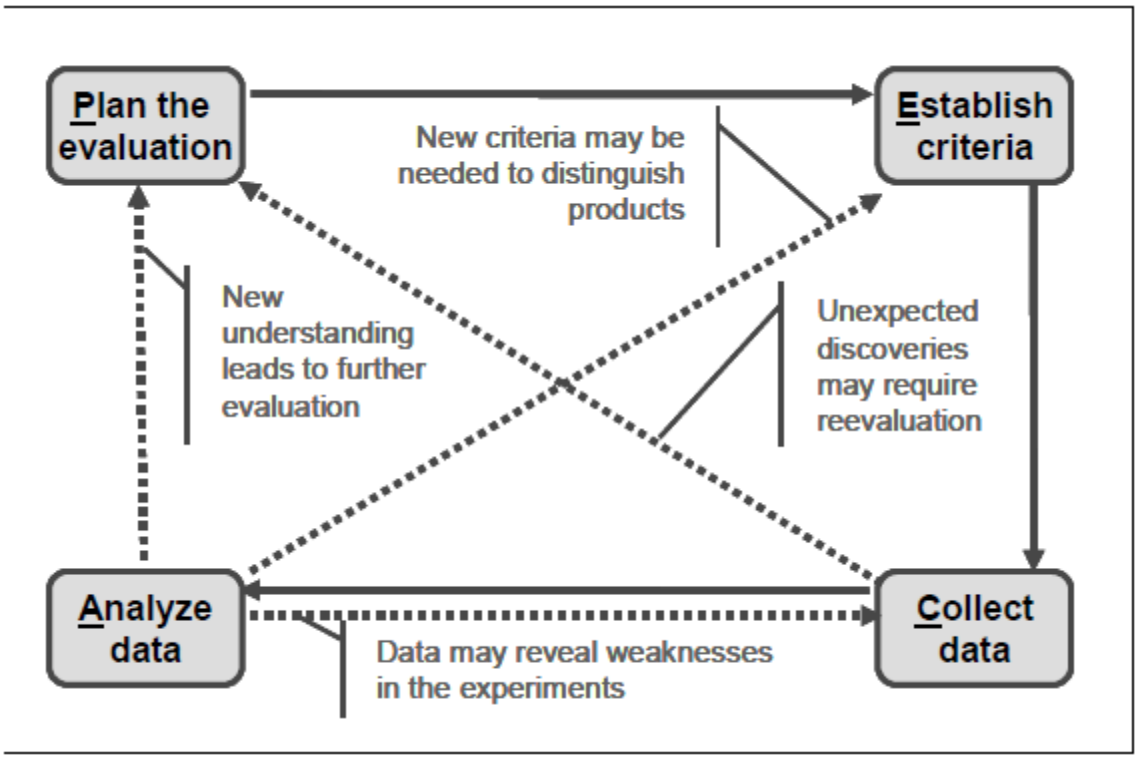

Fig 3-9: Recommended PECA process

3.4.2 The STACE process

This approach pre-supposes that essential customer requirements which have been acquired, possibly in the form of high level scenarios and use cases. These requirements can be acquired using simple techniques such as interviews. It also assumes that the customers will be part of the evaluation team. The procedure of STACE process is:

- Selection of the technology or standard on which the products will be based and proceeds to the selection of the COTS products.
- Define social-technical evaluation criteria for COTS products based on the selected technology.
- Search and screen for available COTS products. Search the marketplace to identify candidate COTS components through market surveys and other techniques.
- Revise requirements and social-technical criteria based on available COTS products. The selection of the "best" among the packages available depends on the assessment of their compatibility with the requirements specification.

3.4.3 Selecting evaluation process

The STACE process defines only the customer requirement in the evaluation while PECA process support by providing relevant information for the decision making. Both STACE and PECA didn't support the decision making process expected in this study fully. But finally a new evaluation process had been designed by deriving the best character of the each evaluation process of the existing methods and customizing to make it fit for this study. Commercial software solutions require particular evaluation processes that affect the way to select the appropriate software system. For example, using ERP packages has huge process implications. The processes actually used by people in an organization are rarely the ones found in available documentation, so it will probably be necessary to find out what the users are really doing and different evaluation consideration.

Considering the kind of software system which is going to be evaluated and their characteristics; three common evaluation and selection process had been selected for the study including activities inside in different phase; those selected process are namely:

- Preparation process
- Evaluation process
- Selection process

Preparation phase

In this process potential evaluation criteria attribute used were determines and discussed. This process was achieved by listing the criteria used to screening the commercial software alternative and selects the most suitable alternative that can be estimated best in resources handling and functioning. The main purpose of this process is to collect and prepare the basic data that needed for further detail evaluation. Therefore, the process starts with establishing the evaluation criteria including both functional and non-functional criteria in order to identifying the potential software candidates [34]. As stated in requirement analysis result almost 25 different criteria have been derived for further evaluation process whose detail were discussed above.

Evaluation phase

The evaluation process includes different activities find out the candidate software system for the evaluation process, identify evaluation technique for and identify mismatch between software systems. Each of the evaluation criteria, the identified requirement and software alternatives have important role on the evaluation success [34]. This process plays a vital role to determine how each of the software alternatives achieves the evaluation criteria and it also determine mismatch between software components. The objective of this process was to check software alternative against the evaluation criteria and sort the software alternatives based on their rank. The other thing was determine how the software system can achieve the organizations goal based on the requirement.

Selection phase

The outputs of the evaluation process are aggregate value and the criterion that made a mismatch with the expected software characteristics. The system display the software system detail weights, the point where mismatch between the software system and user requirement and user. Those data had been organized and interpreted into more understandable information [37]. However, the decision maker requires knowing about each selection process, and identifying mismatches between the software system features and customer requirements in order to select the required system [38]. It is necessary to perform a careful impact analysis of the selected product over the organization. The following table show us how aggregate value of alternative software component for decision making.

3.5 Evaluation Methods

Good process is imperious for the consistent evaluation of commercial software system. Successful evaluation relies on more than just a process. In addition, a set of techniques is needed to complete each step in the evaluation. While the evaluation process is a high-level description of what we need to do, techniques are low-level descriptions of how to do it. Some steps in the selected evaluation process require techniques that might fulfill the evaluation objective. For example, requirements elicitation is a technique used to draw out stakeholders' expectations about the new system.

3.5.1 Analytical Hierarchical Process (AHP)

The method is defined as a mathematical approach to a given decision making problem that also incorporates and builds upon subjective input from the personal involved in the decision making process [21]. As has been stated earlier in this thesis in the description of AHP, this combination of decision makers' judgment combined with AHPs use of mathematical constructs in the form of an eigenvector is what makes this methodology so flexible and popular in multiple criteria decision making domains.

In the first stage of AHP the main objective is formulated, for example “Select a suitable system”. For this evaluation problem, the quality characteristics would be inserted into the AHP hierarchy-tree below the main objective followed by the identified system alternatives.

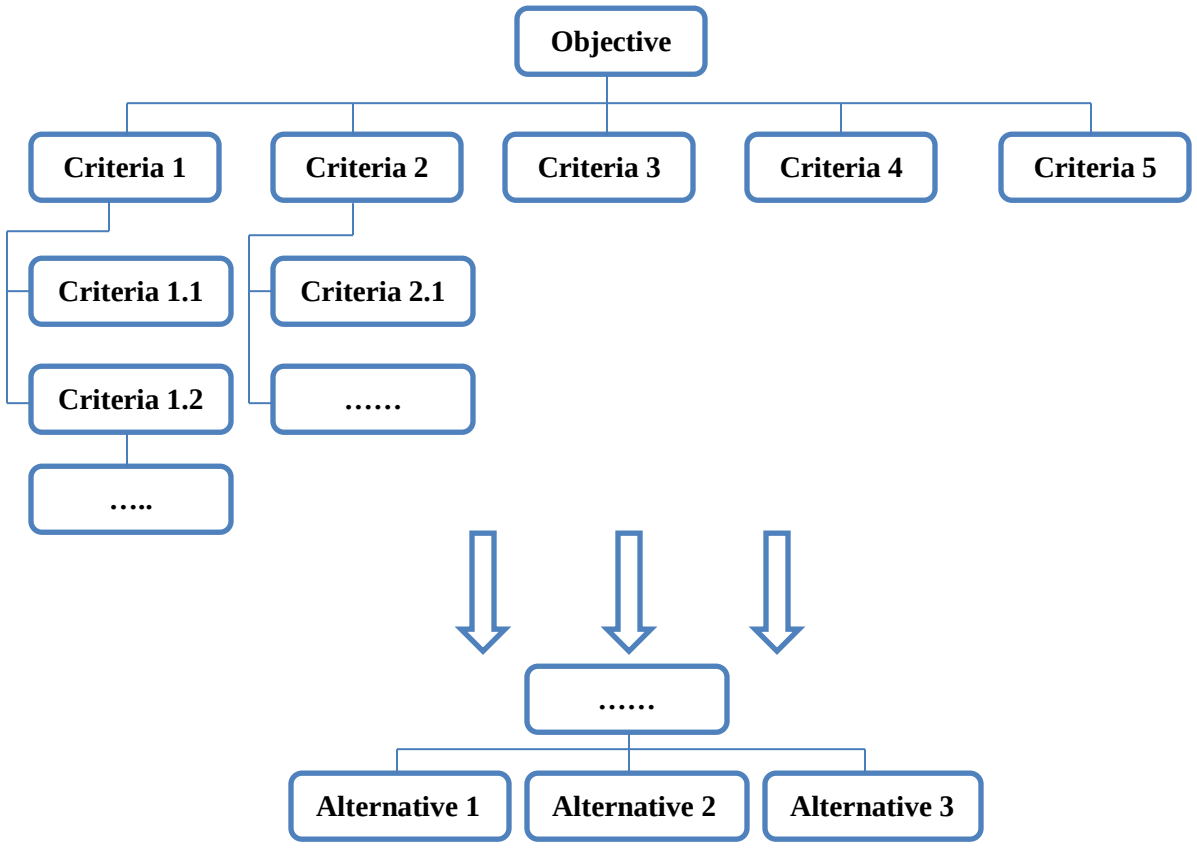

Fig 3-10: AHP hierarchy tree

3.5.2 Multicriteria Decision Aid (MCDA)

The Multiple Criteria Decision Aid methodology has several similarities with AHP, which according to [23] is a decision making tool based on the core functionality of MCDA. MCDA itself has expanded into a great many number of derivations of the original MCDA methodology, each aimed at fitting certain types of evaluation problems better than the standard method. In MCDA, the most important step is the selection of attributes for the object under evaluation. These attributes greatly affect the direction and focus of the evaluation and must consequently be carefully. Thus, an MCDA approach to

the evaluation problem would support the incorporation of quality characteristics into the process, while also having the capability to support other important factors such as cost.

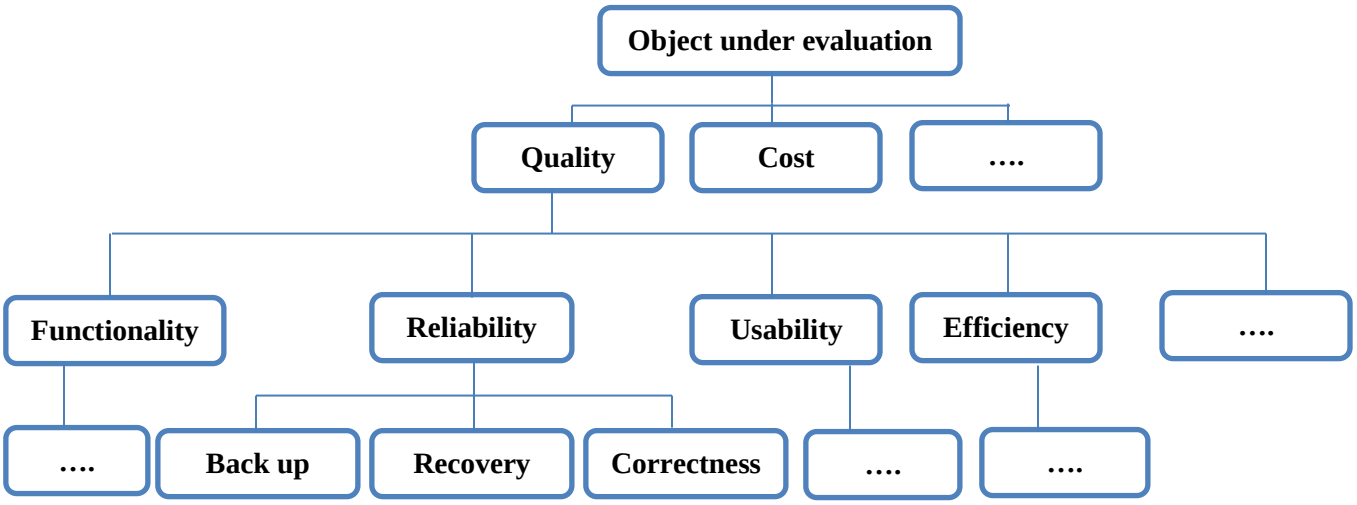

Fig 3-11: MCDA attribute hierarchy tree

3.5.3 Goal, Question, Metric (GQM)

The Goal Question Metric approach to an evaluation problem is different from the more traditional decision making methodologies. Here the main focus is on the actual goals set forth by the organization, both for the project in question but also long-term business goals and strategic goals [30]. Without a properly defined goal, the methodology will most likely generate less usable results. A series of questions are then associated with the goal/goals in an effort to break down the objective into measurable pieces or metrics that are in turn connected to the questions. This mode of procedure makes it less straightforward to incorporate the quality characteristics. In order to include the characteristics into the process, they have to be derived as questions from a goal.

The outcome of a successfully implemented GQM method is not necessarily a recommendation as to which out of a set of N alternatives is the best, although it could be utilized for that purpose also. In fact, alternatives do not even have to be part of the equation. Instead, GQM can be implemented in such a way that its outcome offers is a greater insight into the problem and an opportunity for the organization to further refine their requirements definitions.

3.5.4 Selection of evaluation method

Having described the methods and discussed their suitability for software quality characteristics and evaluation process, no one method can be singled out exclusively if they are capable of incorporating software qualities as criteria and attributes. To reach a conclusion as to which method to choose, one has to look at the following aspects and characteristics of the methods in conjunction with their ability to evaluate software system. Features supported by the evaluation methods should be considered is described in table 3-5.

Table 3-5: Summary of features supported by the methods

Feature	AHP	MCDA	GQA
• Support software quality characteristics	✓	✓	×
• Synthesize pairwise comparison	✓	×	×
• Ability to rank alternatives	✓	✓	×
• Include other organization factor	✓	✓	✓
• Long term organizational perspective	✓	×	✓
• Well defines structure	✓	✓	✓
• Easy to follow	✓	✓	×

Based up on the key feature the evaluation methods support during evaluation AHP method has been selected because it satisfy all of the evaluation features stated above.

3.6 Case study on failed system

3.6.1 Implementing property control system in ERCA

The Ethiopian Revenue and Custom Authority system case scenario is based on reality investigation. In 2007, ERCA set out to upgrade its working system legacy systems by procuring a software system that can control the imported properties of the customer and related activities. Then after the organization decides to purchase a software system called '*ASHEHUDA*', to control properties of the customer imported from outside country. The system was about 2700$ system. Different business process and systems issues cause the system to paralyzed, leading to a drop more percent in quarterly profits and decline in stock price. In the analysis that follows, system failure as a case study to offer advice on how effective the evaluation process were exposure to failure risks and related damages. The system was DOS version with different sub function inside. The organization made a mistake on systems testing for the sake of convenience.

As a result, effective critical data producing, protecting data from threat, interoperability of the system, ease of the system to all kind of employee of the organization and customization capability issues may have remained undetected during evaluation and selection. Testing the component of the system is one should never be compromised. The users test the system to make sure that all of the various part work together as intended. Finally the organization decided to change the system with new customizable web based version system that can fit more of requirement of the organization intended task.

3.6.2 Implementing finance system in Trans Ethiopia

Tran Ethiopia is one of governmental organization in Ethiopia which provides a transportation service for huge properties to customer. This case study is also based on reality scenario investigation. The organization purchases a financial software system to upgrade the payroll system. This system also has plenty of problems which made because of unorganized testing different parts of the system during evaluation.

As a result error reporting capability in process if problem occurred, inability to run on different system, backup and recovery of data, processing for long time consistently and provide training for the system installed are issues remain undetected. Problem stated above made the employee of the organization unsatisfied. Therefore, the organization decides to purchase customizable and integrated ERP system to solve the problem.

CHAPTER FOUR
DESIGN AND IMPLEMENTATION

4.1 Introduction

The business critical software evaluation model is a model that used to evaluate software system based on the user requirement and the user can select the system that can meet the criteria stated. Matching software product to decide on the best system based on their final. The software evaluation system was implemented in a way that involves easy movement or process from one to other parts.

Selecting the software system that meets organizational requirements and business objectives with perfect evaluation process could prove to be a challenge considering the number of vendors and software available. The designed evaluation models may provide a standardized way of establishing successful procedures in the practice of software system evaluation especially for business critical system, and the model is best tool for comparing different business software system [29].

4.2 Design of evaluation model

The overall system design is shown in fig 4-3. The model would provide decision makers more confidence in the decisions they made for software selection process. Different factor have been included in the model which might use as basic in software evaluation and selection process starting from organizational requirement analysis regarding the software system functionality and features besides to their characteristics. The requirement collected is an attribute used for software evaluation and the way how software system could be evaluated in procurement process before use in different organization. The architecture has different components specially evaluation criteria, requirement collected from organization and the process to evaluate the software system. Evaluation criteria is consists of functional criteria, nonfunctional criteria, vendors, cost for the evaluation process.

The functional criteria are what the system should do mainly like it should process the data accurately and response the exact result and it should support to simultaneous process.

The nonfunctional criteria are those can support the system in different feature and quality form; some of nonfunctional criteria are like reliability for the system to function consistently without an error; Usability is for making the system usable for the user or making the system easy for use; Maintainability is to maintain the system if any error or fail is occurred during process.

The data collected from organizations are also used for identifying criteria used for evaluation and identifying features of the software product and also the way they need to evaluate software system before. Those collected data are used for the design of the model in a way to include the basic functionality that any business software system includes and what the business software system should look like.

The system should process the data in different phases. Preparation process is for identifying a kind of software system to screen or to evaluate and identifying different criteria for evaluation. The evaluation process has different techniques or method to process data for the evaluation and comparing the software product. Analyzing the score of each software system product for evaluation and the last process is selection process which is the step to compare the score of the software product have during evaluation process and identify/select/ the best software system based on the final score of each software system. Generally the designed model has its own phases for process of data in different form. The system should pass all those stage to help the user on decision making process. Figure 4-1 shows the proposed model.

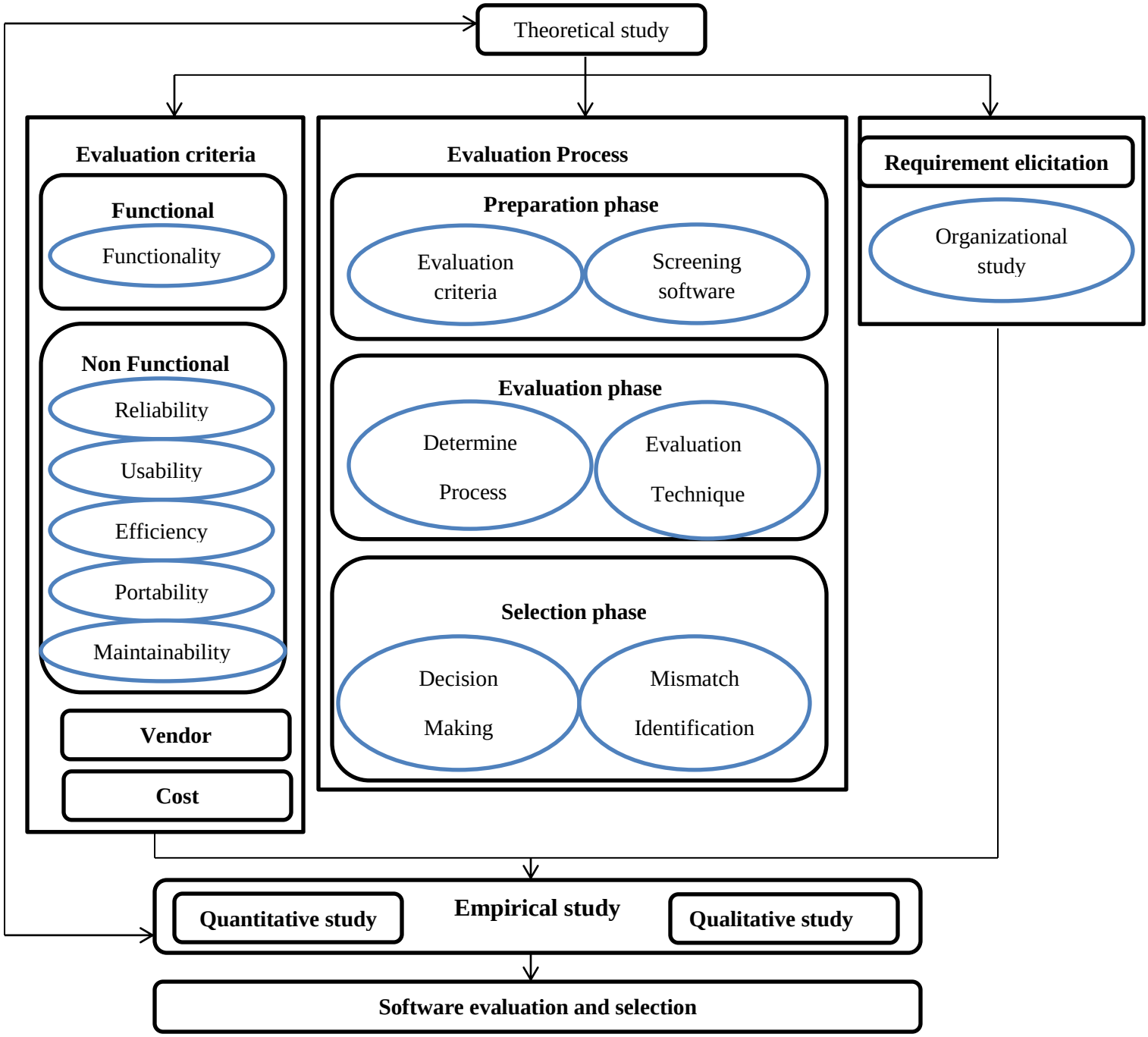

Fig 4-1: Proposed software evaluation process model

4.3 Evaluation using AHP

AHP is multi-criteria decision making method [35]. It has been applied in wide variety of applications in various fields like banking, education and manufacturing. In short, it uses an 'Eigen Vector' approach to derive ratio scales from paired comparisons. How important is criteria one relative to other. The performances of the various alternatives are calibrated with the help of a numeric scale which ranges from 1 over 9 up to 9. The score 1over 9 means the least value give, a score 1 means it is equal and a score 9 means most important. The AHP decision making technique is based on three different principles: decomposition, comparative judgments; and synthesis of priorities.

- Decomposition step defining the problem, state the criteria that used for the selection process and structuring the criteria.
- Comparative judgments define comparison matrix at each hierarchy level by comparing pairs of criteria or sub criteria and rate the relative importance of these criteria using pair-wise comparisons.
- The final step is to calculate aggregate performance value for each alternative and ranking of alternatives according to their performance.

Figure 4-2; shows us how AHP method process in selection from alternatives.

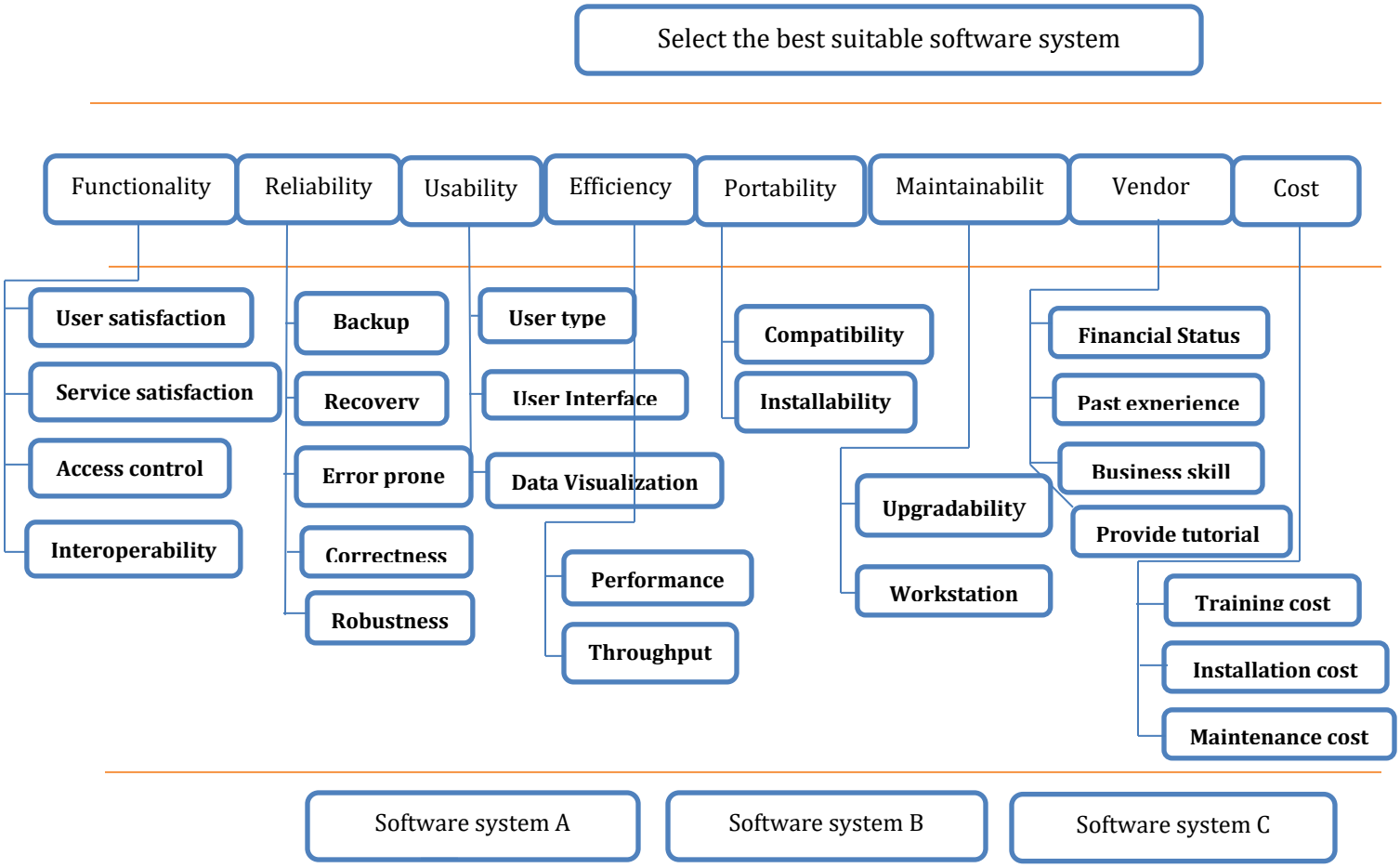

Figure 4-2: Decision hierarchy for software system selection [36]

4.3.1 Pairwise comparison

A technique that has gained some popularity is pair-wise comparison. Pair-wise comparison is at the core of the Analytical Hierarchy Process (AHP), which provides a means of making decisions or choices among alternatives, particularly where multiple criteria are involved. In pairwise comparison, the relative importance of each criterion against every other criterion is determined, often in a group session that was preceded by individual. In the case where criteria are grouped into categories, such as when one does criteria decomposition, each criterion is compared to the other criterion in its group. As a rating for each possible pairing is determined, a formula is applied that identifies inconsistencies (e.g.,

Functionality two times as important as Reliability, Reliability two times Usability and Usability two times Efficiency etc.).
Finally, when no inconsistencies remain, weights for the criteria are computed. Identifying the relative importance of criterion pairs are critical that the imperative to assign a value.

This section describes how to make a reciprocal matrix from pairwise comparison. There are 8 main criteria to be compared and made subjective judgment on which criteria like to be prioritized. There are 28 ways to estimate relative importance of the criteria from 8 criteria and the same procedure is applied for the rest of sub criteria to bring with their weight or prioritize them.

- Functionality is 2 times as importance as Reliability
- Functionality is 3 times as importance as Usability
- Functionality is 4 times as importance as Efficiency
- Functionality is 5 times as importance as Portability
- Functionality is 6 times as importance as Maintainability
- Functionality is 7 times as importance as Vendor
- Functionality is 7 times as importance as Cost
- Reliability is 2 times as importance as Usability
- Reliability is 3 times as importance as Efficiency
- Reliability is 4 times as importance as Portability
- Reliability is 5 times as importance as Maintainability
- Reliability is 6 times as importance as Vendor
- Reliability is 6 times as importance as Cost
- Usability is 2 times as importance as Efficiency
- Usability is 3 times as importance as Portability
- Usability is 4 times as importance as Maintainability
- Usability is 5 times as importance as Vendor
- Usability is 5 times as importance as Cost
- Efficiency is 2 times as importance as Portability
- Efficiency is 3 times as importance as Maintainability

- Efficiency is 4 times as importance as Vendor
- Efficiency is 4 times as importance as Cost
- Portability is 2 times as importance as Maintainability
- Portability is 3 times as importance as Vendor
- Portability is 3 times as importance as Cost
- Maintainability is 2 times as importance as Vendor
- Maintainability is 2 times as importance as Cost
- Vendor and Cost are equal important

4.3.2 Making comparison Matrix

Table 4-1: Criteria comparison matrix

	Functionality	**Reliability**	**Usability**	**Efficiency**	**Portability**	**Maintainability**	**Vendor**	**Cost**
Functionality	1	2	3	4	5	6	7	7
Reliability	1/2	1	2	3	4	5	6	6
Usability	1/3	1/2	1	2	3	4	5	5
Efficiency	1/4	1/3	1/2	1	2	3	4	4
Portability	1/5	1/4	1/3	1/2	1	2	3	3
Maintainability	1/6	1/5	1/4	1/3	1/2	1	2	2
Vendor	1/7	1/6	1/5	1/4	1/3	1/2	1	1
Cost	1/7	1/6	1/5	1/4	1/3	1/2	1	1

4.3.3 Compute Eigen vector of the matrix

Having a comparison matrix, now the next step is to compute *priority vector*, which is the normalized Eigen vector of the matrix. Nevertheless it is easy to compute because all we need to do is just to normalize each column of the matrix. To find the Eigen vector of the matrix

Matrix N for (n=8) criteria

$$N=\begin{bmatrix} 1 & a_{12} & a_{13} \\ a_{12}^{-1} & 1 & a_{23} \\ a_{13}^{-1} & a_{23}^{-1} & 1 \end{bmatrix}$$

Sum of columns S_{c1} S_{c2} S_{c3}

Normalize and calculate Eigen vector:

$$|N| = \begin{bmatrix} 1/s_{c1} & a_{12}/s_{c2} & a_{13}/s_{c3} \\ a_{12}^{-1}/s_{c1} & 1/s_{c2} & a_{23}/s_{c3} \\ a_{13}^{-1}/s_{c1} & a_{23}^{-1}/s_{c2} & 1/s_{c3} \end{bmatrix} \quad \text{Eigen Vector } x = \begin{bmatrix} \frac{\sum \text{row1}}{n} \\ \frac{\sum \text{row2}}{n} \\ \frac{\sum \text{row3}}{n} \end{bmatrix}$$

As a result we sum each column of the reciprocal matrix to get

Table 4-2: Reciprocal comparison matrix

	Functionality	**Reliability**	**Usability**	**Efficiency**	**Portability**	**Maintainability**	**Vendor**	**Cost**
Functionality	1	2	3	4	5	6	7	7
Reliability	1/2	1	2	3	4	5	6	6
Usability	1/3	1/2	1	2	3	4	5	5
Efficiency	1/4	1/3	1/2	1	2	3	4	4
Portability	1/5	1/4	1/3	1/2	1	2	3	3
Maintainability	1/6	1/5	1/4	1/3	1/2	1	2	2
Vendor	1/7	1/6	1/5	1/4	1/3	1/2	1	1
Cost	1/7	1/6	1/5	1/4	1/3	1/2	1	1
	$\frac{157}{60}$	$\frac{277}{60}$	$\frac{449}{60}$	$\frac{34}{3}$	$\frac{97}{6}$	22	29	29

Then divide each element of the matrix with the sum of its column to get normalized relative weight.

The sum of each column is 1.

Table 4-3: Normalized relative weight

	Functionality	Reliability	Usability	Efficiency	Portability	Maintainability	Vendor	Cost
Functionality	60/157	120/277	180/449	12/34	30/97	6/22	7/29	7/29
Reliability	30/157	60/277	120/449	9/34	24/97	5/22	6/29	6/29
Usability	20/157	30/277	60/449	6/34	18/97	4/22	5/29	5/29
Efficiency	30/314	20/277	30/449	3/34	12/97	3/22	4/29	4/29
Portability	12/157	30/554	20/449	3/68	6/97	1/11	3/29	3/29
Maintainability	10/157	12/277	30/898	1/34	3/97	1/22	2/29	2/29
Vendor	60/1099	10/277	12/449	3/136	2/97	1/44	1/29	1/29
Cost	60/1099	10/277	12/449	3/136	2/97	1/44	1/29	1/29
Sum	1	1	1	1	1	1	1	

The normalized principal Eigen vector is obtained by averaging across the row or by adding each row and divide by the number of criteria compared (n=8). Finally the following result had been obtained from the comparison. The Eigen vector is also called ***priority vector.***

Table 4-4: Eigen Vector and weight

Criteria	Priority Vector (Eigen Vector)	Weight
Functionality	0.3291	32%
Reliability	0.2284	23%
Usability	0.1572	15%
Efficiency	0.1073	11%
Portability	0.0723	7%
Maintainability	0.0579	6%
Vendor	0.0314	3%
Cost	0.0314	3%

Since it is normalized, the sum of all elements in priority vector is 1. The priority vector shows relative weight among the criteria. In this study for example Functionality is 32% with the highest rank followed by reliability and usability and so on. Functionality is much important than the rest of criteria for computing. The same procedure is applied for the rest of sub criteria for the purpose of prioritizing or gives a weight for.

Table 4-5 shows us weight of each sub criteria for further evaluation which is calculated using the Eigen vector value.

Table 4-5: Summary of weight of sub criteria

Weighting and prioritizing criteria					
Criteria	Eigen vector	Weight	Sub criteria	Eigen vector	Weight
Functionality	0.3291	32%	• Service satisfaction	0.4658	15%
			• User satisfaction	0.0959	3.3%
			• Access control	0.2771	8.6%
			• Interoperability	0.1610	5.12%
Reliability	0.2284	23%	• Backup	0.0894	2%
			• Recovery	0.0894	2%
			• Error reporting	0.2488	6%
			• Correctness	0.3777	9%
			• Robustness	0.1944	4%
Usability	0.1572	15%	• User type	0.2243	3%
			• User interface	0.156	2.4%
			• Data visualization	0.619	9.6%
Efficiency	0.1073	11%	• Performance	0.4444	7%
			• Throughput	0.3333	4%
Portability	0.0723	7%	• Compatibility	0.1665	2%
			• Installabilty	0.3333	5%
Maintainability	0.0579	6%	• Upgradability	0.3333	4%
			• No of workstation	0.1665	2%

Vendor	0.0314	3%	• Financial status	0.0975	0.5%
			• Past experience	0.3274	1%
			• Business skill	0.1450	1%
			• Provide tutorial	0.4299	1.5%
Cost	0.0314	3%	• Training cost	0.1722	0.5%
			• Maintenance cost	0.4777	1.5%
			• Installation cost	0.35	1.5%

4.4 Software Product Scoring Method

This section determined the steps to determine how software products will be scored against the evaluation criteria. Evaluation and score products against a set of evaluation criteria in order to determine the best choice to meet their customer needs. There must be a clear assessment of the products and provide a rationale that can be used to make and justify decisions. The process involved. The study Likert scaling method which is extremely popular for measuring attributes, because the method is simple to administer. With the Likert scale the respondent indicate their own attributes by checking how strongly they are agree or disagree with carefully worded statement that ranges from very positive to very negative towards the attitudinal objects. Respondents generally choose from five alternative (Very high, High, Medium, Low, Very low).

The following way could be used for scoring the values.

0 if the alternative fully dissatisfies evaluation criteria and requirement (Very low).
1 if the alternative dissatisfies evaluation criteria and requirement (Low).
2 if the alternative partially satisfies evaluation criteria or requirement (Medium).
3 if the alternative partially satisfies the evaluation criteria and requirement (High).
4 if the alternative fully satisfies the evaluation criteria and requirement (Very high).

The alternative that fully satisfies the criterion and requirement during testing will receive a score value of 4, the alternative that partially satisfies the criterion and requirement will receive a score of 3, the alternative that partially satisfies the criterion or requirement during testing will receive a score 2, the alternative that dissatisfies the criterion and requirement during testing will receive a score 1 and the alternative that fully dissatisfies the criterion and requirement will receive a 0 for the software product types. The detail of the scoring values is discussed in table 4-5 with their respective evaluation criteria.

The following questionnaire is provided for the user to measure the characteristics of the system with regards to its vision and organizations requirement.

The user is required to select the values for all questions and the system process using evaluation methods and techniques discussed before and display the final table having their final aggregated value for decision making process.

Table 4-6: Software System Criteria Scaling

Criteria	Very low	Low	Medium	High	Very High
How could you describe the software system is performing for the intended functions.	0	1	2	3	4
How could the users satisfied on the system	0	1	2	3	4
Ease of the software for user to learn and operate	0	1	2	3	4
Capability of the software to present data effectively	0	1	2	3	4
Error reporting and messaging ability of the software system	0	1	2	3	4
Capability of the software system to operate consistently without crashing.	0	1	2	3	4
Capability of the software system to support backup and recovery	0	1	2	3	4
Capability of the software system to produce result in reasonable amount of time related to data size.	0	1	2	3	4

The software system to support for managing and enforcing access right.	0	1	2	3	4
Cost of additional requirement to train the software system.	0	1	2	3	4
The ability of the system to run in different computer platform	0	1	2	3	4
How do you rate the correctness of the output of the system	0	1	2	3	4
The maintenance cost of the system	0	1	2	3	4
Ease of the interface of the system	0	1	2	3	4
The capacity of the system to handle the task	0	1	2	3	4
Ease of Installability of the system	0	1	2	3	4
Compatibility of the system with other hardware.	0	1	2	3	4
The capability of the system to upgrade and customize	0	1	2	3	4
The cost for implementing the system	0	1	2	3	4
Number of workstation or computers the system support during process	0	1	2	3	4
The financial status of the vendor	0	1	2	3	4
The tutorial provided for the system	0	1	2	3	4
Business skill of the vendor in system development	0	1	2	3	4
Past experience of the vendor in software development	0	1	2	3	4
Installation cost of the system	0	1	2	3	4

After identifying the necessary requirement the final step is to assign weights to each criterion. Those weights serve as scaling numbers to specify the relative importance of each criterion. Because they are scaling numbers that specify comparative importance in the overall set of criteria and they should be nonnegative numbers. There are methods for choosing weights; the choice was depends on the decision

maker needs to follow, level of detail desired for the weights, and the computing way available for calculating the weights. Selecting the scaled value is not as simple; the criterion which is going to be evaluated should fulfill some of pre attribute criteria to get the specified scaling value. For example to evaluate the performance of the system it needs to fulfill criteria to measure performance then after can select value score to evaluate the system. User satisfaction is evaluated through the criteria such as well-trained of the system, well supervised, standard conduct and overall satisfaction.

The system use the weight of each derived sub criteria and the rating values from the user to process based on the evaluation methods selected. It multiply the weight and the rating to find the aggregate value for comparing and the software component with the highest value had been taken as the best software system by assuming it fulfill more requirement of the organization rather than that score less in value.

4.5 Matching software system

The outputs of the evaluation process are aggregate value and the criterion that made a mismatch with the expected software characteristics. The system display the software system detail weights, the point where mismatch between the software system and user requirement and user. Those data had been organized and interpreted into more understandable information [37]. However, the decision maker requires knowing about each selection process, and identifying mismatches between the software system features and customer requirements in order to select the required system [38]. It is necessary to perform a careful impact analysis of the selected product over the organization. Table 4-7 show us how aggregate value of alternative software component for decision making.

Table 4-7: software system matching

Software system	Matching result (%)
System A	Value A
System B	Value B
System C	Value C

4.4.1 Mismatch identification

The mismatch between software system and user requirements was arising from unequal need between the user requirement and the system requirement [39]. Identifying the mismatches between software features and customer requirements have an important advantage for supporting the decision making in software selection process or better understanding of the mismatches between the software system and user requirement that would support and provide valuable insight on the decision of software selection [40-41].

Scenario for the mismatch

The software system selected should fulfill the requirement stated by the organization. The way of identifying the mismatch between the software product and the user requirement is based on suggested values given by the evaluator or decision maker. The software product is expected to have the highest score value stated for each evaluation criteria attribute so that it should fulfill the requirement set. For example; user satisfaction is a criteria to check whether the system satisfy the user in different case; and the system expected to have a capability to satisfy the user very well; so that this criteria need to get the highest value set which is very high (4) or high (3) to mean that the system enough satisfy the requirement. If the decision maker suggest less than the maximum number such as medium (2) meaning here is a mismatch to the requirement of the expected system; so that this criteria are collected as mismatch and displayed for the final decision. The same procedure is applied for the rest of all criteria attribute identified.

4.5 Implementing the system using java

Java is a high-level programming language originally developed by Sun Microsystems. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. In this section the general architecture of the implemented prototype model is discussed. To develop the evaluation application java Eclipse neon was used. Therefore a new java project was created and imported. The main function of the evaluation system is start after SEBegin () and InitialDialog () the next step is loading a first page. When the system starts the following first dialog is displayed.

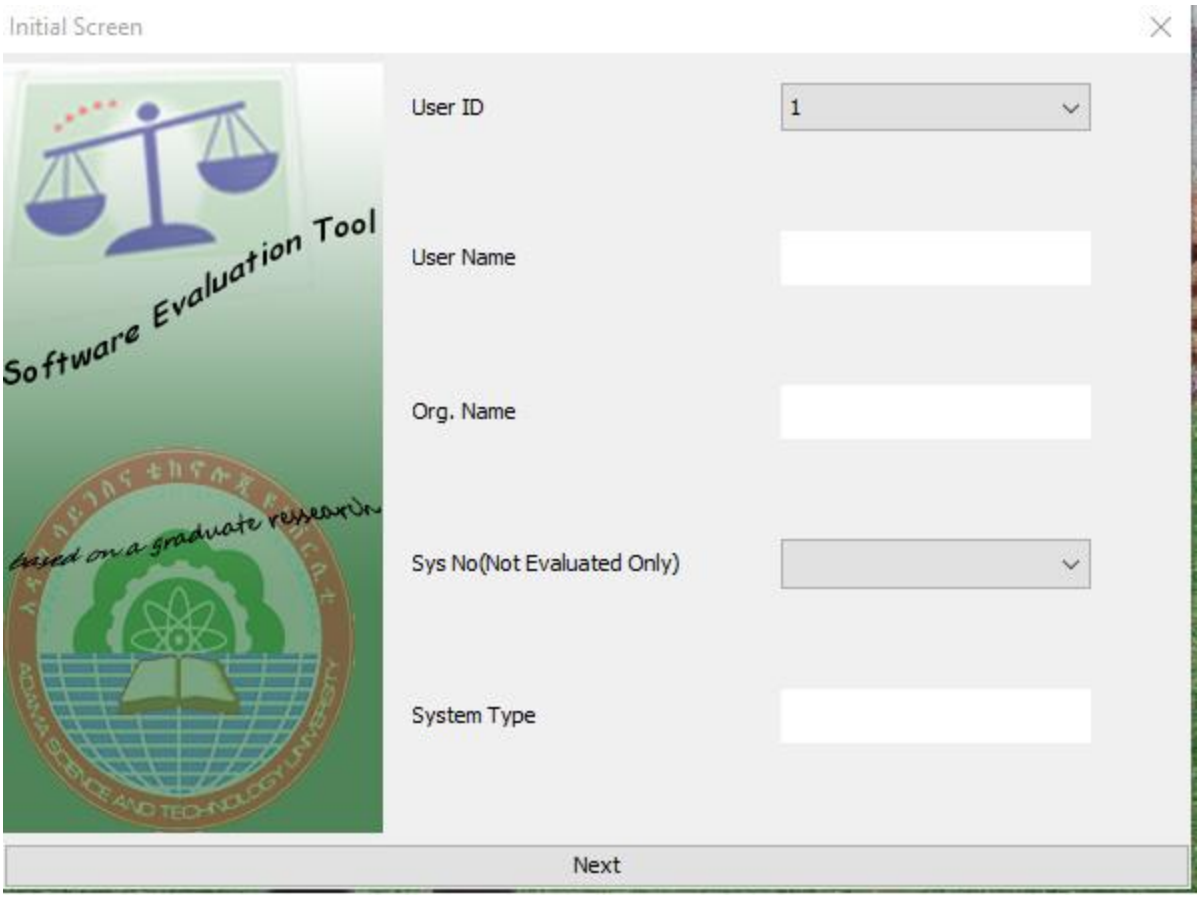

Fig 4-3: Initial screen

4.5.1 Criteria selection dialog

As described in above section; evaluation criteria attributes are grouped into their appropriate categories. The designed application is also categorize the criteria based on their group. The following screen shows us functional criteria of the software system to be evaluated with some default value.

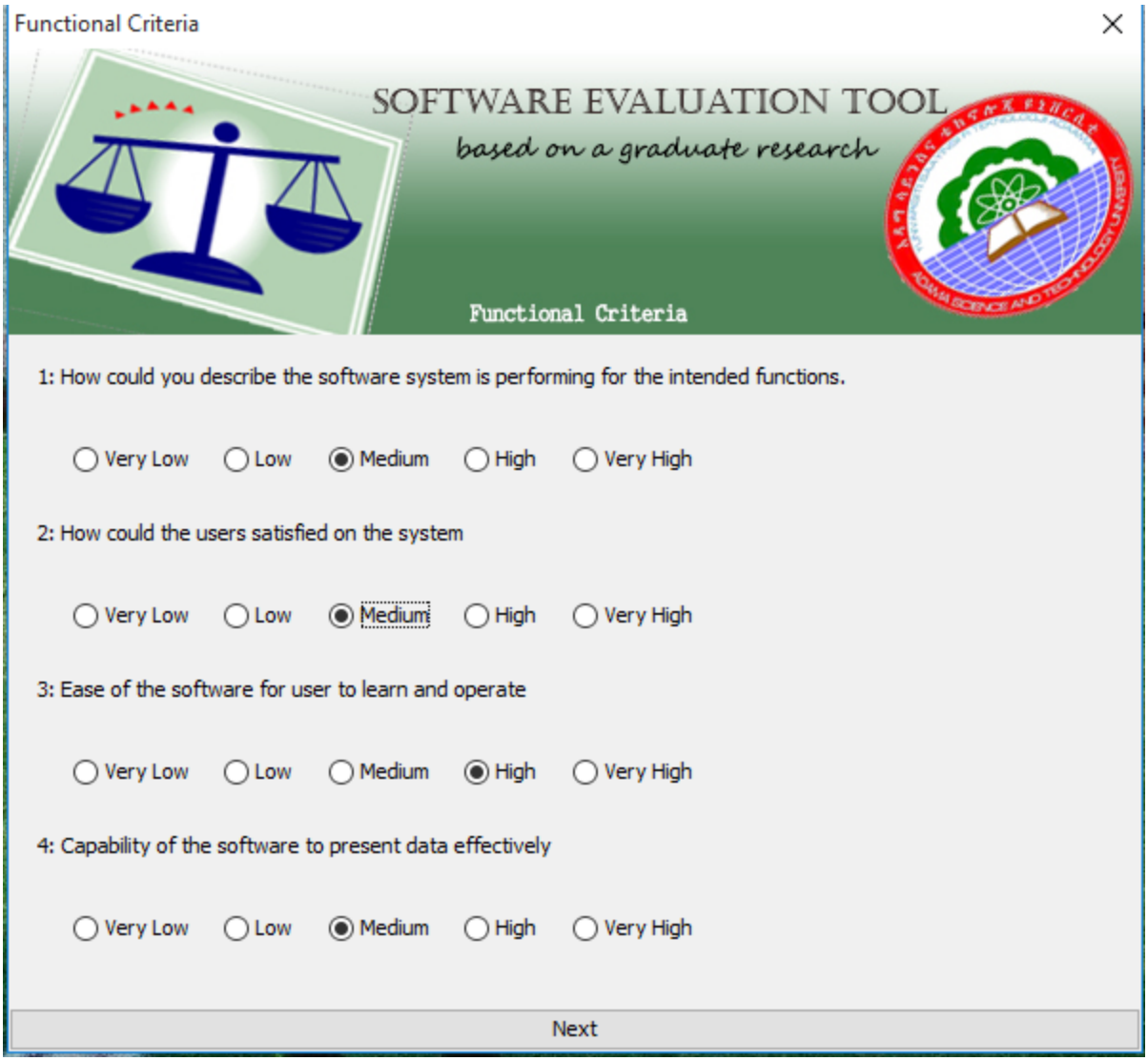

Fig 4-4: Functional criteria dialog

The following screen shows us Reliability criteria of the software system to be evaluated.

Fig 4-5: Reliability criteria dialog

4.5.2 Result dialog

The result screen dialog shows us the detail of the evaluation process including evaluation criteria, the value selected by the user and normalized score each criteria. The dialog display the mismatch indication too; means whether the criteria selected fulfill the expected result or not.

Fig 4-6: Final result dialog

4.5.3 Aggregate value and mismatch

The screen displays the aggregated value of the evaluated software system and the criteria that made a mismatch.

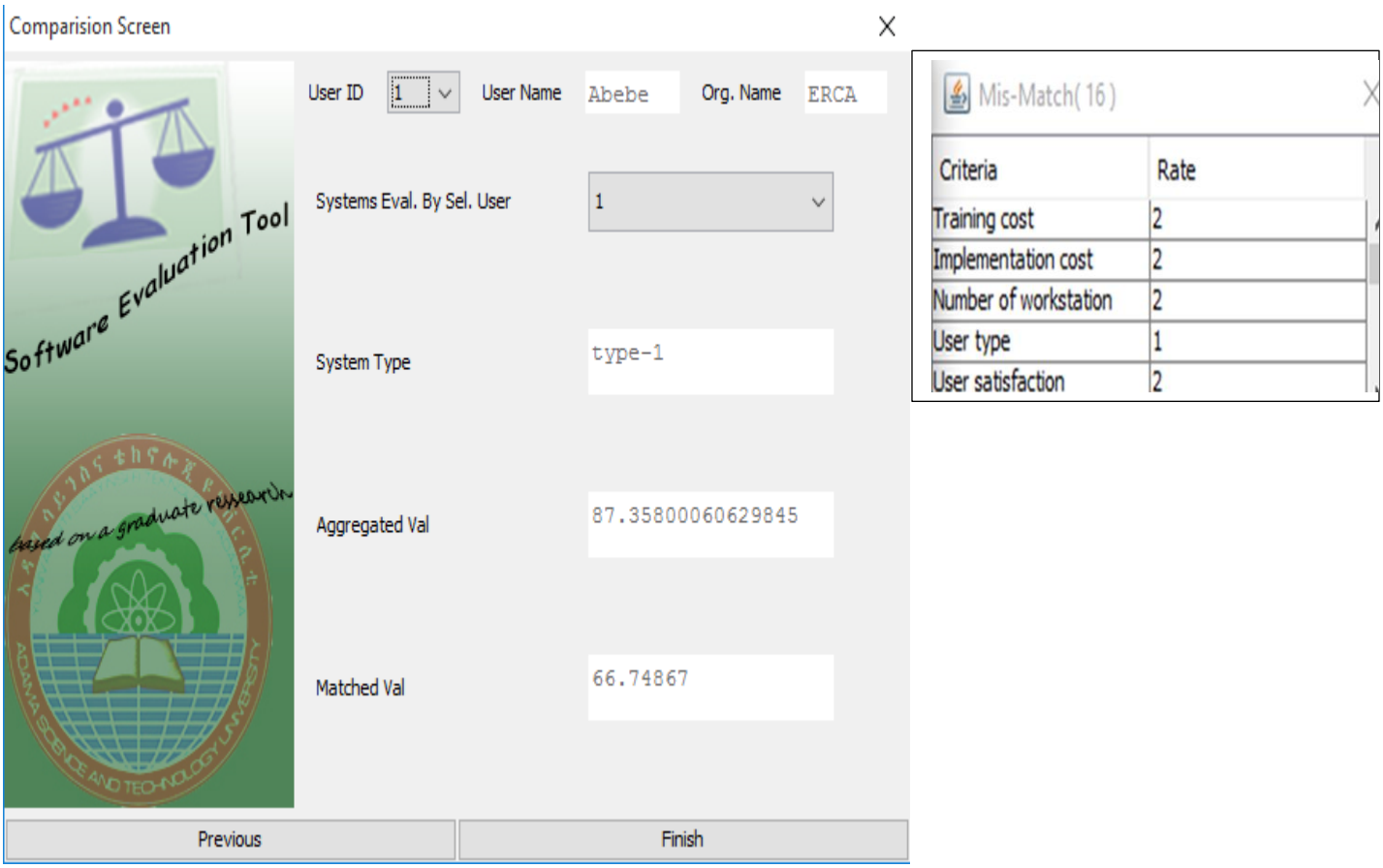

Fig 4-7: Mismatch criteria

CHAPTER FIVE
EVALUATION AND INTERPRETATION

5.1 Overview

After implementation is completed, the system has to be evaluated against the requirements to make sure that it satisfies what was expected from it. During any formal decision-making process, an investigation should be done in different way. For this study also an appropriate software selection for organization task had been done to evaluate the proposed model.

5.2 Testing performance of the system

The model had to be evaluated against the requirements to make sure that whether it satisfies what was expected from it using one case study. In addition, user acceptance testing was performed to make sure that the end users are satisfied and willing to accept the system. Finally, the testing results were analyzed and interpreted. Testing process of the system was applied by selecting organization which needs to procure a software system.

5.2.1 Case study for testing the system

Ethiopian Revenues and Customs Authority (ERCA) need to select the appropriate ERP system. Ethiopian Revenues and Customs Authority (ERCA) is government organization in the body responsible for collecting revenue from customs duties and domestic taxes. The Ethiopian Revenues and Customs Authority are responsible to protect the society from adverse effects of smuggling. It takes legal action on the people and vehicles involved in the act of smuggling while it facilitates the legitimate movement of goods and people across the border. The organization need to select or procure ERP system.

The organization wants to procure ERP system which integrates different system operation inside for the organization that can reduce the time consumed in manually process and the payment process in the organization.

The organization needs the software system to have the basic functionality like payroll system and produce reports. The system expected to have more of the functional and nonfunctional requirements for just to make easy life for the user. In addition to the basic functional and nonfunctional criteria there need to be a requirement of the organization provided for further evaluation process. The organization need to know how the software system can solve the problem of the organization based on their requirements. The organization provides a consulting service that can help in selection of ERP software system to improve the business work. Finally the organizations found vendor that can provide the system and try to evaluate based on the requirement provided.

Base on the provided questionnaire for evaluation; the decision maker rate the software system according to its feature and characteristics stated in table 5-1. User rate is the value given or selected by the user about the software system they are evaluating. The organization need to select among two software system that can be best for their organization in general after looking the final output of the system.

Table 5-1: User rate

Criteria	Rating(1)	Rating (2)
How could you describe the software system is performing for the intended functions.	High (3)	Medium (2)
How could the users satisfied on the system	Very High (4)	Very High (4)
Ease of the software for user to learn and operate	Medium (2)	Very High (4)
Capability of the software to present data effectively	High (3)	High (3)
Error reporting and messaging ability of the software system	Very high (4)	Medium (2)
Capability of the software system to operate consistently without crashing.	High (3)	Medium (2)
Capability of the software system to support backup and recovery	High (3)	Low (1)
Capability of the software system to produce	Medium (2)	Low (1)

result in reasonable amount of time related to data size.		
The software system to support for managing and enforcing access right.	Medium (2)	Medium (2)
Cost of additional requirement to train the software system.	High (3)	Low (1)
The ability of the system to run in different computer platform	Medium (2)	Low (1)
How do you rate the correctness of the output of the system	Very High (4)	High (3)
The maintenance cost of the system	High (3)	Very High (4)
Ease of the interface of the system	High (3)	High (3)
The capacity of the system to handle the task	Very High (4)	Medium (2)
Ease of Installability of the system	High (3)	High (3)
Compatibility of the system with other hardware.	High (3)	Medium (2)
The capability of the system to upgrade and customize	High (3)	Low (1)
The cost for implementing the system	High (3)	Medium (2)
Number of workstation or computers the system support during process	High (3)	Very High (4)
The financial status of the vendor	Very High (4)	Medium (2)
The tutorial provided for the system	High (3)	Medium (2)
Technical capability of the vendor	High (3)	Medium (2)
Business skill of the vendor in system development	High (3)	High (3)
Past experience of the vendor in software development	High (3)	High (3)
Installation cost of the system	High (3)	Medium (2)

Table 5-2: Compute aggregate value (1)

Criteria	Weight	Rate	Score
Service satisfaction	15%	3	45
User satisfaction	3.3%	4	13.2
Access Control	8.6%	2	17.2
Interoperability	5.12%	3	15.36
Error prone	2%	4	8
Backup	2%	3	6
Recovery	6%	3	18
Correctness	9%	2	18
Robustness	4%	2	8
Throughput	3%	3	9
Capacity	2.4%	2	4.8
Upgradability	9.6%	4	38.4
Number of workstation	7%	3	21
User type	4%	3	12
User interface	2%	4	8
Data visualization	5%	3	15
Compatibility	4%	3	12
Install ability	2%	3	6
Financial status	0.3%	3	0.9
Past experience	1%	3	3
Business skill	0.5%	4	2
Tutorial	1.2%	3	3.6
Training cost	0.5%	3	1.5
Installation cost	1.44%	3	4.32
Implementation cost	1.06%	3	3.18
	Score		293.46

Table 5-3: Compute aggregate value (1)

Criteria	Weight	Rate	Score
Service satisfaction	15%	2	30
User satisfaction	3.3%	4	13.2
Access Control	8.6%	4	34.4
Interoperability	5.12%	3	15.36
Error prone	2%	2	4
Backup	2%	2	4
Recovery	6%	1	6
Correctness	9%	1	9
Robustness	4%	2	8
Throughput	3%	1	3
Capacity	2.4%	1	2.4
Upgradability	9.6%	3	28.8
Number of workstation	7%	4	28
User type	4%	3	12
User interface	2%	2	4
Data visualization	5%	3	15
Compatibility	4%	2	8
Install ability	2%	1	2
Financial status	0.3%	2	0.6
Past experience	1%	4	4
Business skill	0.5%	2	1
Tutorial	1.2%	2	2.4
Training cost	0.5%	2	1
Installation cost	1.44%	3	4.32
Implementation cost	1.06%	3	3.18
	Score		243.66

The aggregate value is calculated by multiplying the weight of each criterion with the rate given by the user and come with final score value. In the selection process the aggregate result shows that software system 1 is better than software system 2. The evaluation result (ranking) of software system obtained using new weighted scoring method and ranking the obtained values. The final result of the processed data is displayed and the user can select the software system with greater aggregate value; because it means the software system can meet the requirement of the organization and is selected as best software as described in Table 5-4.

Matching software system

Table 5-4: Matching software system

Software system	Aggregate value	Matching (%)
System 1	293.46	73%
System 2	243.66	60%

Based on the scaling method stated above recommended selection of suitable software system can be summarized as follows. So that the decision maker can decide based on the table 5-5.

Table 5-5: Recommended values for decision

Score value	**0**	**1**	**2**	**3**	**4**
Scaling	Very low	Low	Medium	High	Very high
Aggregate value	0	100	200	300	400
Matching	0	~25%	~50%	~75%	~100%

Finally the tested software system fall under '*high'* scored value which means more of the criteria fit the requirement of the organization; because the software system score 293.46 aggregate value which approaches to 300 (high) score. The second software system score 246.66 aggregate which is approach to medium score values; so that the decision maker can decide the first software system is selected as an appropriate system.

Table 5-6: Software system matching

Software system	Aggregate value	Scaling
System 1	293.46	3 (high)
System 2	246.66	2 (medium)

Mismatch identification

Table 5-7 indicated which criteria made a mismatch with respect to the requirement stated by the organization. As tried to describe on the way of identifying the mismatch between the software system and organization requirement, the criteria was identified under a mismatch for the evaluation process if it can score below 3 or if the indicated software characteristics score a '*medium* 'or below that; because the software system under evaluation is expected to score '*high'* or '*very high*' for all criteria as. Therefore; having the above table with the user rate or the value selected by the user, we can conclude that there are mismatches on the alternative software system.

Table 5-7: Mismatched criteria (1)

Criteria	Score value
Access Control	Medium (2)
Correctness	Medium (2)
Robustness	Medium (2)
Capacity	Medium (2)

Table 5-7: Mismatch for software system (2)

Criteria	Score value
Service satisfaction	Medium (2)
Error prone	Medium (2)
Backup	Medium (2)
Recovery	Low (1)
Correctness	Low (1)

Robustness	Medium (2)
Throughput	Low (1)
Capacity	Low (1)
User interface	Medium (2)
Compatibility	Medium (2)
Install ability	Low (1)
Financial status	Medium (2)
Business skill	Medium (2)
Tutorial	Medium (2)
Training cost	Medium (2)

As we can see from table 5-5, 1 has got only four mismatches from the expected functionality or feature of the system. Therefore the decision maker can decide whether the unmatched criteria are more necessary for the organization or not and decide on those criteria. Table 5-6 showed us the second software system unmatched criteria, this system has got much in number than the first one; it is almost 15.

The same procedure with the first one; the decision maker can decide on very necessity of unmatched criteria with the organization requirements.

5.3 Functional testing

The functional testing part of a testing methodology is typically broken down into different components such as unit testing, system testing and.

5.3.1 Unit testing

The Unit testing part of a testing methodology is the testing of individual software modules or components that make up an application or system. These tests are usually written by the developers of the module and in a test-driven-development methodology.

```
public class FunctionalCriteria extends JDialog {
	private static final long serialVersionUID = 1L;
	JLabel image;
	DBConnector db;
	ArrayList<Evaluation> evalData;
	Evaluation f1,f2,f3,f4;

	ArrayList<String> functional=new ArrayList<String>();
	ArrayList<String> optionsList=new ArrayList<String>();

	JLabel questionf1,questionf2,questionf3,questionf4;
	JRadioButton f1op1,f1op2,f1op3,f1op4,f1op5;
	JRadioButton f2op1,f2op2,f2op3,f2op4,f2op5;
	JRadioButton f3op1,f3op2,f3op3,f3op4,f3op5;
	JRadioButton f4op1,f4op2,f4op3,f4op4,f4op5;
	ButtonGroup
questionc1group,questionc2group,questionc3group,questionc4group;

	 public ArrayList<Evaluation> getEvalData() {
			return evalData;
		}

	public void fillQuestions()
	{
		functional.add("1: How could you describe the software system is
performing for the intended functions.");
		functional.add("2: How could the users satisfied on the system");
		functional.add("3: Ease of the software for user to learn and
operate ");
		functional.add("4: Capability of the software to present data
effectively");

		optionsList.add("Very low");
		optionsList.add("Low");
		optionsList.add("Medium");
		optionsList.add("High");
		optionsList.add("Very High");

	}
```

5.3. 2 System testing

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system (that have been previously unit tested and integration tested), and then testing it as a whole. This testing is listed under the black-box testing method, where the software is checked for user-expected working conditions as well as potential exception and edge conditions. The user is expected to select values and the system store and test the values for evaluation.

When the prototype system is run; the system is expected to display the initial dialog first with. The user needs to click on next to start evaluation process. After the test had been done figure 5-1 is displayed and confirmed success message.

The criteria dialog should display after the initial dialog is displayed. Then after, the user expected to select one of the provided radio button choice values which can represent the real characteristics of the evaluating system. Figure 5-1 shows functional criteria dialog.

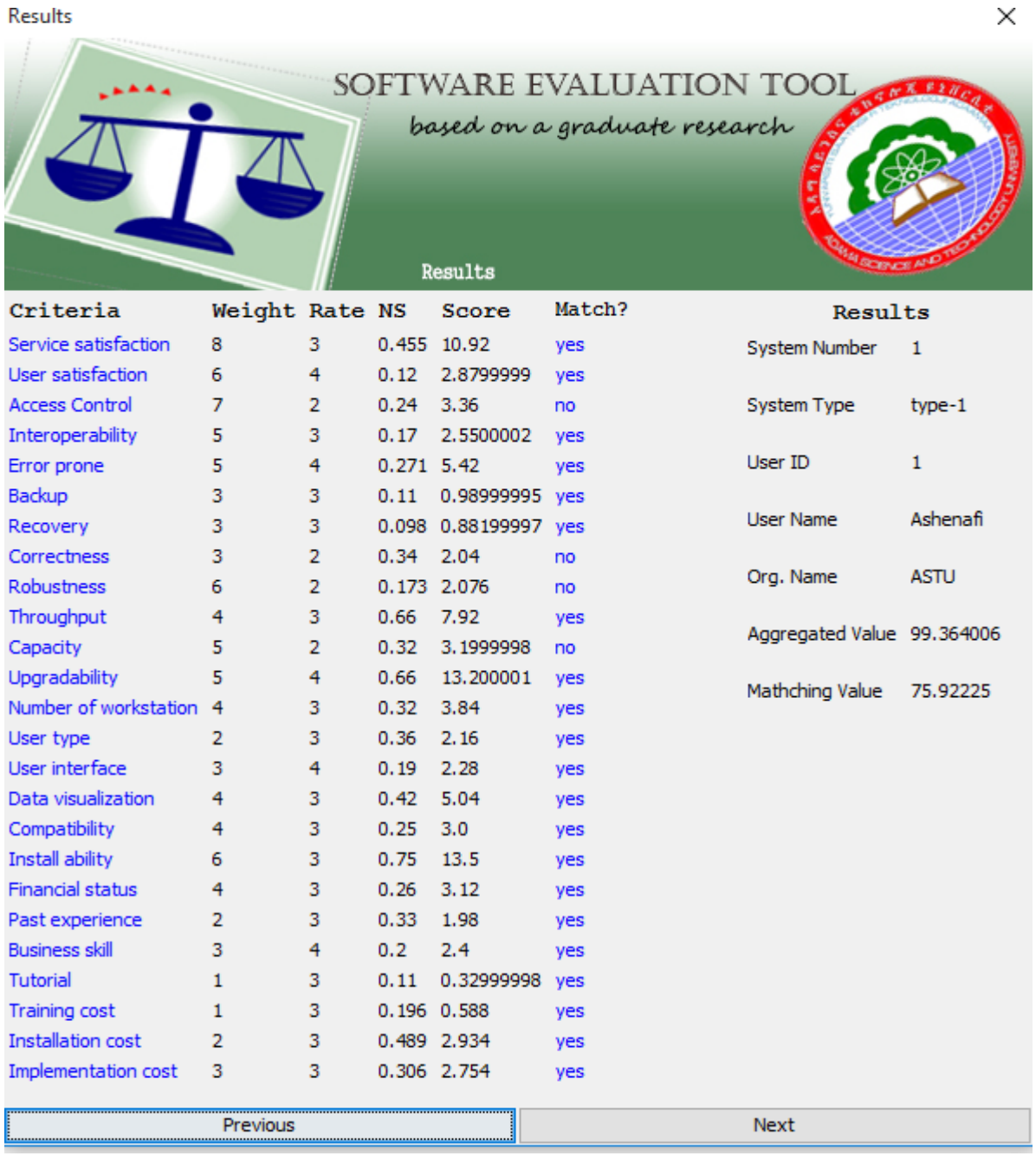

Fig 5-1: Testing result dialog

CHAPTER SIX
FINDINGS OF THE STUDY

6.1 Overview

The purpose of this research was to explore the issue encountered in software evaluation in software selection process. This chapter discussed mainly the research findings. The findings relate to the research questions has guided the study. Data are obtained from different organization who was involved in software operation activities and literatures used to obtain information about software system quality and selection process were incorporated in the study.

6.2 Findings of the study

The evaluation model has requirements elicitation and the software system selection process including the criteria attribute identified. Thus, the process and attribute criteria were used to evaluate business software systems are the major findings to the study. The organizations indicated that software products in the marketplace must drive the requirements definition. This suggested that while it is important to initially define the high requirements, to realize the benefits of software procurement process must be in place that defines requirements according to what is available in the marketplace.

The importance of the Analytical Hierarchical Processing (AHP) and outranking were brought to this study for evaluation techniques. Resulting score of AHP indicated relative ranking of the alternatives. On this study, not only ranking of the alternatives but also how each alternative meet user requirements of that package were discussed using the mismatch concept. The aggregate score of each software alternative may not be consistent even though requirements are same because aggregate score depends on decision makers own judgment which may not remain consistent for all the time.

The ISO software quality characteristics including functional criteria and nonfunctional criteria such as reliability criteria, usability criteria of the system, efficiency criteria, maintainability and portability criteria of the system were used as evaluation criteria in this study for the evaluation process.

There are also additional criteria identified through organizational requirement such as vendors and cost criteria; and all this criteria were identified into different groups according to their characteristics. Among the standards related to software quality criteria related to functional requirement of the system and nonfunctional features of the system or features and cost and benefits of the software packages were also commonly used for further evaluation process.

The Analytical hierarchical processing (AHP) evaluation technique is based on a hierarchical framework of criteria [42]. As we can see in the discussion of the technique; the upper level deals with the goal of the selection process. The next or the second level explored the major subgroup of the main criteria identified where it was grouped into lower levels of hierarchy. The last level contained the software alternatives to be evaluated and select the process. Priorities for the criteria under each group were established based on their necessity for evaluation process and the level were assigned as shown in the discussion part. Finally, pair-wise comparison between the factors at each level was applied for evaluation process using the technique. If N criteria attributes were being compared then N (N_1)/2 pair-wise comparisons are made for each and every main and subgroup criteria. The comparisons were the basis for calculation of the relative weight of each factor at each level and to find normalized number. The last step is computing the relative score of each alternative with respect to the decision making goal and indicate the best evaluation system at last.

The designed model was incorporated components that used to indicate the best software system for decision maker. The first step to start this research was to find out different studies regarding the aim of the research such as theoretical study on software evaluation and the way of making the business environment very satisfied for that using software system in their running their business. Requirement elicitation was taken as the best contribution for the evaluation to find out the problem in evaluation and selection of software system and the problem of the current system to identify the best criteria used as attribute for evaluation processes. The data collected were analyzed in qualitative and quantitative empirical study form and the major criteria for evaluation were identified through requirement elicitation and literature review and grouped as of their respective categories. The analyzed data and identified criteria attribute are processed in selected evaluation technique of AHP. Finally, the selection process was made after the completion of the evaluation process.

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

7.1 Conclusion

Selecting the best software system is very critical for the success of organizations, and that is depends on the evaluation method that the organization used to select the desired software system [42]. The technical support, system admin, stakeholders, and management have subjective opinions on the requirements, performance and capabilities of the software system [43]. These opinions of individuals were collected and aggregated into a different composite value for evaluation process so that this would have a valuable contribution to choose the right software system for organizations. The purpose of this thesis was to use its results as a good step for possible future evaluations.

The study investigated common problems and limitations of the previous studies on the evaluation and selection software system. It also classified the main criteria and process that are required for evaluating and selecting software. Methodology for developing the software evaluation model was also presented in this paper. The Finding of this thesis was initially aimed to find a process and propose a new model for evaluating and selecting software system where it will be developed to support decision maker in software selection process. Issues such as identifying software evaluation criteria, establishing hierarchy between criteria and defining their common measures and application of sound decision aid methodology rose frequently and made the evaluation and selection process boring.

Evaluation process is the major task need to be accomplished for selecting the appropriate software system. Even though there is process supported by different evaluation method more of them had been thought as not compatible for this study and needed to find a process by deriving best feature of the processes supported by other techniques. In preparation phase evaluation criteria need to be identified and provided for evaluation process and determine kind of software system which is going to be evaluating. In evaluation phase evaluation technique for evaluation process and the way of achieved to the objective are identified and used. In selection phase the decision maker need to decide on performance of the evaluating system based on the information delivered from the system; such as aggregate value and mismatched criteria.

There are several evaluation methods that support decision making process in software selection. AHP is selected as best method among those alternatives because it has advantage over the rest of the techniques for evaluation process. Mismatch is unequal need between the software systems provided with the user requirement. Mismatch is identified by collecting the criteria attributes for decision making process. The decision maker decides based on the necessity of the mismatch criteria for the requirement of the organization.

Criteria are identified as they are used to evaluate for business software system. More of the evaluation criteria are derived from ISO 9126 quality characteristics by changing their type and content to make it fit for the study. Besides to the quality characteristics requirement of are collected from different organization for deriving criteria and way to evaluate software system. The evaluation criteria are selected as best for any business software needed to have.

The model is designed in the way of process to evaluate and select the best software system. It have different component including a kind of criteria and process used to evaluate the software system. Prototype is designed by java programming for demonstration purpose and tested using case study that need to select a business software system from alternative.

The result achieved from this study is the procedure needed to evaluate a software system; especially systematic way to select a system. Most appropriate criteria are selected from different sources and evaluation method is also selected from alternative provided considering their characteristics and usefulness to the study.

Different aspects related to the software selection such as methodology and the way of developing software evaluation model and issues that needs to be taken into consideration during the process of software selection was critically discussed at each stage.

7.2 Future work

More evaluation model in order both to enhance the knowledge base of evaluation and to obtain more feedback on the benefit of the adoption of the knowledge based situation are needed. The knowledge gathered during the organizational study could be useful in such an evaluation. From an academic perspective, future work based upon this thesis could well entail further development of a software evaluation for every kind of software system with different new techniques and methods. In developing this evaluation model, it would be interesting to expand it so that it can be applied to other kinds of evaluation projects. Widening its applicability and use to other contexts and domains is also an essential part of this study.

REFERENCES

[1] Orci I, "*Kritiska IT system*". ISSN : 1101-8526, 1995

[2] Jon Arvid Børretzen "*Software Fault Reporting Processes in Business-Critical Systems*".

[3] Chris Vandersluis "*The chanllenges of selecting Enterprise software*".

[4] Galorath,D.(2004) Software Reuse and Commercial Off-the-Shelf Software. Pp6-10

[5] T. Neubauer and C. Stummer, "Interactive Decision Support for Multi Objective COTS Selection," in System Sciences,HICSS 2007 40th Annual Hawaii International Conference on, pp. 283b-283b, 2007.

[6] C. Couts and P. Gerdes, "Integrating COTS Software: Lessons from a Large Healthcare organization," IT Professional, vol. 12, pp. 50- 58, 2010.

[7] D. Kunda, "STACE: Social Technical Approach to COTS Software Evaluation," Lecture Notes in

[8] Anil S.Jadhava, Rajendra M.Sonar, Evaluating and selecting software packages : A review

[9] Basili, V. R. & Weiss, M. D. 1984. A Methodology for Collecting Valid Software Engineering Data. In IEEE Transactions On Software Engineering, 6, pp. 728-738.

[10] Wrenn, B., Stevens, R., & Loudon, D. (2002). Marketing Research: Text and Cases. NY: Best Business Books.

[11] Anil S.Jadhava, Rajendra M.Sonar, Evaluating and selecting software packages : A review

[12] Polit D.F. and Hungler B.P. (1999) Nursing Research: Principles and Methods. 6th ed. Philadelphia. Lippincott

[13] J. Kontio, "OTSO: A Systematic Process for Reusable Software Component Selection," University of Maryland at College Park, College Park, MD, 1995.

[14] A. Mohamed, G. Ruhe, A. Eberlein. "COTS Selection: Past, Present, and Future". In Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07), Tucson, Arizona (USA), 2007

[15] Vlahavas, I., Stemalos, I., Refanidis, I., & Tsoukiàs, A. 1999. ESSE: an expert system for software evaluation. Knowledge-Based Systems , 12, pp. 183-197.

[16] https://mthoyibi.files.wordpress.com/.../action-research_patrick-j-m-coste.

[18] Collins J. Carlton, "How to select the right accounting software", Journal of Accountancy; New York; Vol. 188, Issue 2, Aug 1999; PP. 61-69

[19] Swartz, D., & Orgill, K. (2001). Higher education ERP: Lessons learned. *Educause Quarterly*, *24*(2), 20-27.

[20] Fletcher, L. A. (2001). Going beyond the buzzword: what exactly is CRM?.*Learned Publishing*, *14*(3), 213-222.

[21] Ho W. 2008. Integrated analytic hierarchy process and its applications – A literature review. European Journal of Operational Research, 186, pp. 211–228.

[22] Comella-Dorda, S., Dean, J., Lewis, G., Morris, E., Oberndorf, P., & Harper, E. 2004. A Process for COTS Software Product Evaluation. Software Engineering Institute, Carnegie Mellon University. Technical report CMU/SEI-2003-TR-017. Encyclopædia Britannica 2009 Student and Home Edition.

[23] Vaidya O. S. & Kumar S. 2006. Analytic hierarchy process: An overview of applications. European Journal of Operational Research, 169, pp. 1–29

[24] Hansen, S., Dorling, A., Schweigert, T. and Lovett, J. ”IT Purchasing Guidebook for small enterprises. A selfhelp guide to higher value from IT purchases”, QAI Europe, UK, 1999.

[25] Carney, D. Requirements and COTS-Based Systems: A Thorny Question Indeed. SEI Interactive, Carnegie Mellon University, June 1999.

[26] Nils Rosen. “Evaluation methods for procurement of business critical software systems”. 2006

[27] Elena Sanchez Sanchez. “Procurement usability and work friendly software”. 2011

[29] Gediga, G., Hamborg, K.-C. & Willumeit, H. (1997). The IsoMetrics manual (Version 1.15). Osnabrück: Universität Onabrück: Fachbereich Psychologie. http://www.evalinstitut. de/isometrics/.

[30] Basili V. R. 1992. Software Modeling and Measurement: The Goal Question Metric Paradigm. University of Maryland, College Park, MD. Computer Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96).

[32] Land R., Blankers L., "Classifying and Consolidating Software Component Selection Methods," MRTC report ISSN 1404-3041 ISRN MDH-MRTC-218/2007-1-SE, 2007.

[33] S. Comella-Dorda, J. C. Dean, G. Lewis, E. Morris, P. Oberndorf, and E. Harper, "A Process for COTS Software Product Evaluation," CMU Univ., SEI Institute, CMU/SEI-2003-TR-017, July 2004.

[34] R. Land, L. Blankers, M. Chaudron, and I. Crnkovic, "COTS Selection Best Practices in Literature and in Industry," in Proceedings of 10th International Conference on software Reuse, Beijing, China, pp. 100-111, 2008.

[35] Saaty, T.L. 1980. The Analytic Hierarchy Process. McGraw-Hill, New York.

[36] Mollaghasemi M. and Pet- Edwards J., “Technical briefing: making multiple-objective

decisions", IEEE computer society press, Los Alamitos, California, 1997.

[37] A. Mohamed, G. Ruhe, A. Eberlein. "COTS Selection: Past, Present, and Future". In Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07), Tucson, Arizona (USA), 2007.

[38] R. Land, L. Blankers, M. Chaudron, and I. Crnkovic, "COTS Selection Best Practices in Literature and in Industry," in Proceedings of 10th International Conference on software Reuse, Beijing, China, pp. 100-111, 2008.

[39] D. Carney, S. A. Hissam, and D. Plakosh, "Complex COTS-Based Software Systems: Practical Steps for Their Maintenance," Journal of Software Maintenance, vol. 12 (6), 2000 pp. 357-376.

[40] A. Kvale, J. Li, and, R. Conradi, "A Case Study on Building COTS-Based System Using Aspect-Oriented Programming," in Proceedings of the 2005 ACM symposium on Applied computing, Santa Fe, New Mexico, pp. 1491 – 1498, 2005.

[41] T. Neubauer and C. Stummer, "Interactive Decision Support for Multi Objective COTS Selection," in System Sciences,HICSS 2007 40th Annual Hawaii International Conference on, pp. 283b-283b, 2007.

Appendix A: AHP Evaluation Process

Table A1: Pairwise comparison judgment between sub criteria attributes

Criteria	Sub criteria	Pairwise comparison
Functionality	• Service satisfaction • User satisfaction • Access control • Interoperability	Service satisfaction is 4 times as important than user satisfaction Access control is 3 times as important than user satisfaction Interoperability is 2 times as important than user satisfaction Service satisfaction is 2 times as important than access control Service satisfaction is 3 times as important than interoperability Access control is 2 times as important than interoperability
Reliability	• Backup • Recovery • Error reporting • Correctness • Robustness	Backup and recovery has equal important Error reporting is 2 times as important than backup Correctness is 4 times as important than backup Robustness is 3 times as important than backup Error reporting is 2 times as important than recovery Correctness is 4 times as important than recovery Robustness is 3 times as important than recovery Correctness is 2 times as important than error reporting Error reporting is 3 times as important than robustness Correctness is 2 times as important than robustness
Usability	• User type • User Interface • Data Visualization	User type is 2 times as important than user interface Data visualization is 4 times as important than user type Data visualization is 3 times as important than user interface
Efficiency	• Performance • Throughput	Performance is 2 times as important than throughput
Portability	• Compatibility • Installability	Installability is 2 times as important than compatibility
Maintainability	• Upgradability • Number of workstation	Upgradability is 2 times as important than number of workstation support
		Past experience is 4 times as important than financial status

Vendor	• Financial status • Past experience • Business skill • Provide tutorial	Business skill is 2 times as important than financial status Provide tutorial is 3 times as important than financial status Past experience is 3 times as important than business skill Provide tutorial is 2 times as important than past experience Provide tutorial is 3 times as important than business skill
Cost	• Training cost • Installation cost • Maintenance cost	Installation cost is 2 times as important than training cost Maintenance cost is 3 times as important than training cost Installation cost is 2 times as important than Maintenance cost

Table A2: Functionality sub criteria matrix

	User satisfaction	Service satisfaction	Access control	Interoperability
User satisfaction	1/10	3/25	2/23	1/13
Service satisfaction	4/10	12/25	12/23	6/13
Access control	3/10	6/25	6/23	4/13
Interoperability	1/5	4/25	3/23	2/13

Table A3: Reliability sub criteria matrix

	Backup	Recovery	Error reporting	Correctness	Robustness
Backup	1/11	1/11	3/26	1/10	1/20
Recovery	1/11	1/11	3/26	1/10	1/20
Error reporting	2/11	2/11	3/13	1/5	9/20
Correctness	4/11	4/11	6/13	2/5	6/20
Robustness	3/11	3/11	1/13	1/5	3/20

Table A4: Usability sub criteria matrix

	User type	User interface	Data visualization
User type	2/11	1/3	3/19
User interface	1/11	1/6	4/9
Data visualization	8/11	1/2	12/19

Table A5: Efficiency sub criteria matrix

	Performance	Throughput
Performance e	2/3	2/3
Throughput	1/3	1/3

Table A6: Portability sub criteria matrix

	Compatibility	Installability
Compatibility	1/3	1/3
Installability	2/3	2/3

Table A7: Maintainability sub criteria matrix

	Upgradability	Number of workstation
Upgradability	2/3	2/3
Number of workstation	1/3	1/3

Table A8: Vendor sub criteria matrix

	Financial status	Past experience	Business skill	Provide tutorial
Financial status	1/10	3/43	1/15	2/13
Past experience	4/10	12/43	6/15	3/13
Business skill	1/5	4/43	2/15	2/13
Provide tutorial	3/10	24/43	6/15	6/13

Table A9: Cost sub criteria matrix

	Training cost	Installation cost	Maintenance cost
Training cost	1/6	1/4	1/10
Installation cost	1/3	1/2	6/10
Maintenance cost	1/2	1/4	3/10

Appendix B: Questionnaire of the study

1. **What is your current position in the organization**

   O Procurement manager O ICT officer
   O System user O Senior management
   O Other, specify here_____

2. **How many years of working-experience related to software intensive systems do you have?**

   O Less than 1 year O 1-2 years O 2-4 years
   O 4-6 years O 6-10 years O More than 10 years

3. **How could you describe the interface of the software system you are using?**

   O Extremely user friendly O Very user friendly
   O Moderately user friendly O Slightly user friendly
   O Not at all user friendly

4. **How successful is the software system you are using is in performing its intended task?**

   O Extremely successful O Very successful
   O Moderately successful O Slightly successful
   O Not at all successful

   Comment: *(What problem does the system have? What do you suggest for the system to have in addition to the functions it is performing?)*

5. **How could you rate the result or effect of the system functions?**

   O Extremely successful O Very successful
   O Moderately successful O Slightly successful
   O Not at all successful

6. **What kind of mechanism the system is using to prevent unauthorized access of data**

   O Software system to protect unauthorized access of data
   O Hardware to protect unauthorized access of data
   O There is no mechanism to protect unauthorized access of data
   O Any other mechanism please specify _____

Comment:*(What mechanism do you recommend to use in data protection)*

7. **How can the system prevent /avoid/ failure caused by fault in the software itself?**

   O The system by itself can handle and fix the error caused by software itself
   O The system report the user where the error occur to be fixed manually
   O Any other way please specify _____

8. **What is the way to maintain a certain level of performance if software fault occur?**

   O There is a software system to correct error occurred.
   O Manual system to maintain the error occurred
   O Any other way to maintain _____

9. **Is the system in a possession to recover potentially lost data?**

   O Yes it can successfully recover the lost data
   O No it cannot recover if data is lost

   Comment: *(What do you recommend for a commercial system to protect data from accidentally damaged **AND** the way to recover data if is lost)*

10. **Is the system to be adopted Or understood by user**

    O Yes it is easy to understand the system
    O It makes a little confusion
    O It is difficult to understand

11. **Is the system enable the user to control and use the software**

    O Yes the user can control and use on the system easily
    O Difficulty may face in the middle though it is easy.
    O No, user cannot control everything by himself

12. **How can you describe the system functioning in timely manner; response time, processing time and throughput rate.**

    O It is extremely fast
    O It is moderately fast
    O It is slow in responding and processing data.
    O Very slow

    Comment: *(What is your task on the system? How do you describe the performance of the system related to the task you have on?)*

13. **Is the system efficient/ using the amount of resource needed as well as duration/.**

    O Yes it is efficient
    O It is Moderately efficient
    O Not efficient

    Comment: *(Write any comment on the efficiency of the system)*

14. **Can the system identify the part that analyzes for potential cause of failures?**

    O Yes it can identify the cause of failure.
    O No, there is no way to know where the fault is occurred; only admin knows.
    O Any other way please specify_____

15. **Is the system in a way to support the implementation to be modified?**

O Yes it can be modified
O Support partial modification
O Not at all

Overall opinion about the system

16. **Overall, how do you rate the quality of the software system you are using?**

O Very good
O Good
O OK
O Poor
O Very Bad

17. **What do you recommend for commercial software systems should have as functionality?**

	Excellent	Good	OK	Poor	Unacceptable	N/A
Training of the system	☐	☐	☐	☐	☐	☐
Installation process	☐	☐	☐	☐	☐	☐
Data entry interface	☐	☐	☐	☐	☐	☐
Reports producing	☐	☐	☐	☐	☐	☐
Quality of the system	☐	☐	☐	☐	☐	☐
Ease of use	☐	☐	☐	☐	☐	☐
Usefulness	☐	☐	☐	☐	☐	☐
Support to user	☐	☐	☐	☐	☐	☐
Meets requirements	☐	☐	☐	☐	☐	☐
Test fault as modification	☐	☐	☐	☐	☐	☐
Adopt another platform	☐	☐	☐	☐	☐	☐
Replace another system	☐	☐	☐	☐	☐	☐

Appendix C: Source code of the prototype

Source code for functional criteria

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.swing.BorderFactory;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.SpringLayout;
import javax.swing.UIManager;

public class FunctionalCriteria extends JDialog {
      private static final long serialVersionUID = 1L;
      JLabel image;
      DBConnector db;
      ArrayList<Evaluation> evalData;
      Evaluation f1,f2,f3,f4;

      ArrayList<String> functional=new ArrayList<String>();
      ArrayList<String> optionsList=new ArrayList<String>();

      JLabel questionf1,questionf2,questionf3,questionf4;
      JRadioButton f1op1,f1op2,f1op3,f1op4,f1op5;
      JRadioButton f2op1,f2op2,f2op3,f2op4,f2op5;
      JRadioButton f3op1,f3op2,f3op3,f3op4,f3op5;
      JRadioButton f4op1,f4op2,f4op3,f4op4,f4op5;
      ButtonGroup
questionc1group,questionc2group,questionc3group,questionc4group;

       public ArrayList<Evaluation> getEvalData() {
                  return evalData;
            }
```

```
	public void fillQuestions()
	{
		functional.add("1: How could you describe the software system is
performing for the intended functions.");
		functional.add("2: How could the users satisfied on the system");
		functional.add("3: Ease of the software for user to learn and
operate ");
		functional.add("4: Capability of the software to present data
effectively");

		optionsList.add("Very low");
		optionsList.add("Low");
		optionsList.add("Medium");
		optionsList.add("High");
		optionsList.add("Very High");

	}
	public FunctionalCriteria(JFrame parent) {
		super(parent, true);
		configureFrame();
	}

	protected ImageIcon createImageIcon(String path, String description) {
		java.net.URL imgURL = getClass().getResource(path);
		if (imgURL != null) {
			return new ImageIcon(imgURL, description);
		} else {
			System.err.println("Couldn't find file: " + path);
			return null;
		}
	}
		private void configureFrame() {
		try {
			fillQuestions();
	UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
			this.setTitle("Functional Criteria");
			image = new JLabel();
			ImageIcon icon = createImageIcon("Images/f.jpg",
"blahblah");
			image.setIcon(icon);
			this.getContentPane().setLayout(new BorderLayout());
			this.getContentPane().add(image, BorderLayout.NORTH);
			JPanel qpanel = new JPanel();
			qpanel.setLayout(new SpringLayout());
			JPanel c1panel=new JPanel();
		c1panel.setLayout(new SpringLayout());

			evalData=new ArrayList<Evaluation>();
```

```
 f1=new Evaluation();

int constWeight=8;
float constNS=0.455f;

f1.setCriteria("F"+1);
f1.setWeight(constWeight);
f1.setNs(constNS);

questionf1=new JLabel(functional.get(0));
qpanel.add(questionf1);

questionc1group=new ButtonGroup();

f1op1=new JRadioButton("Very Low");
f1op2=new JRadioButton("Low");
f1op3=new JRadioButton("Medium");
f1op4=new JRadioButton("High");
f1op4.setSelected(true);
f1.setRate(3);
f1op5=new JRadioButton("Very High");

questionc1group.add(f1op1);
questionc1group.add(f1op2);
questionc1group.add(f1op3);
questionc1group.add(f1op4);
questionc1group.add(f1op5);

c1panel.add(f1op1);
c1panel.add(f1op2);
c1panel.add(f1op3);
c1panel.add(f1op4);
c1panel.add(f1op5);

f1op1.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
            f1.setRate(0);

f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
                        f1.setIs_matched(false);
      }
});

f1op2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
            f1.setRate(1);

f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
                        f1.setIs_matched(false);
```

```
                  }
            });
            f1op3.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f1.setRate(2);

f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
                              f1.setIs_matched(false);
                  }
            });
            f1op4.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f1.setRate(3);
            f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
                              f1.setIs_matched(true);
                  }
            });
            f1op5.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {

                        f1.setRate(4);
            f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
                              f1.setIs_matched(true);
                  }
            });
            f1.setScore(f1.getWeight()*f1.getNs()*f1.getRate());
            if(f1.getRate()<3)
                  f1.setIs_matched(false);
            else
                  f1.setIs_matched(true);
            Utils.makeCompactGrid(c1panel, 1, 5, // rows, cols
                        15, 15, // initX, initY
                        10, 10); // xPad, yPad
            qpanel.add(c1panel);

            JPanel c2panel=new JPanel();
      c2panel.setLayout(new SpringLayout());

            f2=new Evaluation();
            int constWeight2=6;
            float constNS2=0.12f;
            f2.setCriteria("F"+2);
            f2.setWeight(constWeight2);
            f2.setNs(constNS2);
            questionf2=new JLabel(functional.get(1));
            qpanel.add(questionf2);
            questionc2group=new ButtonGroup();
            f2op1=new JRadioButton("Very Low");
            f2op2=new JRadioButton("Low");
```

```
            f2op3=new JRadioButton("Medium");
            f2op4=new JRadioButton("High");
            f2op5=new JRadioButton("Very High");
            f2op5.setSelected(true);
            f2.setRate(4);
            questionc2group.add(f2op1);
            questionc2group.add(f2op2);
            questionc2group.add(f2op3);
            questionc2group.add(f2op4);
            questionc2group.add(f2op5);

            c2panel.add(f2op1);
            c2panel.add(f2op2);
            c2panel.add(f2op3);
            c2panel.add(f2op4);
            c2panel.add(f2op5);

            f2op1.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f2.setRate(0);

f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
                        f2.setIs_matched(false);
                  }
            });

            f2op2.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f2.setRate(1);

f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
                        f2.setIs_matched(false);
                        }
            });
            f2op3.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f2.setRate(2);

f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
                        f2.setIs_matched(false);
                        }
            });
            f2op4.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {
                        f2.setRate(3);
            f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
                        f2.setIs_matched(true);

                  }
```

```
            });
            f2op5.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent e) {

                        f2.setRate(4);

f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
                        f2.setIs_matched(true);
                        }
            });


            f2.setScore(f2.getWeight()*f2.getNs()*f2.getRate());
            if(f2.getRate()<3)
                  f2.setIs_matched(false);
            else
                  f2.setIs_matched(true);
            Utils.makeCompactGrid(c2panel, 1, 5, // rows, cols
                        15, 15, // initX, initY
                        10, 10); // xPad, yPad
            qpanel.add(c2panel);

                        JPanel c3panel=new JPanel();
                  c3panel.setLayout(new SpringLayout());

                        f3=new Evaluation();

                        int constWeight3=7;
                        float constNS3=0.24f;

                        f3.setCriteria("F"+3);
                        f3.setWeight(constWeight3);
                        f3.setNs(constNS3);

                        questionf3=new JLabel(functional.get(2));
                        qpanel.add(questionf3);
                        questionc3group=new ButtonGroup();

                        f3op1=new JRadioButton("Very Low");
                        f3op2=new JRadioButton("Low");
                        f3op3=new JRadioButton("Medium");
                        f3op3.setSelected(true);
                        f3.setRate(2);

                        f3op4=new JRadioButton("High");
                        f3op5=new JRadioButton("Very High");

                        questionc3group.add(f3op1);
                        questionc3group.add(f3op2);
```

```
                              questionc3group.add(f3op3);
                              questionc3group.add(f3op4);
                              questionc3group.add(f3op5);

                              c3panel.add(f3op1);
                              c3panel.add(f3op2);
                              c3panel.add(f3op3);
                              c3panel.add(f3op4);
                              c3panel.add(f3op5);

                              f3op1.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent
e) {
                                          f3.setRate(0);

      f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                                                f3.setIs_matched(false);
                                    }
                              });

                              f3op2.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent
e) {
                                          f3.setRate(1);

      f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                                                f3.setIs_matched(false);
                                    }
                              });
                              f3op3.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent
e) {
                                          f3.setRate(2);

      f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                                                f3.setIs_matched(false);
                                    }
                              });
                              f3op4.addActionListener(new ActionListener() {
                                    public void actionPerformed(ActionEvent
e) {
                                          f3.setRate(3);


      f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                                                f3.setIs_matched(true);
                                    }
                              });
                              f3op5.addActionListener(new ActionListener() {
```

```
                                    public void actionPerformed(ActionEvent
e) {

                                          f3.setRate(4);


      f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                                                f3.setIs_matched(true);
                                    }
                              });
                  f3.setScore(f3.getWeight()*f3.getNs()*f3.getRate());
                              if(f3.getRate()<3)
                                    f3.setIs_matched(false);
                              else
                                    f3.setIs_matched(true);


                              // Lay out the panel.
                              Utils.makeCompactGrid(c3panel, 1, 5, // rows,
cols
                                          15, 15, // initX, initY
                                          10, 10); // xPad, yPad
                              qpanel.add(c3panel);

                                          JPanel c4panel=new JPanel();
                                    c4panel.setLayout(new SpringLayout());
                                          f4=new Evaluation();

                                          int constWeight4=5;
                                          float constNS4=0.17f;

                                          f4.setCriteria("F"+4);
                                          f4.setWeight(constWeight4);
                                          f4.setNs(constNS4);

                                          questionf4=new
JLabel(functional.get(3));
                                          qpanel.add(questionf4);
                                          questionc4group=new ButtonGroup();
                                          f4op1=new JRadioButton("Very Low");
                                          f4op2=new JRadioButton("Low");
                                          f4op3=new JRadioButton("Medium");
                                          f4op4=new JRadioButton("High");
                                          f4op4.setSelected(true);
                                          f4.setRate(3);
                                          f4op5=new JRadioButton("Very
High");

                                          questionc4group.add(f4op1);
```

```
                                        questionc4group.add(f4op2);
                                        questionc4group.add(f4op3);
                                        questionc4group.add(f4op4);
                                        questionc4group.add(f4op5);

                                        c4panel.add(f4op1);
                                        c4panel.add(f4op2);
                                        c4panel.add(f4op3);
                                        c4panel.add(f4op4);
                                        c4panel.add(f4op5);

                                        f4op1.addActionListener(new
ActionListener() {
                                              public void
actionPerformed(ActionEvent e) {
                                                    f4.setRate(0);

      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                                    if(f4.getRate()<3)

      f4.setIs_matched(false);
                                              }
                                        });

                                        f4op2.addActionListener(new
ActionListener() {
                                              public void
actionPerformed(ActionEvent e) {
                                                    f4.setRate(1);

      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                                    if(f4.getRate()<3)

      f4.setIs_matched(false);

                                              }
                                        });
                                        f4op3.addActionListener(new
ActionListener() {
                                              public void
actionPerformed(ActionEvent e) {
                                                    f4.setRate(2);

      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                                    if(f4.getRate()<3)

      f4.setIs_matched(false);
                                              }
                                        });
```

```
                                        f4op4.addActionListener(new
ActionListener() {
                                              public void
actionPerformed(ActionEvent e) {
                                                    f4.setRate(3);


      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                                    if(f4.getRate()<3)

      f4.setIs_matched(false);

                                              }
                                        });
                                        f4op5.addActionListener(new
ActionListener() {
                                              public void
actionPerformed(ActionEvent e) {

                                                    f4.setRate(4);


      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                                    if(f4.getRate()<3)

      f4.setIs_matched(false);

                                              }
                                        });



      f4.setScore(f4.getWeight()*f4.getNs()*f4.getRate());
                                        if(f4.getRate()<3)
                                              f4.setIs_matched(false);
                                        Utils.makeCompactGrid(c4panel, 1,
5, // rows, cols
                                                    15, 15, // initX, initY
                                                    10, 10); // xPad, yPad
                                        qpanel.add(c4panel);

                  Utils.makeCompactGrid(qpanel, 8, 1, // rows, cols
                              15, 15, // initX, initY
                              10, 10); // xPad, yPad

                  this.getContentPane().add(qpanel,BorderLayout.CENTER);

                  JPanel buttons = new JPanel();
                  buttons.setLayout(new GridLayout(1,1));
```

```
                JButton next;
          //    previous = new JButton("Previous");
                next = new JButton("Next");

                //buttons.add(previous);
                buttons.add(next);

                next.addActionListener(new ActionListener() {
                      public void actionPerformed(ActionEvent e) {
                       setQuery();
                      }
                });
                this.getContentPane().add(buttons,BorderLayout.SOUTH);
                this.pack();
                this.setSize(600, this.getHeight());
                this.setResizable(false);
                Dimension screenSize = java.awt.Toolkit.getDefaultToolkit()
                            .getScreenSize();
                setBounds((screenSize.width - this.getWidth()) / 2,
                            (screenSize.height - this.getHeight()) / 2,
getWidth(),
                            getHeight());

          } catch (Exception e1) {
                javax.swing.JOptionPane.showMessageDialog(null,
                            "Catched Exception in constructor");
                e1.printStackTrace();
          }
      }

      void setQuery() {

            db=new DBConnector();
            db.init("ashueval");
            evalData=new ArrayList<Evaluation>();

            evalData.add(f1);
            evalData.add(f2);
            evalData.add(f3);
          evalData.add(f4);
          int noOfSuccessess=0;
            for(int i=0;i<4;i++)
               {
                        boolean is_matched=evalData.get(i).getIs_matched();
                              String criteria=evalData.get(i).getCriteria();
                            Integer weight=evalData.get(i).getWeight();
                            Integer rate=evalData.get(i).getRate();
                            Float ns=evalData.get(i).getNs();
                            Float score=evalData.get(i).getScore();
```

```
                              System.out.println(criteria + "    " + weight + "
" + rate + "   " +
                                                    ns + "   " + score);
         String query = "UPDATE evaluation SET is_matched="+is_matched+",
weight=" + weight + ",rate=" + rate + ",ns=" + ns + ",score=" + score
                                                   + " WHERE eval_number=" +
SEBegin.evalNumber + " AND criteria='" + criteria + "'";
                                        try {
                                              if (db.st.executeUpdate(query)>0)
                                              {
                                                    noOfSuccessess+=1;
                                              }
                                        } catch (SQLException e) {
//                                            // TODO Auto-generated catch block
                                              e.printStackTrace();
                                        }
 }
                  if (noOfSuccessess==4)
                  {
                        this.setVisible(false);
                        SEBegin.rd = new ReliabilityDialog(SEBegin.main);
                        SEBegin.rd.setVisible(true);
                  }
      }

      public static void main(String[] args) {
            FunctionalCriteria qf = new FunctionalCriteria(null);
            qf.setVisible(true);
            System.out.println("Bye");
      }

}
```